

# Stealthwatch API documentation version v1

Base URL is `https://{serviceAddress}/sw-reporting/v1`

## Documentation

Retrieves this API documentation.

`/docs`

GET

GET /docs

### Response

#### HTTP status code 200

The HTML for this API document.

## Tenants

Provides access to basic information about the **Tenants** (*domains*) and the **Tags** (*host groups*) in the Stealthwatch System. The API results are based on permissions. The API can fetch a list of tenants and get the tags for each tenant (either as a list or organized in a tree).

`/tenants`

GET

GET /tenants

### Request

#### Headers

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0Ij0jMTYxOTU2MzQifQ.ChrisArcher
```

### Response

#### HTTP status code 200

#### Body

Type: application/json

#### Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": {
          "$ref": "tenant"
        }
      }
    }
  }
}
```

#### Example:

```
{
  "data": [
    {
      "id": 123,
      "displayName": "Acme Corporation"
    },
    {
      "id": 52,
      "displayName": "Acme Corporation Test"
    }
  ]
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}

GET

Retrieves a single tenant given an ID.

GET

/tenants/{tenantId}

### Request

**URI Parameters**

- **tenantId**: required (integer)

**Headers**

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiYXRhcnRlbnNpdz9kZXZhdGVkInQ=
```

### Response

**HTTP status code 200**

**Body**

Type: application/json

Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": {
        "$ref": "tenant"
      }
    }
  }
}
```

**Example:**

```
{
  "data": {
    "id": 123,
    "displayName": "Acme Corporation"
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

## Custom Hosts

`/tenants/{tenantId}/customHosts/tags`

GET

Retrieves all Custom Host Tags for the specific Tenant.

GET `/tenants/{tenantId}/customHosts/tags`

### Request

#### URI Parameters

- **tenantId**: *required (string)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI0IiwiaXNjaWkiOiJ0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91
```

### Response

#### HTTP status code 200

#### Body

Type: application/json

#### Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": {
          "$ref": "tags"
        }
      }
    }
  }
}
```

**Example:**

```
{
  "data": [
    {
      "id": 27,
      "displayName": "Tag Name 1"
    },
    {
      "id": 28,
      "displayName": "Tag Name 2"
    }
  ]
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

`/tenants/{tenantId}/customHosts/tags/{tagId}`

GET

Retrieves a single Custom Host Tag given an ID for the specific Tenant.

GET `/tenants/{tenantId}/customHosts/tags/{tagId}`

Request

**URI Parameters**

- **tenantId:** *required (string)*
- **tagId:** *required (integer)*

**Headers**

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.
```

Response

**HTTP status code 200**

**Body**

**Type:** application/json

**Schema:**

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": {
        "$ref": "tag"
      }
    }
  }
}
```



- **value:** For alarm trends, this element has the following aggregated values:

- **sourceCount:** The number of unique hosts that are the alarm sources for the day. Only hosts that the

user has access to are counted.

- **targetCount:** The number of unique hosts that are the alarm targets for the day. Only hosts that the

user has access to are counted.

- **severity:** This is the maximum severity of all the alarms for the day. Severity per alarm is the

ratio of the total number of points accumulated divided by the threshold.

#### Body

Type: application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      },
      "withinByteCount": {
        "type": "integer"
      },
      "granularity": {
        "type": "integer"
      }
    },
    "required": [
      "sentByteCount",
      "receivedByteCount",
      "withinByteCount"
    ]
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
```

```
    "timestamp": {
      "type": "string"
    },
    "value": {
      "type": "array",
      "items": {
        "type": {
          "oneOf": [
            {
              "$ref": "#/definitions/alarmTrend"
            },
            {
              "$ref": "#/definitions/trafficTrend"
            }
          ]
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



**HTTP status code 500**

Internal Server error.

**/tenants/{tenantId}/customHosts/tags/{tagId}/alarms/topHosts****GET**

Retrieves the top alarming hosts for a Custom Host Tag (tagId) of a Tenant (tenantId).

**GET** /tenants/{tenantId}/customHosts/tags/{tagId}/alarms/topHosts**Request****URI Parameters**

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*

**Headers**

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGU6IiwiaWF0IjoiYXNjaWkiLCJ1aW4iOiJlbnR5dXkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJkaWkiLCJ1aW4iOiJlbnR5dXkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJkaWkiLCJ1aW4iOiJlbnR5dXkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

**Query Parameters**

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

**Response****HTTP status code 200**If no *query parameters* are provided, then the response is top alarming hosts for today (since reset hour to now).The response has a **header** element with the following fields:

- **startTime**: start time for the top alarming hosts.
- **endTime**: end time for the top alarming hosts.

It has a **data** element which represents top alarming hosts data. Each element in the series has the following fields:

- **ipAddress**: IP Address of the Host.
- **hostGroupIds**: The set of host group IDs, which is the union of all the host group IDs associated with

```
the host (source/target) in all the alarms.
```

• **sourceCategoryEvents**:

- **typeId**: The category event type ID where this host is the source of an alarm.
- **severity**: The maximum severity level for this type of alarm category with the host as the source.

```
Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
```

- **alwaysBadCount**: The number of times an "always bad" alarm occurred for this alarm category with the

```
host as the source.
```

• **sourceSecurityEvents**:

- **typeId**: The security event type ID where this host is the source of a security event.
- **severity**: The maximum severity level for this type of security event with the host as the source.

```
Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
```

- **alwaysBadCount**: Number of times the alarm for this security event was always bad with the host as the source.

- **targetCategoryEvents:**

- **typeid:** The category event type ID where this host is the target of an alarm.
- **severity:** The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- **typeid:** The security event type ID where this host is the target of a security event.
- **severity:** The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this security event with the host as the target.

### Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            }
          },
          "required": [
            "typeid",
            "severity",
            "alwaysBadCount"
          ]
        }
      }
    }
  }
}
```

```
    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
},
"required": [
  "hostname"
```

```

        header ,
        "data"
    ]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [

```

```

        "typeId": 47,
        "severity": 0.0,
        "alwaysBadCount": 1
      },
      {
        "typeId": 56,
        "severity": 8.01,
        "alwaysBadCount": 0
      }
    ],
    "sourceSecurityEvents": [
      {
        "typeId": 63,
        "severity": 300.0,
        "alwaysBadCount": 0
      },
      {
        "typeId": 276,
        "severity": 140.0,
        "alwaysBadCount": 0
      }
    ],
    "targetCategoryEvents": [
      {
        "typeId": 46,
        "severity": 0.0,
        "alwaysBadCount": 1
      },
      {
        "typeId": 32,
        "severity": 300.0,
        "alwaysBadCount": 0
      }
    ],
    "targetSecurityEvents": [
      {
        "typeId": 286,
        "severity": 300.0,
        "alwaysBadCount": 0
      },
      {
        "typeId": 267,
        "severity": 140.0,
        "alwaysBadCount": 0
      }
    ]
  ]
}
}
}
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

**/tenants/{tenantId}/customHosts/tags/{tagId}/traffic/hourly**

**GET**

Retrieves the hourly traffic trend for a Custom Host Tag (tagId) of a Tenant (tenantId).

**GET**

/tenants/{tenantId}/customHosts/tags/{tagId}/traffic/hourly

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZC54ZW50LXBpZW4iLCJpYXN0LXBvdiI6IjEiLCJ0eXBlIjoiYm9keSIsImV4cCI6MTYwMjM0NTY3ODkwfQ.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZC54ZW50LXBpZW4iLCJpYXN0LXBvdiI6IjEiLCJ0eXBlIjoiYm9keSIsImV4cCI6MTYwMjM0NTY3ODkwfQ.
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not  
classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not  
classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

### Body

**Type:** application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount"
  ]
}
```

```
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
```

**Example:**





- **filter[startAbsolute]:** (*integer*)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      }
    }
  }
}
```

```

    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
}
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  }
},
"data": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "string"
      },
      "value": {
        "type": "array",
        "items": {
          "type": {
            "oneOf": [
              {
                "$ref": "#/definitions/alarmTrend"
              },
              {
                "$ref": "#/definitions/trafficTrend"
              }
            ]
          }
        }
      }
    }
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/customHosts/tags/{tagId}/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for a Custom Host Tag (tagId) of a Tenant (tenantId).

GET /tenants/{tenantId}/customHosts/tags/{tagId}/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)

- **tagId**: *required (integer)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpvcXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9
```

#### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

### Response

#### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trends for 25 hours (current hour and past 24 hours).

The response is an array of traffic trends with each element representing a trend per application.

A trend has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.
- **applicationId**: ID of the application.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

#### Body

Type: application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  }
}
```

```

    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      },
      "required": [
        "timestamp",
        "value"
      ]
    }
  },
  "required": [
    "header",
    "data"
  ]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.



**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```



```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request



- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

Type: application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

```

    },
    {
      "$ref": "#/definitions/trafficTrend"
    }
  ]
},
{
  "required": [
    "timestamp",
    "value"
  ]
}
},
{
  "required": [
    "header",
    "data"
  ]
}
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

Retrieves the raw traffic trend of an application (applicationId) for a Custom Host Tag (tagId) of a Tenant (tenantId).

**GET** /tenants/{tenantId}/customHosts/tags/{tagId}/applications/{applicationId}/traffic/raw

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*
- **applicationId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGV
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity**: This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
```

```

        "type": "integer"
    },
    "targetCount": {
        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [

```

```
required : [
  "header",
  "data"
]
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

`/tenants/{tenantId}/customHosts/tags/tree`

GET

Retrieves all Custom Host Tags for the specific Tenant organized in a hierarchy.



## Request

### URI Parameters

- **tenantId**: *required (string)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

## Response

### HTTP status code 200

#### Body

Type: application/json

#### Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": {
        "$ref": "tagTree"
      }
    }
  }
}
```

#### Example:

```
{
  "id": 20,
  "displayName": "Parent Tag 1",
  "tags": [
    {
      "id": 27,
      "displayName": "Child Tag 11"
    },
    {
      "id": 28,
      "displayName": "Child Tag 12"
    }
  ]
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

Retrieves all External Geo Tags for the specific Tenant.

**GET** /tenants/{tenantId}/externalGeos/tags

Request

URI Parameters

- **tenantId**: required (string)

Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

Response

HTTP status code 200

Body

Type: application/json

Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": {
          "$ref": "tags"
        }
      }
    }
  }
}
```

Example:

```
{
  "data": [
    {
      "id": 27,
      "displayName": "Tag Name 1"
    },
    {
      "id": 28,
      "displayName": "Tag Name 2"
    }
  ]
}
```

HTTP status code 400

Bad or invalid request

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 403

Forbidden.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

HTTP status code 500

Internal Server error.

Retrieves a single External Geo Tag (tagId) given an ID for the specific Tenant (tenantId).

GET /tenants/{tenantId}/externalGeos/tags/{tagId}

Request

URI Parameters

- **tenantId**: required (string)
- **tagId**: required (integer)

Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZSIsImNpdWUiOiJhbnR5bWVudC5jb20iLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6Mj7,
```

Response

HTTP status code 200

Body

Type: application/json

Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": {
        "$ref": "tag"
      }
    }
  }
}
```

Example:

```
{
  "data": {
    "id": 27,
    "displayName": "Tag Name"
  }
}
```

HTTP status code 400

Bad or invalid request

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 403

Forbidden.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

HTTP status code 500

Internal Server error.

Retrieves the daily alarm trend for an External Geo Tag (tagId) of a Tenant (tenantId)



```

    }
  },
  "required": [
    "sourceCount",
    "targetCount",
    "severity"
  ]
},
"trafficTrend": {
  "type": "object",
  "properties": {
    "sentByteCount": {
      "type": "integer"
    },
    "receivedByteCount": {
      "type": "integer"
    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
}
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
}
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



- **targetCategoryEvents:**

- **typeid:** The category event type ID where this host is the target of an alarm.
- **severity:** The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- **typeid:** The security event type ID where this host is the target of a security event.
- **severity:** The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this security event with the host as the target.

### Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            }
          },
          "required": [
            "typeid",
            "severity",
            "alwaysBadCount"
          ]
        }
      }
    }
  }
}
```



```
    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
},
"required": [
  "hostname"
```

```

        header ,
        "data"
    ]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [

```

```

        "typeId": 47,
        "severity": 0.0,
        "alwaysBadCount": 1
      },
      {
        "typeId": 56,
        "severity": 8.01,
        "alwaysBadCount": 0
      }
    ],
    "sourceSecurityEvents": [
      {
        "typeId": 63,
        "severity": 300.0,
        "alwaysBadCount": 0
      },
      {
        "typeId": 276,
        "severity": 140.0,
        "alwaysBadCount": 0
      }
    ],
    "targetCategoryEvents": [
      {
        "typeId": 46,
        "severity": 0.0,
        "alwaysBadCount": 1
      },
      {
        "typeId": 32,
        "severity": 300.0,
        "alwaysBadCount": 0
      }
    ],
    "targetSecurityEvents": [
      {
        "typeId": 286,
        "severity": 300.0,
        "alwaysBadCount": 0
      },
      {
        "typeId": 267,
        "severity": 140.0,
        "alwaysBadCount": 0
      }
    ]
  ]
}
}
}
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalGeos/tags/{tagId}/traffic/hourly

GET

Retrieves the hourly traffic trend for an External Geo Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalGeos/tags/{tagId}/traffic/hourly

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not  
classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not  
classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

### Body

**Type:** application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount".
```

```
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
```

**Example:**



- **filter[startAbsolute]:** (*integer*)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      }
    }
  }
}
```

```

    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
}
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  }
},
"data": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "string"
      },
      "value": {
        "type": "array",
        "items": {
          "type": {
            "oneOf": [
              {
                "$ref": "#/definitions/alarmTrend"
              },
              {
                "$ref": "#/definitions/trafficTrend"
              }
            ]
          }
        }
      }
    }
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**



```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for an External Geo Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)

- **tagId**: *required (integer)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiMTUxMjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiMTUxMjM0NTY3ODkwIn0.
```

#### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trends for 25 hours (current hour and past 24 hours).

The response is an array of traffic trends with each element representing a trend per application.

A trend has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.
- **applicationId**: ID of the application.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning

of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not  
classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not  
classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

#### Body

Type: `application/json`

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  }
}
```

```

    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      },
      "required": [
        "timestamp",
        "value"
      ]
    }
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

`/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/traffic/raw`

GET

Retrieves the raw traffic trends of all applications for an External Geo Tag (tagId) of a Tenant (tenantId).

GET

`/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/traffic/raw`

### Request

#### URI Parameters

- `tenantId`: required (string)
- `tagId`: required (integer)

#### Headers

- `Cookie`: required (string)  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9
```

#### Query Parameters

- `filter[startAbsolute]`: (integer)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- `filter[endAbsolute]`: (integer)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- `filter[startRelative]`: (integer)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- `filter[intervalLength]`: (integer)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

### Response

#### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trends for 25 hours (current hour and past 24 hours).

The response is an array of traffic trends with each element representing a trend per application.

A trend has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.
- **applicationId**: ID of the application.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning

of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not

classified by the tag.

- **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not

classified by the tag.

- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

- **granularity**: This value is provided only for the raw traffic trend and represents the granularity

of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```

```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request





- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

Type: application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  }
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

```

    },
    {
      "$ref": "#/definitions/trafficTrend"
    }
  ]
},
"required": [
  "timestamp",
  "value"
]
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

Retrieves the raw traffic trend of an application (applicationId) for an External Geo Tag (tagId) of a Tenant (tenantId).

**GET** /tenants/{tenantId}/externalGeos/tags/{tagId}/applications/{applicationId}/traffic/raw

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*
- **applicationId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:

```
stealthwatch, jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGV
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity**: This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
```

```

        "type": "integer"
    },
    "targetCount": {
        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [

```

```
required : [
  "header",
  "data"
]
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/externalGeos/tags/tree

GET

Retrieves all External Geo Tags for the specific Tenant (tenantId) organized in a hierarchy.

GET

/tenants/{tenantId}/externalGeos/tags/tree

## Request

### URI Parameters

- **tenantId**: *required (string)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6Ikpvc3Q9Lm00NTY3ODkwIiwiaWF0IjoiMj01NjU0ODAwIn0=
```

## Response

### HTTP status code 200

#### Body

Type: application/json

#### Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": {
        "$ref": "tagTree"
      }
    }
  }
}
```

#### Example:

```
{
  "id": 20,
  "displayName": "Parent Tag 1",
  "tags": [
    {
      "id": 27,
      "displayName": "Child Tag 11"
    },
    {
      "id": 28,
      "displayName": "Child Tag 12"
    }
  ]
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalGeos/alarms/{alarmTypeId}/trend/daily

GET

Retrieves the daily alarm trend for External Geo Tags of a Tenant (tenantId).

## Request

## URI Parameters

- **tenantId**: *required (string)*
- **alarmTypeId**: *required (integer)*

## Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

## Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMjAxNi0xMi0xNSUwOTIyOjA5OjA5LjA9eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMjAxNi0xMi0xNSUwOTIyOjA5OjA5LjA9eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMjAxNi0xMi0xNSUwOTIyOjA5OjA5LjA9
```

## Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

## HTTP status code 200

If no *query parameters* are provided, then the response is a 7-day alarm trend.

The response has a **header** element that contains the following fields:

- **startTime**: Start time for the alarm trend.
- **endTime**: End time for the alarm trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For alarm trend this is the reset hour for the day.
- **value**: For alarm trends, this element has the following aggregated values:
  - **sourceCount**: The number of unique hosts that are the alarm sources for the day. Only hosts that the `user` has access to are counted.
  - **targetCount**: The number of unique hosts that are the alarm targets for the day. Only hosts that the `user` has access to are counted.
  - **severity**: This is the maximum severity of all the alarms for the day. Severity per alarm is the `ratio` of the total number of points accumulated divided by the threshold.

## Body

**Type:** application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
```



```

        "sourceCount",
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



- o **typeid**: The category event type ID where this host is the target of an alarm.
- o **severity**: The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- o **typeid**: The security event type ID where this host is the target of a security event.
- o **severity**: The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this security event with the host as the target.

## Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            },
            "required": [
              "typeid",
              "severity",
              "alwaysBadCount"
            ]
          }
        }
      }
    }
  }
}
```

```
    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
},
"required": [
  "header",
  "data"
```

```

    data
  }
},
"required": [
  "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,

```

```
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 56,
    "severity": 8.01,
    "alwaysBadCount": 0
  }
],
"sourceSecurityEvents": [
  {
    "typeId": 63,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 276,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
],
"targetCategoryEvents": [
  {
    "typeId": 46,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 32,
    "severity": 300.0,
    "alwaysBadCount": 0
  }
],
"targetSecurityEvents": [
  {
    "typeId": 286,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 267,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
]
}
]
}
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

`/tenants/{tenantId}/externalGeos/traffic/hourly`

GET

Retrieves the hourly traffic trend for External Geo Tags of a Tenant (tenantId).

GET

`/tenants/{tenantId}/externalGeos/traffic/hourly`

Request

## URI Parameters

- **tenantId**: *required (string)*

## Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZSIsInR5cCI6IkpXVCJ9.eyJ1b290eXkiOiJpvaG4gRGVhZSIsInR5cCI6IkpXVCJ9.eyJ1b290eXkiOiJpvaG4gRGVhZSIsInR5cCI6IkpXVCJ9.
```

## Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not  
classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not  
classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

**Type:** application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount",
    "targetCount",
    "severity"
  ]
},
```



```

"trafficTrend": {
  "type": "object",
  "properties": {
    "sentByteCount": {
      "type": "integer"
    },
    "receivedByteCount": {
      "type": "integer"
    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**



Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.

- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      },
      "withinByteCount": {
```

```

        "type": "integer"
    },
    "granularity": {
        "type": "integer"
    }
},
"required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [
    "header",
    "data"
]
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalGeos/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for External Geo Tags of a Tenant (tenantId).

GET /tenants/{tenantId}/externalGeos/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)



```

"type": "object",
"properties": {
  "sentByteCount": {
    "type": "integer"
  },
  "receivedByteCount": {
    "type": "integer"
  },
  "withinByteCount": {
    "type": "integer"
  },
  "granularity": {
    "type": "integer"
  }
},
"required": [
  "sentByteCount",
  "receivedByteCount",
  "withinByteCount"
]
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.





**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```

```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request



**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

```
GET /tenants/{tenantId}/externalGeos/{applicationId}/traffic/raw

    ],
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}
```

- HTTP status code 400**  
Bad or invalid request

---

- HTTP status code 401**  
Expired or invalid token. Client should re-authenticate.

---

- HTTP status code 403**  
Forbidden.

---

- HTTP status code 404**  
Not Found. Invalid or inaccessible path parameters.

---

- HTTP status code 500**  
Internal Server error.

## Request

## URI Parameters

- **tenantId**: *required (string)*
- **applicationId**: *required (integer)*

## Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

## Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG
```

## Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

## HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity**: This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

## Body

Type: application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
```



```

        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}

```

```
}  
}
```

**Example:**

```
{  
  "data": {  
    "header": {  
      "startTime": "2016-04-28T04:00:00Z",  
      "endTime": "2016-04-28T05:30:00Z"  
    },  
    "data": [  
      {  
        "timestamp": "2016-04-28T05:25:00Z",  
        "value": {  
          "outboundByteCount": 72294547625,  
          "inboundByteCount": 89717837575,  
          "withinByteCount": 12122355650,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T05:05:00Z",  
        "value": {  
          "outboundByteCount": 6103355775,  
          "inboundByteCount": 82136226125,  
          "withinByteCount": 11125132250,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T05:00:00Z",  
        "value": {  
          "outboundByteCount": 72294547625,  
          "inboundByteCount": 89717837575,  
          "withinByteCount": 12122355650,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T04:00:00Z",  
        "value": {  
          "outboundByteCount": 5069793300,  
          "inboundByteCount": 88965886200,  
          "withinByteCount": 8615046900,  
          "granularity": 3600  
        }  
      }  
    ]  
  }  
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

## External Hosts

**/tenants/{tenantId}/externalHosts/tags**

**GET**

Retrieves all External Host Tags for the specific Tenant (tenantId)



Retrieves a single External Host Tag (tagId) given an ID for the specific Tenant (tenantId).

**GET** /tenants/{tenantId}/externalHosts/tags/{tagId}

#### Request

##### URI Parameters

- **tenantId**: required (string)
- **tagId**: required (integer)

##### Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

##### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsInNpdCI6ImF1dG8ifQ==
```

#### Response

##### HTTP status code 200

##### Body

Type: application/json

##### Schema:

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": {
        "$ref": "tag"
      }
    }
  }
}
```

##### Example:

```
{
  "data": {
    "id": 27,
    "displayName": "Tag Name"
  }
}
```

##### HTTP status code 400

Bad or invalid request

##### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

##### HTTP status code 403

Forbidden.

##### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

##### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalHosts/tags/{tagId}/alarms/{alarmTypeId}/trend/daily

**GET**

Retrieves the alarm category (alarmTypeId) trend for an External Host Tag (tagId) of a Tenant (tenantId).

**GET** /tenants/{tenantId}/externalHosts/tags/{tagId}/alarms/{alarmTypeId}/trend/daily



```

        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            },
            "required": [
                "timestamp",
                "value"
            ]
        }
    }
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

**/tenants/{tenantId}/externalHosts/tags/{tagId}/alarms/topHosts****GET**

Retrieves top alarming hosts for an External Host Tag (tagId) of a Tenant (tenantId).

**GET**

/tenants/{tenantId}/externalHosts/tags/{tagId}/alarms/topHosts

## Request

**URI Parameters**

- **tenantId**: required (string)
- **tagId**: required (integer)

**Headers**

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMjAxMjE2MjMxMjE0In0
```

**Query Parameters**

- **filter[startAbsolute]**: (integer)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: (integer)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: (integer)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: (integer)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

**HTTP status code 200**If no *query parameters* are provided, then the response is top alarming hosts for today (since reset hour to now).The response has a **header** element with the following fields:

- **startTime**: start time for the top alarming hosts.
- **endTime**: end time for the top alarming hosts.

It has a **data** element which represents top alarming hosts data. Each element in the series has the following fields:

- **ipAddress**: IP Address of the Host.
- **hostGroupIds**: The set of host group IDs, which is the union of all the host group IDs associated with

```
the host (source/target) in all the alarms.
```

• **sourceCategoryEvents**:

- **typeId**: The category event type ID where this host is the source of an alarm.
- **severity**: The maximum severity level for this type of alarm category with the host as the source.

```
Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
```

- **alwaysBadCount**: The number of times an "always bad" alarm occurred for this alarm category with the

```
host as the source.
```

• **sourceSecurityEvents**:

- **typeId**: The security event type ID where this host is the source of a security event.
- **severity**: The maximum severity level for this type of security event with the host as the source.

```
Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
```

- **alwaysBadCount**: Number of times the alarm for this security event was always bad with the host as the source.



- **targetCategoryEvents:**

- **typeid:** The category event type ID where this host is the target of an alarm.
- **severity:** The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- **typeid:** The security event type ID where this host is the target of a security event.
- **severity:** The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this security event with the host as the target.

### Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            }
          },
          "required": [
            "typeid",
            "severity",
            "alwaysBadCount"
          ]
        }
      }
    }
  }
}
```

```

    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
}
},
"required": [
  "hostname"

```

```

        header ,
        "data"
    ]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [

```

```
        "typeId": 47,
        "severity": 0.0,
        "alwaysBadCount": 1
      },
      {
        "typeId": 56,
        "severity": 8.01,
        "alwaysBadCount": 0
      }
    ],
    "sourceSecurityEvents": [
      {
        "typeId": 63,
        "severity": 300.0,
        "alwaysBadCount": 0
      },
      {
        "typeId": 276,
        "severity": 140.0,
        "alwaysBadCount": 0
      }
    ],
    "targetCategoryEvents": [
      {
        "typeId": 46,
        "severity": 0.0,
        "alwaysBadCount": 1
      },
      {
        "typeId": 32,
        "severity": 300.0,
        "alwaysBadCount": 0
      }
    ],
    "targetSecurityEvents": [
      {
        "typeId": 286,
        "severity": 300.0,
        "alwaysBadCount": 0
      },
      {
        "typeId": 267,
        "severity": 140.0,
        "alwaysBadCount": 0
      }
    ]
  ]
}
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/externalHosts/tags/{tagId}/traffic/hourly

GET

Retrieves the hourly traffic trend for an External Host Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalHosts/tags/{tagId}/traffic/hourly

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjEiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not  
classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not  
classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

### Body

**Type:** application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount"
  ]
}
```

```
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
```

**Example:**



- **filter[startAbsolute]:** (*integer*)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      }
    }
  }
}
```



```

    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
}
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for an External Host Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)



```

    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      },
      "required": [
        "timestamp",
        "value"
      ]
    }
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.



**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```

```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**



```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

`/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/{applicationId}/traffic/hourly`

GET

Retrieves the hourly traffic trend of an application (applicationId) for an External Host Tag (tagId) of a Tenant (tenantId).

GET

`/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/{applicationId}/traffic/hourly`

#### Request

##### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*
- **applicationId**: *required (integer)*

##### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

##### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiNjA5MjAwMDAwMCJ9.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiNjA5MjAwMDAwMCJ9
```

##### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

#### Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:

- **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not

```
classified by the tag.
```

- **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not

```
classified by the tag.
```

- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

Type: application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

```

    },
    {
      "$ref": "#/definitions/trafficTrend"
    }
  ]
},
{
  "required": [
    "timestamp",
    "value"
  ]
}
},
{
  "required": [
    "header",
    "data"
  ]
}
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

Retrieves the raw traffic trend of an application (applicationId) for an External Host Tag (tagId) of a Tenant (tenantId).

**GET** /tenants/{tenantId}/externalHosts/tags/{tagId}/applications/{applicationId}/traffic/raw

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*
- **applicationId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsInNpdCI6Im9udGVudC51b250IHR5eSI6ImF1dG8iLCJ1aW4iOiJ1b250In0=
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity**: This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
```

```

        "type": "integer"
    },
    "targetCount": {
        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [

```

```
required :
  "header",
  "data"
]
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/externalHosts/tags/tree

GET

Retrieves all External Host Tags for the specific Tenant (tenantId) organized in a hierarchy.





## Request

## URI Parameters

- **tenantId**: *required (string)*
- **alarmTypeId**: *required (integer)*

## Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

## Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiMjAxNi0xMi0xNSUwOTIyOjA5OjA5Lm51LnR5cGU6Ij09eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

## Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

## HTTP status code 200

If no *query parameters* are provided, then the response is a 7-day alarm trend.

The response has a **header** element that contains the following fields:

- **startTime**: Start time for the alarm trend.
- **endTime**: End time for the alarm trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For alarm trend this is the reset hour for the day.
- **value**: For alarm trends, this element has the following aggregated values:
  - **sourceCount**: The number of unique hosts that are the alarm sources for the day. Only hosts that the `user` has access to are counted.
  - **targetCount**: The number of unique hosts that are the alarm targets for the day. Only hosts that the `user` has access to are counted.
  - **severity**: This is the maximum severity of all the alarms for the day. Severity per alarm is the `ratio` of the total number of points accumulated divided by the threshold.

## Body

**Type:** application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
```

```

        "sourceCount",
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



- o **typeid**: The category event type ID where this host is the target of an alarm.
- o **severity**: The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- o **typeid**: The security event type ID where this host is the target of a security event.
- o **severity**: The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this security event with the host as the target.

## Body

Type: application/json

## Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            },
            "required": [
              "typeid",
              "severity",
              "alwaysBadCount"
            ]
          }
        }
      }
    }
  }
}
```

```
    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
},
"required": [
  "header",
  "data"
```

```

    data
  }
},
"required": [
  "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,

```

```
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 56,
    "severity": 8.01,
    "alwaysBadCount": 0
  }
],
"sourceSecurityEvents": [
  {
    "typeId": 63,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 276,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
],
"targetCategoryEvents": [
  {
    "typeId": 46,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 32,
    "severity": 300.0,
    "alwaysBadCount": 0
  }
],
"targetSecurityEvents": [
  {
    "typeId": 286,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 267,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
]
}
]
}
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

`/tenants/{tenantId}/externalHosts/traffic/hourly`

GET

Retrieves the hourly traffic trend for External Host Tags of a Tenant (tenantId).

GET

`/tenants/{tenantId}/externalHosts/traffic/hourly`

Request



## URI Parameters

- **tenantId**: *required (string)*

## Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsInVudCI6IiJ9.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsInVudCI6IiJ9
```

## Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

**Type:** application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount",
    "targetCount",
    "severity"
  ]
},
```

```

"trafficTrend": {
  "type": "object",
  "properties": {
    "sentByteCount": {
      "type": "integer"
    },
    "receivedByteCount": {
      "type": "integer"
    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**



Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.

- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      },
      "withinByteCount": {
```

```

        "type": "integer"
    },
    "granularity": {
        "type": "integer"
    }
},
"required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [
    "header",
    "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalHosts/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for External Host Tags of a Tenant (tenantId).

GET /tenants/{tenantId}/externalHosts/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)



```

"type": "object",
"properties": {
  "sentByteCount": {
    "type": "integer"
  },
  "receivedByteCount": {
    "type": "integer"
  },
  "withinByteCount": {
    "type": "integer"
  },
  "granularity": {
    "type": "integer"
  }
},
"required": [
  "sentByteCount",
  "receivedByteCount",
  "withinByteCount"
]
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    }
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}

```

**Example:**



```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalHosts/applications/traffic/raw

GET

Retrieves the raw traffic trends of all applications for External Host Tags of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalHosts/applications/traffic/raw

#### Request

##### URI Parameters

- **tenantId**: *required (string)*

##### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

##### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiYXNjaWkiLCJpc0kiOiJhbnR5bW9udCJ9
```

##### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

#### Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trends for 25 hours (current hour and past 24 hours).

The response is an array of traffic trends with each element representing a trend per application.

A trend has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.
- **applicationId**: ID of the application.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning

```
of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:

- **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not

```
classified by the tag.
```

- **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not

```
classified by the tag.
```

- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

- **granularity**: This value is provided only for the raw traffic trend and represents the granularity

```
of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.
```

**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```

```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

`/tenants/{tenantId}/externalHosts/applications/{applicationId}/traffic/hourly`

GET

Retrieves the hourly traffic trend of an application (applicationId) for External Host Tags of a Tenant (tenantId).

GET

`/tenants/{tenantId}/externalHosts/applications/{applicationId}/traffic/hourly`

### Request

#### URI Parameters

- **tenantId**: *required (string)*
- **applicationId**: *required (integer)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.
```

#### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

### Response

**HTTP status code 200**

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not

```
classified by the tag.
```

- **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not

```
classified by the tag.
```

- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

```
GET /tenants/{tenantId}/externalHosts/applications/{applicationId}/traffic/raw

{
  "required": [
    "timestamp",
    "value"
  ]
},
{
  "required": [
    "header",
    "data"
  ]
}
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}
```

- HTTP status code 400**  
Bad or invalid request

---

- HTTP status code 401**  
Expired or invalid token. Client should re-authenticate.

---

- HTTP status code 403**  
Forbidden.

---

- HTTP status code 404**  
Not Found. Invalid or inaccessible path parameters.

---

- HTTP status code 500**  
Internal Server error.

Retrieves the raw traffic trend of an application (applicationId) for External Host Tags of a Tenant (tenantId).





```

        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}

```

```
}  
}
```

**Example:**

```
{  
  "data": {  
    "header": {  
      "startTime": "2016-04-28T04:00:00Z",  
      "endTime": "2016-04-28T05:30:00Z"  
    },  
    "data": [  
      {  
        "timestamp": "2016-04-28T05:25:00Z",  
        "value": {  
          "outboundByteCount": 72294547625,  
          "inboundByteCount": 89717837575,  
          "withinByteCount": 12122355650,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T05:05:00Z",  
        "value": {  
          "outboundByteCount": 6103355775,  
          "inboundByteCount": 82136226125,  
          "withinByteCount": 11125132250,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T05:00:00Z",  
        "value": {  
          "outboundByteCount": 72294547625,  
          "inboundByteCount": 89717837575,  
          "withinByteCount": 12122355650,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T04:00:00Z",  
        "value": {  
          "outboundByteCount": 5069793300,  
          "inboundByteCount": 88965886200,  
          "withinByteCount": 8615046900,  
          "granularity": 3600  
        }  
      }  
    ]  
  }  
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

## External Threats

`/tenants/{tenantId}/externalThreats/tags`

GET

Retrieves all External Threat Tags for the specific Tenant (tenantId)







```

        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            },
            "required": [
                "timestamp",
                "value"
            ]
        }
    }
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



**HTTP status code 500**

Internal Server error.

**/tenants/{tenantId}/externalThreats/tags/{tagId}/alarms/topHosts**

**GET**

Retrieves the top alarming hosts for an External Threat Tag (tagId) of a Tenant (tenantId).

**GET**

/tenants/{tenantId}/externalThreats/tags/{tagId}/alarms/topHosts

**Request**

**URI Parameters**

- **tenantId:** *required (string)*
- **tagId:** *required (integer)*

**Headers**

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiMTY4MjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiMTY4MjM0NTY3ODkwIn0
```

**Query Parameters**

- **filter[startAbsolute]:** *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]:** *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

**Response**

**HTTP status code 200**

If no *query parameters* are provided, then the response is top alarming hosts for today (since reset hour to now).

The response has a **header** element with the following fields:

- **startTime:** start time for the top alarming hosts.
- **endTime:** end time for the top alarming hosts.

It has a **data** element which represents top alarming hosts data. Each element in the series has the following fields:

- **ipAddress:** IP Address of the Host.
- **hostGroupIds:** The set of host group IDs, which is the union of all the host group IDs associated with

```
the host (source/target) in all the alarms.
```

• **sourceCategoryEvents:**

- **typeId:** The category event type ID where this host is the source of an alarm.
- **severity:** The maximum severity level for this type of alarm category with the host as the source.

```
Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
```

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this alarm category with the

```
host as the source.
```

• **sourceSecurityEvents:**

- **typeId:** The security event type ID where this host is the source of a security event.
- **severity:** The maximum severity level for this type of security event with the host as the source.

```
Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
```

- **alwaysBadCount:** Number of times the alarm for this security event was always bad with the host as the source.

- **targetCategoryEvents:**

- **typeid:** The category event type ID where this host is the target of an alarm.
- **severity:** The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- **typeid:** The security event type ID where this host is the target of a security event.
- **severity:** The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this security event with the host as the target.

### Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            }
          },
          "required": [
            "typeid",
            "severity",
            "alwaysBadCount"
          ]
        }
      }
    }
  }
}
```

```

    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
}
},
"required": [
  "hostname"

```

```

        header ,
        "data"
    ]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [

```

```
    "typeId": 47,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 56,
    "severity": 8.01,
    "alwaysBadCount": 0
  }
],
"sourceSecurityEvents": [
  {
    "typeId": 63,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 276,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
],
"targetCategoryEvents": [
  {
    "typeId": 46,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 32,
    "severity": 300.0,
    "alwaysBadCount": 0
  }
],
"targetSecurityEvents": [
  {
    "typeId": 286,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 267,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
]
}
]
}
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/externalThreats/tags/{tagId}/traffic/hourly

**GET**

Retrieves the hourly traffic trend for an External Threat Tag (tagId) of a Tenant (tenantId).

**GET**

/tenants/{tenantId}/externalThreats/tags/{tagId}/traffic/hourly

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjE2MzQ1NjM0NTY3ODkwIiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not  
classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not  
classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

### Body

**Type:** application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount"
  ]
}
```

```
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
```

**Example:**





- **filter[startAbsolute]:** (*integer*)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      }
    }
  }
}
```

```

    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
}
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  }
},
"data": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "string"
      },
      "value": {
        "type": "array",
        "items": {
          "type": {
            "oneOf": [
              {
                "$ref": "#/definitions/alarmTrend"
              },
              {
                "$ref": "#/definitions/trafficTrend"
              }
            ]
          }
        }
      }
    }
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/traffic/hourly

GET

Retrieves hourly traffic trends of all applications for an External Threat Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)



```

    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      },
      "required": [
        "timestamp",
        "value"
      ]
    }
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/traffic/raw

GET

Retrieves raw traffic trends of all applications for an External Threat Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/traffic/raw

### Request

#### URI Parameters

- **tenantId**: required (string)
- **tagId**: required (integer)

#### Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGR
```

#### Query Parameters

- **filter[startAbsolute]**: (integer)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: (integer)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: (integer)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: (integer)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

### Response

#### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trends for 25 hours (current hour and past 24 hours).

The response is an array of traffic trends with each element representing a trend per application.

A trend has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.
- **applicationId**: ID of the application.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity**: This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```



```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request



- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

Type: application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

```

    },
    {
      "$ref": "#/definitions/trafficTrend"
    }
  ]
},
"required": [
  "timestamp",
  "value"
]
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.



```

        "type": "integer"
    },
    "targetCount": {
        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    }
},
"required": [
    "timestamp",
    "value"
]
}
},
"required": [

```

```
required : [
  "header",
  "data"
]
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/externalThreats/tags/tree

GET

Retrieves all External Threat Tags for the specific Tenant (tenantId) organized in a hierarchy.







```

        "sourceCount",
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



- o **typeId**: The category event type ID where this host is the target of an alarm.
- o **severity**: The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents**:

- o **typeId**: The security event type ID where this host is the target of a security event.
- o **severity**: The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this security event with the host as the target.

## Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeId": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            },
            "required": [
              "typeId",
              "severity",
              "alwaysBadCount"
            ]
          }
        }
      }
    }
  }
}
```

```
    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
},
"required": [
  "header",
  "data"
```

```

    data
  }
},
"required": [
  "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,

```



```
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 56,
    "severity": 8.01,
    "alwaysBadCount": 0
  }
],
"sourceSecurityEvents": [
  {
    "typeId": 63,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 276,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
],
"targetCategoryEvents": [
  {
    "typeId": 46,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 32,
    "severity": 300.0,
    "alwaysBadCount": 0
  }
],
"targetSecurityEvents": [
  {
    "typeId": 286,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 267,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
]
}
]
}
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

## Flow Reports

This API provides the ability to perform a flow search and obtain the search status and results.

`/tenants/{tenantId}/flow-reports/top-ports/queries`

POST

POST

/tenants/{tenantId}/flow-reports/top-ports/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included
      - **excludes**: List of predefined application ids to be excluded
    - **direction**: Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols**: Port Protocols used or unused in the flow.
      - **includes**: List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes**: List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp

The following protocols are allowed to include or exclude

      - tcp: Should have a valid port number
      - udp: Should have a valid port number
      - icmp: Should not have a port number
  - **subject**:
    - **tags**: Specify the tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: Specify the IP addresses or range of IP addresses involved or not involved in the flow.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **peer**:
    - **tags**: Specify the peer tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: List of IP addresses to be included or excluded.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **orientation**: Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
    - EITHER (Default)
    - CLIENT
    - SERVER
  - **maxRows**: The maximum no. of records to be returned. The maximum value for this field is 5000.
    - Default: 50
  - **flowCollectors**: List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.
  - **orderBy**: The order based on which the records will be retrieved and sorted by (i.e. Bytes or Packets or Flows or TCP Connection). The value must be one of the following:
    - TOTAL\_BYTES (Default)
    - TOTAL\_PACKETS
    - TOTAL\_FLOWS
    - TOTAL\_CONNECTIONS
  - **standardOptions**: Flag to return the default columns.
    - Value should be either true or false
    - In case If the value is set as "true" or set to default, then implicitly fields part of AdvancedOptions will be set to its default value.
    - Default value: true
  - **advanceOptions**
    - **excludeBpsPps**: Flag to excludes bps/pps values.
      - Value should be either true or false

- Default value: true
- excludeOthers:** Flag to excludes Other Records
  - Value should be either true or false
  - Default value: true
- excludeCounts:** Flag to excludes Counts
  - Value should be either true or false
  - Default value: false

## Request

### URI Parameters

- tenantId:** *required (string)*

### Headers

- Cookie:** *required (string)*  
 JSON Web Token for the authenticated user.

### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsImVudCI6IjE5ODkxMjM0NTY3ODkwIiwiaWF0IjoiMTUxMjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsImVudCI6IjE5ODkxMjM0NTY3ODkwIiwiaWF0IjoiMTUxMjM0NTY3ODkwIn0.
```

### Body

**Type:** application/json

### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "searchName": {
      "type": "string"
    },
    "startTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "endTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "connection": {
      "type": "object",
      "properties": {
        "applications": {
          "type": "object",
          "properties": {
            "includes": {
              "type": "array",
              "items": {}
            },
            "excludes": {
              "type": "array",
              "items": {}
            }
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      }
    },
    "direction": {
      "type": "string",
      "enum": ["INBOUND_PLUS_OUTBOUND", "INBOUND", "OUTBOUND", "WITHIN"],
      "default": "INBOUND_PLUS_OUTBOUND"
    },
    "portProtocols": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      }
    }
  }
}
```

```
    },
    "required": [
      "includes",
      "excludes"
    ]
  }
},
"required": [
  "direction"
]
},
"subject": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    }
  },
  "required": []
},
"peer": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      }
    }
  }
}
```

```

    },
    "required": [
      "includes",
      "excludes"
    ]
  },
  "required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 400000
},
"flowCollectors": {
  "type": "array",
  "items": {}
},
"orderBy": {
  "type": "string",
  "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
  "default": "TOTAL_BYTES"
},
"excludeBpsPps": {
  "type": "boolean",
  "default": true
},
"excludeOthers": {
  "type": "boolean",
  "default": true
},
"excludeCounts": {
  "type": "boolean",
  "default": false
},
"defaultColumns": {
  "type": "boolean",
  "default": true
}
},
"required": [
  "startTime",
  "endTime"
]
}

```

**Example:**

```
{
  "searchName":"Top Reports on 3/1/2017 at 12:36 PM",
  "startTime":"2017-03-10T00:00:00.000",
  "endTime":"2017-03-10T00:05:00.000",
  "connection":{
    "applications":{
      "includes":[103119],
      "excludes":[2000]
    },
    "direction":"INBOUND_PLUS_OUTBOUND",
    "portProtocols":{
      "includes":["70/tcp"],
      "excludes":["80/tcp"]
    }
  },
  "subject":{
    "tags":{
      "includes":[63000],
      "excludes":[63006]
    },
    "ipAddresses":{
      "includes":["10.20.11.11"],
      "excludes":[]
    }
  },
  "peer":{
    "tags":{
      "includes":[47],
      "excludes":[60000]
    },
    "ipAddresses":{
      "includes":[],
      "excludes":["10.20.11.12"]
    }
  },
  "orientation":"either",
  "maxRows":50,
  "flowCollectors":["162"],
  "orderBy":"TOTAL_BYTES",
  "excludeBpsPps":true,
  "excludeOthers":true,
  "excludeCounts":false,
  "defaultColumns": true
}
```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      }
    },
    "required": [
      "queryId",
      "status"
    ]
  }
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/flow-reports/top-ports/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-ports/queries/{queryId}

Get the status of the search

Request

**URI Parameters**

- **tenantId**: *required (string)*
- **queryId**: *required (string)*

**Headers**

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      },
      "required": [
        "queryId",
        "status"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.



GET

/tenants/{tenantId}/flow-reports/top-ports/results/{queryId}

Get the Top Ports search results.

## Request

## URI Parameters

- **tenantId**: required (string)
- **queryId**: required (string)

## Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

## Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiYXNjaWkiLCJpc0kiOiJ1b3RlbnQifQ.eyJ0b28iOiJ1b3RlbnQifQ
```

## Response

## HTTP status code 200

It has a **data** element which contains the below sections

- **Sections:**
  - **summary**: The Total summary details of the search results.
  - **results**: The list of top flow records returned based on the search criteria. The number of records should not be greater than maxRows.
  - **others**: The consolidated result of rest of the records based on the search criteria when the number of records returned greater than maxRows.
- **Properties:** Each section contains the appropriate properties from the below list.
  - **deviceId**: Stealth watch appliance id
  - **rank**: Rank of the top port record
  - **percent**: Percentage of bytes used in the port
  - **records**: The number of flow records using the corresponding port in the selected direction
  - **count**: Number of ports which are included in the record
  - **packets**: The number of packets using the corresponding port
  - **flows**: The number of flows using the corresponding port
  - **connections**: Number of TCP connections used by the flows.
  - **trafficRateMin**: Smallest traffic rate using corresponding port in bits per second
  - **trafficRateMax**: Highest traffic rate using corresponding port in bits per second
  - **trafficRateAvg**: Average traffic rate using corresponding port in bits per second
  - **trafficRate95th**: Highest traffic rate using corresponding port by avoiding highest 5% traffic rate in bits per second
  - **packetRateMin**: Smallest traffic rate using corresponding port in packets per second
  - **packetRateMax**: Highest traffic rate using corresponding port in packets per second
  - **packetRateAvg**: Average traffic rate using corresponding port in packets per second
  - **packetRate95th**: Highest traffic rate using corresponding port by avoiding highest 5% traffic rate in packets per second
  - **hosts**: The number of hosts on network's side of the conversation
  - **hostFlows**: The number of flows used by hosts
  - **hostConnections**: Number of TCP connections used by the hosts
  - **hostClients**: The number of client hosts using the corresponding port
  - **hostClientBytes**: The number of bytes used by client hosts using the corresponding port
  - **hostClientPackets**: The number of bytes used by client hosts using the corresponding port
  - **hostServers**: The number of server hosts using the corresponding port
  - **hostServerBytes**: The number of bytes used by server hosts using the corresponding port
  - **hostServerPackets**: The number of bytes used by server hosts using the corresponding port
  - **peers**: The number of "Other Hosts" (i.e.) hosts on the other side of the conversation
  - **peerFlows**: The number of flows used by "Other Hosts"(peers)
  - **peerConnections**: Number of TCP connections used by the "Other Hosts"(peers)
  - **peerClients**: The number of client hosts among the "Other Hosts"(peers) using the corresponding port
  - **peerClientBytes**: The number of bytes used by client hosts among the "Other Hosts"(peers) using the corresponding port
  - **peerClientPackets**: The number of packets used by client hosts among the "Other Hosts"(peers) using the corresponding port
  - **peerServers**: The number of server hosts among the "Other Hosts"(peers) using the corresponding port
  - **peerServerBytes**: The number of bytes used by server hosts among the "Other Hosts"(peers) using the corresponding port
  - **peerServerPackets**: The number of packets used by server hosts among the "Other Hosts"(peers) using the corresponding port
  - **protocolNumber**: Provides number of the protocol

- o **port**: The port that the corresponding flow was using
- o **protocol**: The protocol that the corresponding flow was using
- o **hostRole**: The role of the hosts selected in the Filter
- o **hostBytes**: The number of bytes for the hosts selected in the Filter
- o **hostPackets**: The number of packets for the hosts selected in the Filter
- o **peerBytes**: The number of bytes for the "Other Hosts"(peers) selected in the Filter
- o **peerPackets**: The number of packets for the "Other Hosts"(peers) selected in the Filter
- o **peerRole**: The role of the "Other Hosts"(peer) determined in the Filter
- o **hostBytesRatio**: The ratio, in percent, of host bytes to the bytes used by the corresponding port
- o **peerBytesRatio**: The ratio, in percent, of peer bytes to the bytes used by the corresponding port
- o **clientBytesRatio**: The ratio in percent of client bytes to the bytes used by the corresponding port
- o **serverBytesRatio**: The ratio in percent of server bytes to the bytes used by the corresponding port

## Body

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "summary": {
          "type": "object",
          "properties": {
            "bytes": {
              "type": "integer"
            },
            "clientBytesRatio": {
              "type": "number"
            },
            "connections": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "deviceId": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "hostBytes": {
              "type": "integer"
            },
            "hostBytesRatio": {
              "type": "number"
            },
            "hostClientBytes": {
              "type": "integer"
            },
            "hostClientPackets": {
              "type": "integer"
            },
            "hostClients": {
              "type": "integer"
            },
            "hostConnections": {
              "type": "integer"
            },
            "hostFlows": {
              "type": "integer"
            },
            "hostPackets": {
              "type": "integer"
            },
            "hostRole": {
              "type": "string"
            },
            "hostServerBytes": {
              "type": "integer"
            },
            "hostServerPackets": {
              "type": "integer"
            },
            "hostServers": {
              "type": "integer"
            }
          }
        }
      }
    }
  }
}
```

```
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "number"
  },
  "packetRateAvg": {
    "type": "number"
  },
  "packetRateMax": {
    "type": "number"
  },
  "packetRateMin": {
    "type": "number"
  },
  "packets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerBytesRatio": {
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "trafficRate95th": {
    "type": "number"
  },
  "trafficRateAvg": {
    "type": "number"
  },
  "trafficRateMax": {
    "type": "number"
  },
  "trafficRateMin": {
    "type": "number"
  }
},
"required": [
```

```
"hostPackets",
"hostServerBytes",
"hostBytesRatio",
"packetRateMin",
"bytes",
"connections",
"trafficRateMin",
"hostBytes",
"hostClientPackets",
"peerFlows",
"hostServers",
"hostClientBytes",
"percent",
"trafficRateMax",
"trafficRate95th",
"clientBytesRatio",
"hostFlows",
"packetRateMax",
"peerBytes",
"serverBytesRatio",
"peerConnections",
"flows",
"records",
"hostServerPackets",
"peerPackets",
"deviceId",
"protocolNumber",
"peerClientPackets",
"peerServers",
"count",
"peers",
"packetRate95th",
"trafficRateAvg",
"peerBytesRatio",
"peerServerBytes",
"hostConnections",
"peerClients",
"packets",
"peerClientBytes",
"peerServerPackets",
"hosts",
"hostClients",
"peerRole",
"hostRole",
"packetRateAvg"
]
},
"others": {
  "type": "object",
  "properties": {
    "bytes": {
      "type": "integer"
    },
    "clientBytesRatio": {
      "type": "number"
    },
    "connections": {
      "type": "integer"
    },
    "count": {
      "type": "integer"
    },
    "deviceId": {
      "type": "integer"
    },
    "flows": {
      "type": "integer"
    },
    "hostBytes": {
      "type": "integer"
    },
    "hostBytesRatio": {
      "type": "number"
    },
    "hostClientBytes": {
      "type": "integer"
    },
    "hostClientPackets": {
      "type": "integer"
    },
    "hostClients": {
      "type": "integer"
    }
  }
}
```

```
},
"hostConnections": {
  "type": "integer"
},
"hostFlows": {
  "type": "integer"
},
"hostPackets": {
  "type": "integer"
},
"hostRole": {
  "type": "string"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
"peerFlows": {
  "type": "integer"
},
"peerPackets": {
  "type": "integer"
},
"peerRole": {
  "type": "string"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"percent": {
  "type": "number"
},
}
```

```
,
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "totalRank": {
    "type": "boolean"
  },
  "trafficRate95th": {
    "type": "number"
  },
  "trafficRateAvg": {
    "type": "number"
  },
  "trafficRateMax": {
    "type": "number"
  },
  "trafficRateMin": {
    "type": "number"
  }
},
"required": [
  "hostPackets",
  "hostServerBytes",
  "protocol",
  "hostBytesRatio",
  "packetRateMin",
  "bytes",
  "peerClientBytes",
  "connections",
  "trafficRateMin",
  "hostBytes",
  "peerFlows",
  "hostServers",
  "hostClientBytes",
  "percent",
  "clientBytesRatio",
  "trafficRateMax",
  "trafficRate95th",
  "hostFlows",
  "packetRateMax",
  "hostClientPackets",
  "peerBytes",
  "serverBytesRatio",
  "peerConnections",
  "flows",
  "records",
  "hostServerPackets",
  "peerPackets",
  "deviceId",
  "peerClientPackets",
  "peerServers",
  "count",
  "peers",
  "packetRate95th",
  "trafficRateAvg",
  "peerBytesRatio",
  "peerServerBytes",
  "hostConnections",
  "peerClients",
  "packets",
  "totalRank",
  "peerServerPackets",
  "hosts",
  "hostClients",
  "peerRole",
  "hostRole",
  "packetRateAvg"
]
},
"results": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "bytes": {
        "type": "integer"
      },
      "clientBytesRatio": {
        "type": "number"
      }
    }
  }
}
}
```

```
,
"connections": {
  "type": "integer"
},
"deviceId": {
  "type": "integer"
},
"flows": {
  "type": "integer"
},
"hostBytes": {
  "type": "integer"
},
"hostBytesRatio": {
  "type": "number"
},
"hostClientBytes": {
  "type": "integer"
},
"hostClientPackets": {
  "type": "integer"
},
"hostClients": {
  "type": "integer"
},
"hostConnections": {
  "type": "integer"
},
"hostFlows": {
  "type": "integer"
},
"hostPackets": {
  "type": "integer"
},
"hostRole": {
  "type": "string"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
},
```

```
"peerFlows": {
  "type": "integer"
},
"peerPackets": {
  "type": "integer"
},
"peerRole": {
  "type": "string"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"percent": {
  "type": "number"
},
"port": {
  "type": "integer"
},
"portProtocol": {
  "type": "object",
  "properties": {
    "port": {
      "type": "integer"
    },
    "protocol": {
      "type": "string"
    }
  },
  "required": [
    "protocol",
    "port"
  ]
},
"protocol": {
  "type": "string"
},
"protocolNumber": {
  "type": "integer"
},
"rank": {
  "type": "integer"
},
"records": {
  "type": "integer"
},
"serverBytesRatio": {
  "type": "number"
},
"totalRank": {
  "type": "boolean"
},
"trafficRate95th": {
  "type": "number"
},
"trafficRateAvg": {
  "type": "number"
},
"trafficRateMax": {
  "type": "number"
},
"trafficRateMin": {
  "type": "number"
}
},
"required": [
  "hostPackets",
  "hostServerBytes",
  "protocol",
  "hostBytesRatio",
  "packetRateMin",
  "bytes",
  "rank",
  "peerClientBytes",
```



```

        "connections",
        "trafficRateMin",
        "hostBytes",
        "port",
        "peerFlows",
        "hostServers",
        "hostClientBytes",
        "percent",
        "clientBytesRatio",
        "trafficRateMax",
        "trafficRate95th",
        "hostFlows",
        "packetRateMax",
        "hostClientPackets",
        "peerBytes",
        "serverBytesRatio",
        "peerConnections",
        "flows",
        "records",
        "hostServerPackets",
        "peerPackets",
        "deviceId",
        "portProtocol",
        "protocolNumber",
        "peerClientPackets",
        "peerServers",
        "peers",
        "packetRate95th",
        "trafficRateAvg",
        "peerBytesRatio",
        "peerServerBytes",
        "hostConnections",
        "peerClients",
        "packets",
        "totalRank",
        "peerServerPackets",
        "hosts",
        "hostClients",
        "peerRole",
        "hostRole",
        "packetRateAvg"
    ]
}
},
"required": [
    "results",
    "others",
    "summary"
]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data" : {
    "summary": {
      "deviceId": 139,
      "percent": 100.0,
      "records": 42212,
      "count": 1329,
      "bytes": 31217850388,
      "packets": 25724318,
      "flows": 9069,
      "connections": 0,
      "trafficRateMin": 0.0,
      "trafficRateMax": 0.0,
      "trafficRateAvg": 0.0,
      "trafficRate95th": 0.0,
      "packetRateMin": 0.0,
      "packetRateMax": 0.0,
      "packetRateAvg": 0.0,
      "packetRate95th": 0.0,
      "hosts": 8683,
      "hostFlows": 9069,
      "hostConnections": 0,
      "hostClients": 0
    }
  }
}

```

```
    "hostClients": 0,
    "hostClientBytes": 10076315750,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 5532609444,
    "hostServerPackets": 0,
    "peers": 8683,
    "peerFlows": 9069,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 10076315750,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 5532609444,
    "peerServerPackets": 0,
    "hostBytes": 15608925194,
    "hostPackets": 0,
    "peerBytes": 15608925194,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 64.55483401171844,
    "serverBytesRatio": 35.445165988281566,
    "hostRole": "CLIENT_AND_SERVER"
  },
  "others": {
    "deviceId": 139,
    "percent": 5.303421008886668,
    "records": 15052,
    "count": 1299,
    "bytes": 1655614036,
    "packets": 1786766,
    "flows": 2832,
    "connections": 0,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 4096,
    "hostFlows": 2832,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 13773749,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 814033269,
    "hostServerPackets": 0,
    "peers": 4096,
    "peerFlows": 2832,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 13773749,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 814033269,
    "peerServerPackets": 0,
    "hostBytes": 827807018,
    "hostPackets": 0,
    "peerBytes": 827807018,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 1.6638840575763276,
    "serverBytesRatio": 98.33611594242367,
    "hostRole": "CLIENT_AND_SERVER"
  },
  "results": [{
    "deviceId": 139,
    "rank": 1,
    "percent": 53.091093332841815,
    "records": 7008,
    "bytes": 16573898086,
    "packets": 13481334,
    "flows": 896,
    "connections": 0,
    "trafficRateMin": 0.0,
```

```
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 195,
"hostFlows": 896,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 8286530298,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 418745,
"hostServerPackets": 0,
"peers": 195,
"peerFlows": 896,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 8286530298,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 418745,
"peerServerPackets": 0,
"protocolNumber": 17,
"port": 2055,
"portProtocol": {
  "protocol": "UDP",
  "port": 2055
},
"protocol": "UDP",
"hostBytes": 8286949043,
"hostPackets": 0,
"peerBytes": 8286949043,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 99.99494693405465,
"serverBytesRatio": 0.005053065945345888,
"hostRole": "CLIENT_AND_SERVER"
}, {
  "deviceId": 139,
  "rank": 2,
  "percent": 31.75467542060667,
  "records": 4374,
  "bytes": 9913127064,
  "packets": 8667354,
  "flows": 692,
  "connections": 0,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 298,
  "hostFlows": 692,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 460964938,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 4495598594,
  "hostServerPackets": 0,
  "peers": 298,
  "peerFlows": 692,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 460964938,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 4495598594,
  "peerServerPackets": 0,
  "protocolNumber": 6,
  "port": 443,
  "portProtocol": {
    "protocol": "TCP",
    "port": 443
  }
}
```

```

    },
    "protocol": "TCP",
    "hostBytes": 4956563532,
    "hostPackets": 0,
    "peerBytes": 4956563532,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 9.30009138436279,
    "serverBytesRatio": 90.69990861563721,
    "hostRole": "CLIENT_AND_SERVER"
  }, {
    "deviceId": 139,
    "rank": 3,
    "percent": 6.301761574064726,
    "records": 38,
    "bytes": 1967274500,
    "packets": 735148,
    "flows": 5,
    "connections": 0,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 8,
    "hostFlows": 5,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 885776580,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 97860670,
    "hostServerPackets": 0,
    "peers": 8,
    "peerFlows": 5,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 885776580,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 97860670,
    "peerServerPackets": 0,
    "protocolNumber": 6,
    "port": 2049,
    "portProtocol": {
      "protocol": "TCP",
      "port": 2049
    },
    "protocol": "TCP",
    "hostBytes": 983637250,
    "hostPackets": 0,
    "peerBytes": 983637250,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 90.05114232914623,
    "serverBytesRatio": 9.948857670853762,
    "hostRole": "CLIENT_AND_SERVER"
  }
}
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

## HTTP status code 500

Internal Server error.

/tenants/{tenantId}/flow-reports/top-applications/queries

POST

Initiate Search

POST

/tenants/{tenantId}/flow-reports/top-applications/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included
      - **excludes**: List of predefined application ids to be excluded
    - **direction**: Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols**: Port Protocols used or unused in the flow.
      - **includes**: List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes**: List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp
- **subject**:
  - **tags**: Specify the tags involved or not involved in the flow
    - **includes**: List of tag ids to be included.
    - **excludes**: List of tag ids to be excluded.
  - **ipAddresses**: Specify the IP addresses or range of IP addresses involved or not involved in the flow.
    - **includes**: List of IP addresses to be included.
      - Ex: 192.168.10.10 or 192.168.10.10-15
    - **excludes**: List of IP addresses to be excluded.
      - Ex: 192.168.10.10 or 192.168.10.10-15
- **peer**:
  - **tags**: Specify the peer tags involved or not involved in the flow
    - **includes**: List of tag ids to be included.
    - **excludes**: List of tag ids to be excluded.
  - **ipAddresses**: List of IP addresses to be included or excluded.
    - **includes**: List of IP addresses to be included.
      - Ex: 192.168.10.10 or 192.168.10.10-15
    - **excludes**: List of IP addresses to be excluded.
      - Ex: 192.168.10.10 or 192.168.10.10-15
- **orientation**: Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
  - EITHER (Default)
  - CLIENT
  - SERVER
- **maxRows**: The maximum no. of records to be returned. The maximum value for this field is 5000.
  - Default: 50
- **flowCollectors**: List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.
- **orderBy**: The order based on which the records will be retrieved and sorted by (i.e. Bytes or Packets or Flows or TCP Connection). The value must be one of the following:
  - TOTAL\_BYTES (Default)
  - TOTAL\_PACKETS



```
    "portProtocols": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "required": [
      "direction"
    ]
  },
  "subject": {
    "type": "object",
    "properties": {
      "tags": {
        "type": "object",
        "properties": {
          "includes": {
            "type": "array",
            "items": {}
          },
          "excludes": {
            "type": "array",
            "items": {}
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      },
      "ipAddresses": {
        "type": "object",
        "properties": {
          "includes": {
            "type": "array",
            "items": {}
          },
          "excludes": {
            "type": "array",
            "items": {}
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      }
    },
    "required": []
  },
  "peer": {
    "type": "object",
    "properties": {
      "tags": {
        "type": "object",
        "properties": {
          "includes": {
            "type": "array",
            "items": {}
          },
          "excludes": {
            "type": "array",
            "items": {}
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      }
    }
  }
}
```

```

    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    }
  },
  "required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 40000
},
"flowCollectors": {
  "type": "array",
  "items": {}
},
"orderBy": {
  "type": "string",
  "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
  "default": "TOTAL_BYTES"
},
"excludeBpsPps": {
  "type": "boolean",
  "default": true
},
"excludeOthers": {
  "type": "boolean",
  "default": true
},
"excludeCounts": {
  "type": "boolean",
  "default": false
},
"defaultColumns": {
  "type": "boolean",
  "default": true
}
},
"required": [
  "startTime",
  "endTime"
]
}

```

**Example:**



```
{
  "searchName":"Top Reports on 3/1/2017 at 12:36 PM",
  "startTime":"2017-03-10T00:00:00.000",
  "endTime":"2017-03-10T00:05:00.000",
  "connection":{
    "applications":{
      "includes":[103119],
      "excludes":[2000]
    },
    "direction":"INBOUND_PLUS_OUTBOUND",
    "portProtocols":{
      "includes":["70/tcp"],
      "excludes":["80/tcp"]
    }
  },
  "subject":{
    "tags":{
      "includes":[63000],
      "excludes":[63006]
    },
    "ipAddresses":{
      "includes":["10.20.11.11"],
      "excludes":[]
    }
  },
  "peer":{
    "tags":{
      "includes":[47],
      "excludes":[60000]
    },
    "ipAddresses":{
      "includes":[],
      "excludes":["10.20.11.12"]
    }
  },
  "orientation":"either",
  "maxRows":50,
  "flowCollectors":["162"],
  "orderBy":"TOTAL_BYTES",
  "excludeBpsPps":true,
  "excludeOthers":true,
  "excludeCounts":false,
  "defaultColumns": true
}
```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

Type: application/json

Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      }
    },
    "required": [
      "queryId",
      "status"
    ]
  }
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/flow-reports/top-applications/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-applications/queries/{queryId}

Get the status of the search

**Request**

**URI Parameters**

- **tenantId**: *required (string)*
- **queryId**: *required (string)*

**Headers**

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      },
      "required": [
        "queryId",
        "status"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.



- o **peerServerPackets**: The number of packets used by server hosts among the "Other Hosts" using the corresponding application
- o **hostRole**: The role of the hosts selected in the Filter used in the corresponding application
- o **hostBytes**: The number of bytes for the hosts selected in the Filter using the corresponding application
- o **hostPackets**: The number of packets for the hosts selected in the Filter using the corresponding application
- o **peerRole**: The role of the "Other Hosts"(peers)
- o **peerBytes**: The number of bytes used by "Other Hosts" using the corresponding application
- o **peerPackets**: The number of packets for the "Other Hosts" determined in the Filter using the corresponding application
- o **hostBytesRatio**: The ratio, in percent, of host bytes to the bytes using the corresponding application
- o **peerBytesRatio**: The ratio, in percent, of peer bytes to the bytes using the corresponding application
- o **clientBytesRatio**: The ratio, in percent, of client bytes to the bytes using the corresponding application
- o **serverBytesRatio**: The ratio, in percent, of server bytes to the bytes using the corresponding application

## Body

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "others": {
          "type": "object",
          "properties": {
            "bytes": {
              "type": "integer"
            },
            "clientBytesRatio": {
              "type": "number"
            },
            "connections": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "deviceId": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "hostBytes": {
              "type": "integer"
            },
            "hostBytesRatio": {
              "type": "number"
            },
            "hostClientBytes": {
              "type": "integer"
            },
            "hostClientPackets": {
              "type": "integer"
            },
            "hostClients": {
              "type": "integer"
            },
            "hostConnections": {
              "type": "integer"
            },
            "hostFlows": {
              "type": "integer"
            },
            "hostPackets": {
              "type": "integer"
            },
            "hostRole": {
              "type": "string"
            },
            "hostServerBytes": {
              "type": "integer"
            },
            "hostServerPackets": {
              "type": "integer"
            },
            "hostServers": {

```

```
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "number"
  },
  "packetRateAvg": {
    "type": "number"
  },
  "packetRateMax": {
    "type": "number"
  },
  "packetRateMin": {
    "type": "number"
  },
  "packets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerBytesRatio": {
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "trafficRate95th": {
    "type": "number"
  },
  "trafficRateAvg": {
    "type": "number"
  },
  "trafficRateMax": {
    "type": "number"
  },
  "trafficRateMin": {
    "type": "number"
  }
},
"required": [
```

```
"hostPackets",
"hostServerBytes",
"peerFlows",
"packetRateMin",
"packets",
"connections",
"trafficRateMin",
"hostBytes",
"hostClientPackets",
"hostBytesRatio",
"hostClientBytes",
"percent",
"peerClientPackets",
"trafficRateMax",
"trafficRate95th",
"clientBytesRatio",
"hostFlows",
"packetRateMax",
"peerBytes",
"serverBytesRatio",
"peerConnections",
"flows",
"records",
"hostServerPackets",
"peerPackets",
"deviceId",
"hostServers",
"peerServers",
"count",
"peers",
"packetRate95th",
"trafficRateAvg",
"peerBytesRatio",
"peerServerBytes",
"hostConnections",
"peerClients",
"bytes",
"peerClientBytes",
"peerServerPackets",
"hosts",
"hostClients",
"peerRole",
"hostRole",
"packetRateAvg"
]
},
"results": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "application": {
        "type": "object",
        "properties": {
          "id": {
            "type": "integer"
          },
          "name": {
            "type": "string"
          }
        }
      },
      "required": [
        "id",
        "name"
      ]
    },
    "bytes": {
      "type": "integer"
    },
    "clientBytesRatio": {
      "type": "number"
    },
    "connections": {
      "type": "integer"
    },
    "deviceId": {
      "type": "integer"
    },
    "flows": {
      "type": "integer"
    },
    "hostBytes": {
```

```
    "type": "integer"
  },
  "hostBytesRatio": {
    "type": "number"
  },
  "hostClientBytes": {
    "type": "integer"
  },
  "hostClientPackets": {
    "type": "integer"
  },
  "hostClients": {
    "type": "integer"
  },
  "hostConnections": {
    "type": "integer"
  },
  "hostFlows": {
    "type": "integer"
  },
  "hostPackets": {
    "type": "integer"
  },
  "hostRole": {
    "type": "string"
  },
  "hostServerBytes": {
    "type": "integer"
  },
  "hostServerPackets": {
    "type": "integer"
  },
  "hostServers": {
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "number"
  },
  "packetRateAvg": {
    "type": "number"
  },
  "packetRateMax": {
    "type": "number"
  },
  "packetRateMin": {
    "type": "number"
  },
  "packets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerBytesRatio": {
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  }
```



```
    },
    "peerServerPackets": {
      "type": "integer"
    },
    "peerServers": {
      "type": "integer"
    },
    "peers": {
      "type": "integer"
    },
    "percent": {
      "type": "number"
    },
    "rank": {
      "type": "integer"
    },
    "records": {
      "type": "integer"
    },
    "serverBytesRatio": {
      "type": "number"
    },
    "trafficRate95th": {
      "type": "number"
    },
    "trafficRateAvg": {
      "type": "number"
    },
    "trafficRateMax": {
      "type": "number"
    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "rank",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "application",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ]
}
```

```
    }  
  },  
  "summary": {  
    "type": "object",  
    "properties": {  
      "bytes": {  
        "type": "integer"  
      },  
      "clientBytesRatio": {  
        "type": "number"  
      },  
      "connections": {  
        "type": "integer"  
      },  
      "count": {  
        "type": "integer"  
      },  
      "deviceId": {  
        "type": "integer"  
      },  
      "flows": {  
        "type": "integer"  
      },  
      "hostBytes": {  
        "type": "integer"  
      },  
      "hostBytesRatio": {  
        "type": "number"  
      },  
      "hostClientBytes": {  
        "type": "integer"  
      },  
      "hostClientPackets": {  
        "type": "integer"  
      },  
      "hostClients": {  
        "type": "integer"  
      },  
      "hostConnections": {  
        "type": "integer"  
      },  
      "hostFlows": {  
        "type": "integer"  
      },  
      "hostPackets": {  
        "type": "integer"  
      },  
      "hostRole": {  
        "type": "string"  
      },  
      "hostServerBytes": {  
        "type": "integer"  
      },  
      "hostServerPackets": {  
        "type": "integer"  
      },  
      "hostServers": {  
        "type": "integer"  
      },  
      "hosts": {  
        "type": "integer"  
      },  
      "packetRate95th": {  
        "type": "number"  
      },  
      "packetRateAvg": {  
        "type": "number"  
      },  
      "packetRateMax": {  
        "type": "number"  
      },  
      "packetRateMin": {  
        "type": "number"  
      },  
      "packets": {  
        "type": "integer"  
      },  
      "peerBytes": {  
        "type": "integer"  
      },  
      "peerBytesRatio": {
```

```
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "trafficRate95th": {
    "type": "number"
  },
  "trafficRateAvg": {
    "type": "number"
  },
  "trafficRateMax": {
    "type": "number"
  },
  "trafficRateMin": {
    "type": "number"
  }
},
"required": [
  "hostPackets",
  "hostServerBytes",
  "peerFlows",
  "packetRateMin",
  "packets",
  "connections",
  "trafficRateMin",
  "hostBytes",
  "hostClientPackets",
  "hostBytesRatio",
  "hostClientBytes",
  "percent",
  "peerClientPackets",
  "trafficRateMax",
  "trafficRate95th",
  "clientBytesRatio",
  "hostFlows",
  "packetRateMax",
  "peerBytes",
  "serverBytesRatio",
  "peerConnections",
  "flows",
  "records",
  "hostServerPackets",
```

```

        "peerPackets",
        "deviceId",
        "hostServers",
        "peerServers",
        "count",
        "peers",
        "packetRate95th",
        "trafficRateAvg",
        "peerBytesRatio",
        "peerServerBytes",
        "hostConnections",
        "peerClients",
        "bytes",
        "peerClientBytes",
        "peerServerPackets",
        "hosts",
        "hostClients",
        "peerRole",
        "hostRole",
        "packetRateAvg"
    ]
}
},
"required": [
    "others",
    "results",
    "summary"
]
}
},
"required": [
    "data"
]
}
}

```

**Example:**

```

{
  "data" : {
    "summary": {
      "deviceId": 362,
      "percent": 100.0,
      "records": 46918,
      "count": 53,
      "bytes": 30739484298,
      "packets": 24446502,
      "flows": 11156,
      "connections": 14086,
      "trafficRateMin": 0.0,
      "trafficRateMax": 0.0,
      "trafficRateAvg": 0.0,
      "trafficRate95th": 0.0,
      "packetRateMin": 0.0,
      "packetRateMax": 0.0,
      "packetRateAvg": 0.0,
      "packetRate95th": 0.0,
      "hosts": 8192,
      "hostFlows": 11156,
      "hostConnections": 0,
      "hostClients": 0,
      "hostClientBytes": 9437706469,
      "hostClientPackets": 0,
      "hostServers": 0,
      "hostServerBytes": 5932035680,
      "hostServerPackets": 0,
      "peers": 8192,
      "peerFlows": 11156,
      "peerConnections": 0,
      "peerClients": 0,
      "peerClientBytes": 9437706469,
      "peerClientPackets": 0,
      "peerServers": 0,
      "peerServerBytes": 5932035680,
      "peerServerPackets": 0,
      "hostRole": "CLIENT_AND_SERVER",
      "hostBytes": 15369742149,
      "hostPackets": 0,
      "peerBytes": 15369742149,
      "peerPackets": 0,
      "peerRole": "CLIENT_AND_SERVER",
      "hostBytesRatio": 50.0,
      "peerBytesRatio": 50.0
    }
  }
}

```

```
    peerBytesRatio": 50.0,
    "clientBytesRatio": 61.40445543918279,
    "serverBytesRatio": 38.59554456081721
  },
  "others": {
    "deviceId": 362,
    "percent": 13.24443139817049,
    "records": 28446,
    "count": 50,
    "bytes": 4071269910,
    "packets": 2353882,
    "flows": 8617,
    "connections": 7338,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 7588,
    "hostFlows": 8617,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 1260816910,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 774818045,
    "hostServerPackets": 0,
    "peers": 7588,
    "peerFlows": 8617,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 1260816910,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 774818045,
    "peerServerPackets": 0,
    "hostRole": "CLIENT_AND_SERVER",
    "hostBytes": 2035634955,
    "hostPackets": 0,
    "peerBytes": 2035634955,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 61.9372794175663,
    "serverBytesRatio": 38.0627205824337
  },
  "results": [{
    "deviceId": 362,
    "rank": 1,
    "percent": 37.463533195152756,
    "records": 10348,
    "bytes": 11516096904,
    "packets": 10100724,
    "flows": 1407,
    "connections": 0,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 268,
    "hostFlows": 1407,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 4877674474,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 880373978,
    "hostServerPackets": 0,
    "peers": 268,
    "peerFlows": 1407,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 4877674474,
    "peerClientPackets": 0,
```

```
"peerServers": 0,
"peerServerBytes": 880373978,
"peerServerPackets": 0,
"application": {
  "id": 170,
  "name": "Undefined UDP"
},
"hostRole": "CLIENT_AND_SERVER",
"hostBytes": 5758048452,
"hostPackets": 0,
"peerBytes": 5758048452,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 84.71054932345679,
"serverBytesRatio": 15.289450676543213
}, {
  "deviceId": 362,
  "rank": 2,
  "percent": 29.74654398673478,
  "records": 1302,
  "bytes": 9143934218,
  "packets": 7747272,
  "flows": 225,
  "connections": 6748,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 180,
  "hostFlows": 225,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 295725987,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 4276241122,
  "hostServerPackets": 0,
  "peers": 180,
  "peerFlows": 225,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 295725987,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 4276241122,
  "peerServerPackets": 0,
  "application": {
    "id": 171,
    "name": "HTTPS (unclassified)"
  },
  "hostRole": "CLIENT_AND_SERVER",
  "hostBytes": 4571967109,
  "hostPackets": 0,
  "peerBytes": 4571967109,
  "peerPackets": 0,
  "peerRole": "CLIENT_AND_SERVER",
  "hostBytesRatio": 50.0,
  "peerBytesRatio": 50.0,
  "clientBytesRatio": 6.4682439735373,
  "serverBytesRatio": 93.5317560264627
}, {
  "deviceId": 362,
  "rank": 3,
  "percent": 19.54549141994197,
  "records": 6822,
  "bytes": 6008183266,
  "packets": 4244624,
  "flows": 907,
  "connections": 0,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
```

```

"packetRate95th": 0.0,
"hosts": 156,
"hostFlows": 907,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 3003489098,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 602535,
"hostServerPackets": 0,
"peers": 156,
"peerFlows": 907,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 3003489098,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 602535,
"peerServerPackets": 0,
"application": {
  "id": 38,
  "name": "NetFlow/sFlow"
},
"hostRole": "CLIENT_AND_SERVER",
"hostBytes": 3004091633,
"hostPackets": 0,
"peerBytes": 3004091633,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 99.97994285549146,
"serverBytesRatio": 0.020057144508547686
}]
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

## /tenants/{tenantId}/flow-reports/top-protocols/queries

POST

Initiate Search

POST /tenants/{tenantId}/flow-reports/top-protocols/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included
      - **excludes**: List of predefined application ids to be excluded

- **direction**: Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols**: Port Protocols used or unused in the flow.
      - **includes**: List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes**: List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp

The following protocols are allowed to include or exclude

      - tcp: Should have a valid port number
      - udp: Should have a valid port number
      - icmp: Should not have a port number
  - **subject**:
    - **tags**: Specify the tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: Specify the IP addresses or range of IP addresses involved or not involved in the flow.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **peer**:
    - **tags**: Specify the peer tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: List of IP addresses to be included or excluded.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **orientation**: Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
    - EITHER (Default)
    - CLIENT
    - SERVER
  - **maxRows**: The maximum no. of records to be returned. The maximum value for this field is 5000.
    - Default: 50
  - **flowCollectors**: List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.
  - **orderBy**: The order based on which the records will be retrieved and sorted by (i.e. Bytes or Packets or Flows or TCP Connection). The value must be one of the following:
    - TOTAL\_BYTES (Default)
    - TOTAL\_PACKETS
    - TOTAL\_FLOWS
    - TOTAL\_CONNECTIONS
  - **standardOptions**: Flag to return the default columns.
    - Value should be either true or false
    - In case if the value is set as "true" or set to default, then implicitly fields part of AdvancedOptions will be set to its default value.
    - Default value: true
  - **advanceOptions**
    - **excludeBpsPps**: Flag to excludes bps/pps values.
      - Value should be either true or false
      - Default value: true
    - **excludeOthers**: Flag to excludes Other Records
      - Value should be either true or false
      - Default value: true
    - **excludeCounts**: Flag to excludes Counts
      - Value should be either true or false
      - Default value: false

## Request

### URI Parameters

- **tenantId**: *required (string)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:



**Body****Type:** application/json**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "searchName": {
      "type": "string"
    },
    "startTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "endTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "connection": {
      "type": "object",
      "properties": {
        "applications": {
          "type": "object",
          "properties": {
            "includes": {
              "type": "array",
              "items": {}
            },
            "excludes": {
              "type": "array",
              "items": {}
            }
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      },
      "direction": {
        "type": "string",
        "enum": ["INBOUND_PLUS_OUTBOUND", "INBOUND", "OUTBOUND", "WITHIN"],
        "default": "INBOUND_PLUS_OUTBOUND"
      },
      "portProtocols": {
        "type": "object",
        "properties": {
          "includes": {
            "type": "array",
            "items": {}
          },
          "excludes": {
            "type": "array",
            "items": {}
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      }
    },
    "required": [
      "direction"
    ]
  },
  "subject": {
    "type": "object",
    "properties": {
      "tags": {
        "type": "object",
        "properties": {
          "includes": {
            "type": "array",
            "items": {}
          }
        }
      }
    }
  }
}
```

```

    },
    "excludes": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "includes",
    "excludes"
  ]
},
"ipAddresses": {
  "type": "object",
  "properties": {
    "includes": {
      "type": "array",
      "items": {}
    },
    "excludes": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "includes",
    "excludes"
  ]
}
},
"required": []
},
"peer": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      }
    },
    "required": [
      "includes",
      "excludes"
    ]
  },
  "ipAddresses": {
    "type": "object",
    "properties": {
      "includes": {
        "type": "array",
        "items": {}
      },
      "excludes": {
        "type": "array",
        "items": {}
      }
    },
    "required": [
      "includes",
      "excludes"
    ]
  }
},
"required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 400000
},
"flowCollectors": {

```

```

    "type": "array",
    "items": {}
  },
  "orderBy": {
    "type": "string",
    "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
    "default": "TOTAL_BYTES"
  },
  "excludeBpsPps": {
    "type": "boolean",
    "default": true
  },
  "excludeOthers": {
    "type": "boolean",
    "default": true
  },
  "excludeCounts": {
    "type": "boolean",
    "default": false
  },
  "defaultColumns": {
    "type": "boolean",
    "default": true
  }
},
"required": [
  "startTime",
  "endTime"
]
}

```

**Example:**

```

{
  "searchName": "Top Reports on 3/1/2017 at 12:36 PM",
  "startTime": "2017-03-10T00:00:00.000",
  "endTime": "2017-03-10T00:05:00.000",
  "connection": {
    "applications": {
      "includes": [103119],
      "excludes": [2000]
    },
    "direction": "INBOUND_PLUS_OUTBOUND",
    "portProtocols": {
      "includes": ["70/tcp"],
      "excludes": ["80/tcp"]
    }
  },
  "subject": {
    "tags": {
      "includes": [63000],
      "excludes": [63006]
    },
    "ipAddresses": {
      "includes": ["10.20.11.11"],
      "excludes": []
    }
  },
  "peer": {
    "tags": {
      "includes": [47],
      "excludes": [60000]
    },
    "ipAddresses": {
      "includes": [],
      "excludes": ["10.20.11.12"]
    }
  },
  "orientation": "either",
  "maxRows": 50,
  "flowCollectors": ["162"],
  "orderBy": "TOTAL_BYTES",
  "excludeBpsPps": true,
  "excludeOthers": true,
  "excludeCounts": false,
  "defaultColumns": true
}

```

Response

## HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

Type: application/json

Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      },
      "required": [
        "queryId",
        "status"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

Example:

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

## HTTP status code 400

Bad or invalid request

## HTTP status code 401

Expired or invalid token. Client should re-authenticate.

## HTTP status code 403

Forbidden.

## HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

## HTTP status code 500

Internal Server error.

/tenants/{tenantId}/flow-reports/top-protocols/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-protocols/queries/{queryId}

## Request

### URI Parameters

- **tenantId**: required (string)
- **queryId**: required (string)

### Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZCIsImNpdWUiOiJkaW50IiwiaWF0IjoiMjAxNi0xMi0xNSAwOjowOjAwLm55ZXQ
```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results SHOULD be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

Type: application/json

#### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      },
      "required": [
        "queryId",
        "status"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

#### Example:

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403



- o **hostServerBytes**: The number of bytes used by the host as a server.
- o **hostServerPackets**: The number of packets used by the host as a server.
- o **peers**: The number of "Other Hosts" (i.e.) hosts on the other side of the conversation.
- o **peerFlows**: The number of flows used by "Other Hosts"(peers).
- o **peerConnections**: Number of TCP connections used by the "Other Hosts"(peers).
- o **peerClients**: The number of active client hosts among the "Other Hosts"(peers).
- o **peerClientBytes**: The number of bytes used by client hosts among the "Other Hosts"(peers).
- o **peerClientPackets**: The number of packets used by client hosts among the "Other Hosts"(peers).
- o **peerServers**: The number of active server hosts among the "Other Hosts"(peers).
- o **peerServerBytes**: The number of bytes used by server hosts among the "Other Hosts"(peers).
- o **peerServerPackets**: The number of packets used by server hosts among the "Other Hosts" (peers).
- o **protocolNumber**: Provides the number of the protocol.
- o **protocol**: The protocol that the corresponding flow was using.
- o **hostRole**: The role of the hosts selected in the Filter.
- o **hostBytes**: The number of bytes used by the host.
- o **hostPackets**: The number of packets used by the host.
- o **peerBytes**: The number of bytes used by "Other Hosts"(peers).
- o **peerPackets**: The number of packets used by "Other Hosts(peers).
- o **peerRole**: The role of the "Other Hosts"(peers).
- o **hostBytesRatio**: The ratio, in percent, of host bytes to the bytes used.
- o **peerBytesRatio**: The ratio, in percent, of peer bytes to the bytes used.
- o **clientBytesRatio**: The ratio, in percent, of client bytes to the bytes used.
- o **serverBytesRatio**: The ratio, in percent, of server bytes to the bytes used.

### Body

Type: application/json

### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "summary": {
          "type": "object",
          "properties": {
            "deviceId": {
              "type": "integer"
            },
            "bytes": {
              "type": "integer"
            },
            "percent": {
              "type": "integer"
            },
            "records": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "packets": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "connections": {
              "type": "integer"
            },
            "trafficRateMin": {
              "type": "integer"
            },
            "trafficRateMax": {
              "type": "integer"
            },
            "trafficRateAvg": {
              "type": "integer"
            },
            "trafficRate95th": {
              "type": "integer"
            },
            "packetRateMin": {
              "type": "integer"
            },
            "packetRateMax": {
```

```
    "type": "integer"
  },
  "packetRateAvg": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "integer"
  },
  "clientBytesRatio": {
    "type": "number"
  },
  "hosts": {
    "type": "integer"
  },
  "hostFlows": {
    "type": "integer"
  },
  "hostBytes": {
    "type": "integer"
  },
  "hostConnections": {
    "type": "integer"
  },
  "hostClients": {
    "type": "integer"
  },
  "hostClientBytes": {
    "type": "integer"
  },
  "hostClientPackets": {
    "type": "integer"
  },
  "hostBytesRatio": {
    "type": "integer"
  },
  "hostServers": {
    "type": "integer"
  },
  "hostServerBytes": {
    "type": "integer"
  },
  "hostPackets": {
    "type": "integer"
  },
  "hostServerPackets": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerBytesRatio": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peerServerBytes": {
    "type": "integer"
  }
```



```
    },
    "peerServerPackets": {
      "type": "integer"
    },
    },
    "serverBytesRatio": {
      "type": "number"
    },
    },
    "hostRole": {
      "type": "string"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "count",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ],
  "type": "object"
},
"others": {
  "type": "object",
  "properties": {
    "bytes": {
      "type": "integer"
    },
    "clientBytesRatio": {
      "type": "number"
    },
    },
    "connections": {
      "type": "integer"
    },
    },
    "count": {
      "type": "integer"
    },
    },
    "deviceId": {
      "type": "integer"
    },
    },
    "flows": {
      "type": "integer"
    },
    },
    "hostBytes": {
      "type": "integer"
    }
  }
}
```

```
    },
    "hostBytesRatio": {
      "type": "number"
    },
    "hostClientBytes": {
      "type": "integer"
    },
    "hostClientPackets": {
      "type": "integer"
    },
    "hostClients": {
      "type": "integer"
    },
    "hostConnections": {
      "type": "integer"
    },
    "hostFlows": {
      "type": "integer"
    },
    "hostPackets": {
      "type": "integer"
    },
    "hostRole": {
      "type": "string"
    },
    "hostServerBytes": {
      "type": "integer"
    },
    "hostServerPackets": {
      "type": "integer"
    },
    "hostServers": {
      "type": "integer"
    },
    "hosts": {
      "type": "integer"
    },
    "packetRate95th": {
      "type": "integer"
    },
    "packetRateAvg": {
      "type": "integer"
    },
    "packetRateMax": {
      "type": "integer"
    },
    "packetRateMin": {
      "type": "integer"
    },
    "packets": {
      "type": "integer"
    },
    "peerBytes": {
      "type": "integer"
    },
    "peerBytesRatio": {
      "type": "number"
    },
    "peerClientBytes": {
      "type": "integer"
    },
    "peerClientPackets": {
      "type": "integer"
    },
    "peerClients": {
      "type": "integer"
    },
    "peerConnections": {
      "type": "integer"
    },
    "peerFlows": {
      "type": "integer"
    },
    "peerPackets": {
      "type": "integer"
    },
    "peerRole": {
      "type": "string"
    },
    "peerServerBytes": {
      "type": "integer"
    }
  }
}
```

```

    },
    "peerServerPackets": {
      "type": "integer"
    },
    "peerServers": {
      "type": "integer"
    },
    "peers": {
      "type": "integer"
    },
    "percent": {
      "type": "number"
    },
    "records": {
      "type": "integer"
    },
    "serverBytesRatio": {
      "type": "number"
    },
    "trafficRate95th": {
      "type": "integer"
    },
    "trafficRateAvg": {
      "type": "integer"
    },
    "trafficRateMax": {
      "type": "integer"
    },
    "trafficRateMin": {
      "type": "integer"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "count",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ],
  "type": "object"
},
"results": {
  "type": "array",

```

```
"items": {
  "type": "object",
  "properties": {
    "bytes": {
      "type": "integer"
    },
    "clientBytesRatio": {
      "type": "number"
    },
    "connections": {
      "type": "integer"
    },
    "deviceId": {
      "type": "integer"
    },
    "flows": {
      "type": "integer"
    },
    "hostBytes": {
      "type": "integer"
    },
    "hostBytesRatio": {
      "type": "number"
    },
    "hostClientBytes": {
      "type": "integer"
    },
    "hostClientPackets": {
      "type": "integer"
    },
    "hostClients": {
      "type": "integer"
    },
    "hostConnections": {
      "type": "integer"
    },
    "hostFlows": {
      "type": "integer"
    },
    "hostPackets": {
      "type": "integer"
    },
    "hostRole": {
      "type": "string"
    },
    "hostServerBytes": {
      "type": "integer"
    },
    "hostServerPackets": {
      "type": "integer"
    },
    "hostServers": {
      "type": "integer"
    },
    "hosts": {
      "type": "integer"
    },
    "packetRate95th": {
      "type": "integer"
    },
    "packetRateAvg": {
      "type": "integer"
    },
    "packetRateMax": {
      "type": "integer"
    },
    "packetRateMin": {
      "type": "integer"
    },
    "packets": {
      "type": "integer"
    },
    "peerBytes": {
      "type": "integer"
    },
    "peerBytesRatio": {
      "type": "number"
    },
    "peerClientBytes": {
      "type": "integer"
    },
    "peerClientPackets": {
```

```
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "protocol": {
    "type": "string"
  },
  "protocolNumber": {
    "type": "integer"
  },
  "rank": {
    "type": "integer"
  },
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "trafficRate95th": {
    "type": "integer"
  },
  "trafficRateAvg": {
    "type": "integer"
  },
  "trafficRateMax": {
    "type": "integer"
  },
  "trafficRateMin": {
    "type": "integer"
  }
},
"required": [
  "hostPackets",
  "hostServerBytes",
  "protocol",
  "peerFlows",
  "packetRateMin",
  "packets",
  "rank",
  "connections",
  "trafficRateMin",
  "hostBytes",
  "hostClientPackets",
  "hostBytesRatio",
  "hostClientBytes",
  "percent",
  "peerClientPackets",
  "trafficRateMax",
  "trafficRate95th",
  "clientBytesRatio",
  "hostFlows",
  "packetRateMax",
  "peerBytes",
```



```
"hostPackets": 0,
"peerBytes": 1188739937679,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 37.97566298533009,
"serverBytesRatio": 62.02433701466991,
"hostRole": "CLIENT_AND_SERVER"
},
"others": {
  "deviceId": 622,
  "percent": 8.953520673942462,
  "records": 3433332,
  "count": 237733,
  "bytes": 212868152159,
  "packets": 185557066,
  "flows": 957319,
  "connections": 3020024,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 237733,
  "hostFlows": 343269,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 54755140776,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 34553342244,
  "hostServerPackets": 0,
  "peers": 489948,
  "peerFlows": 614095,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 118072738050,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 5486931089,
  "peerServerPackets": 0,
  "hostBytes": 89308483020,
  "hostPackets": 0,
  "peerBytes": 123559669139,
  "peerPackets": 0,
  "peerRole": "CLIENT_AND_SERVER",
  "hostBytesRatio": 41.9548354764182,
  "peerBytesRatio": 58.0451645235818,
  "clientBytesRatio": 81.19010621039625,
  "serverBytesRatio": 18.80989378960375,
  "hostRole": "CLIENT_AND_SERVER"
},
"results": [{
  "deviceId": 622,
  "rank": 1,
  "percent": 19.515433006646784,
  "records": 2437,
  "bytes": 463975492322,
  "packets": 403537191,
  "flows": 54,
  "connections": 0,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 1,
  "hostFlows": 41,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 2922061722,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 1843321,
  "hostServerPackets": 0,
```

```
"peers": 8,
"peerFlows": 13,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 770579,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 461050816700,
"peerServerPackets": 0,
"protocolNumber": 1,
"protocol": "ICMP",
"hostBytes": 2923905043,
"hostPackets": 0,
"peerBytes": 461051587279,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 0.6301852342172426,
"peerBytesRatio": 99.36981476578276,
"clientBytesRatio": 0.6299540276087574,
"serverBytesRatio": 99.37004597239124,
"hostRole": "CLIENT_AND_SERVER"
}, {
"deviceId": 622,
"rank": 2,
"percent": 19.514754852347053,
"records": 2145,
"bytes": 463959369340,
"packets": 403516018,
"flows": 24,
"connections": 402,
"trafficRateMin": 0.0,
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 1,
"hostFlows": 10,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 1764216,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 461048838949,
"hostServerPackets": 0,
"peers": 5,
"peerFlows": 14,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 2905101992,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 3664183,
"peerServerPackets": 0,
"protocolNumber": 6,
"protocol": "TCP",
"hostBytes": 461050603165,
"hostPackets": 0,
"peerBytes": 2908766175,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 99.37305583910552,
"peerBytesRatio": 0.6269441608944833,
"clientBytesRatio": 0.6265346493886154,
"serverBytesRatio": 99.37346535061138,
"hostRole": "CLIENT_AND_SERVER"
}, {
"deviceId": 622,
"rank": 3,
"percent": 11.84168661371347,
"records": 1161,
"bytes": 281533716144,
"packets": 89807482,
"flows": 134,
"connections": 328,
"trafficRateMin": 0.0,
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
```



```

"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 1,
"hostFlows": 104,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 73678643668,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 101668268769,
"hostServerPackets": 0,
"peers": 5,
"peerFlows": 30,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 1465251170,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 104721552537,
"peerServerPackets": 0,
"protocolNumber": 17,
"protocol": "UDP",
"hostBytes": 175346912437,
"hostPackets": 0,
"peerBytes": 106186803707,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 62.282739999536275,
"peerBytesRatio": 37.717260000463725,
"clientBytesRatio": 26.69090433188652,
"serverBytesRatio": 73.30909566811347,
"hostRole": "CLIENT_AND_SERVER"
  }
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/flow-reports/top-hosts/queries

POST

Initiate Search

POST

/tenants/{tenantId}/flow-reports/top-hosts/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included

- **excludes:** List of predefined application ids to be excluded
    - **direction:** Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols:** Port Protocols used or unused in the flow.
      - **includes:** List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes:** List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp

The following protocols are allowed to include or exclude

      - tcp: Should have a valid port number
      - udp: Should have a valid port number
      - icmp: Should not have a port number
  - **subject:**
    - **tags:** Specify the tags involved or not involved in the flow
      - **includes:** List of tag ids to be included.
      - **excludes:** List of tag ids to be excluded.
    - **ipAddresses:** Specify the IP addresses or range of IP addresses involved or not involved in the flow.
      - **includes:** List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes:** List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **peer:**
    - **tags:** Specify the peer tags involved or not involved in the flow
      - **includes:** List of tag ids to be included.
      - **excludes:** List of tag ids to be excluded.
    - **ipAddresses:** List of IP addresses to be included or excluded.
      - **includes:** List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes:** List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **orientation:** Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
    - EITHER (Default)
    - CLIENT
    - SERVER
  - **maxRows:** The maximum no. of records to be returned. The maximum value for this field is 5000.
    - Default: 50
  - **flowCollectors:** List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.
  - **orderBy:** The order based on which the records will be retrieved and sorted by (i.e. Bytes or Packets or Flows or TCP Connection). The value must be one of the following:
    - TOTAL\_BYTES (Default)
    - TOTAL\_PACKETS
    - TOTAL\_FLOWS
    - TOTAL\_CONNECTIONS
  - **standardOptions:** Flag to return the default columns.
    - Value should be either true or false
    - Incase If the value is set as "true" or set to default, then implicitly fields part of AdvancedOptions will be set to its default value.
    - Default value: true
  - **advanceOptions**
    - **excludeBpsPps:** Flag to excludes bps/pps values.
      - Value should be either true or false
      - Default value: true
    - **excludeOthers:** Flag to excludes Other Records
      - Value should be either true or false
      - Default value: true
    - **excludeCounts:** Flag to excludes Counts
      - Value should be either true or false
      - Default value: false

## Request

### URI Parameters

- **tenantId:** *required (string)*

### Headers

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMjAxNi0xMi0xNSUwOTIyOjA5OjA5InQ9PS09
```

**Body**

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "searchName": {
      "type": "string"
    },
    "startTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "endTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "connection": {
      "type": "object",
      "properties": {
        "applications": {
          "type": "object",
          "properties": {
            "includes": {
              "type": "array",
              "items": {}
            },
            "excludes": {
              "type": "array",
              "items": {}
            }
          }
        },
        "required": [
          "includes",
          "excludes"
        ]
      }
    },
    "direction": {
      "type": "string",
      "enum": ["INBOUND_PLUS_OUTBOUND", "INBOUND", "OUTBOUND", "WITHIN"],
      "default": "INBOUND_PLUS_OUTBOUND"
    },
    "portProtocols": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "required": [
      "direction"
    ]
  },
  "subject": {
    "type": "object",
    "properties": {
      "tags": {
        "type": "object",
        "properties": {
          "includes": {
            "type": "array"
          }
        }
      }
    }
  }
}
```

```

    type: "array",
    "items": {}
  },
  "excludes": {
    "type": "array",
    "items": {}
  }
},
"required": [
  "includes",
  "excludes"
]
},
"ipAddresses": {
  "type": "object",
  "properties": {
    "includes": {
      "type": "array",
      "items": {}
    },
    "excludes": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "includes",
    "excludes"
  ]
},
"required": []
},
"peer": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    }
  },
  "required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 400000
}

```

```

},
"flowCollectors": {
  "type": "array",
  "items": {}
},
},
"orderBy": {
  "type": "string",
  "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
  "default": "TOTAL_BYTES"
},
},
"excludeBpsPps": {
  "type": "boolean",
  "default": true
},
},
"excludeOthers": {
  "type": "boolean",
  "default": true
},
},
"excludeCounts": {
  "type": "boolean",
  "default": false
},
},
"defaultColumns": {
  "type": "boolean",
  "default": true
}
}
},
"required": [
  "startTime",
  "endTime"
]
}
}

```

**Example:**

```

{
  "searchName": "Top Reports on 3/1/2017 at 12:36 PM",
  "startTime": "2017-03-10T00:00:00.000",
  "endTime": "2017-03-10T00:05:00.000",
  "connection": {
    "applications": {
      "includes": [103119],
      "excludes": [2000]
    },
    "direction": "INBOUND_PLUS_OUTBOUND",
    "portProtocols": {
      "includes": ["70/tcp"],
      "excludes": ["80/tcp"]
    }
  },
  "subject": {
    "tags": {
      "includes": [63000],
      "excludes": [63006]
    },
    "ipAddresses": {
      "includes": ["10.20.11.11"],
      "excludes": []
    }
  },
  "peer": {
    "tags": {
      "includes": [47],
      "excludes": [60000]
    },
    "ipAddresses": {
      "includes": [],
      "excludes": ["10.20.11.12"]
    }
  },
  "orientation": "either",
  "maxRows": 50,
  "flowCollectors": ["162"],
  "orderBy": "TOTAL_BYTES",
  "excludeBpsPps": true,
  "excludeOthers": true,
  "excludeCounts": false,
  "defaultColumns": true
}

```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

Type: application/json

### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      },
      "required": [
        "queryId",
        "status"
      ]
    },
    "required": [
      "data"
    ]
  }
}
```

### Example:

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/flow-reports/top-hosts/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-hosts/queries/{queryId}

## Get the status of the search

### Request

#### URI Parameters

- **tenantId**: required (string)
- **queryId**: required (string)

#### Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZC54ZW50LXBpZW50IiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjE2IiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

### Response

#### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

#### Body

**Type:** application/json

#### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      },
      "required": [
        "queryId",
        "status"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

#### Example:

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.





- **hostServerBytes**: The number of bytes used by the host as a server
- **hostServerPackets**: The number of packets used by the host as a server
- **peers**: The number of "Other Hosts" (i.e.) hosts on the other side of the conversation
- **peerFlows**: The number of flows used by "Other Hosts"(peers)
- **peerConnections**: Number of TCP connections used by the "Other Hosts"(peers)
- **peerClients**: The number of active client hosts among the "Other Hosts"(peers)
- **peerClientBytes**: The number of bytes used by client hosts among the "Other Hosts"(peers)
- **peerClientPackets**: The number of packets used by client hosts among the "Other Hosts"(peers)
- **peerServers**: The number of active server hosts among the "Other Hosts"(peers)
- **peerServerBytes**: The number of bytes used by server hosts among the "Other Hosts"(peers)
- **peerServerPackets**: The number of packets used by server hosts among the "Other Hosts" (peers)
- **protocolNumber**: Provides number of the protocol
- **host**: It contains the following fields
  - **country**: The country in which the host is located
  - **ipAddress**: The IP address assigned to the host
  - **name**: The name assigned to the host
  - **hostGroupIds**: The ids of the host group(s) to which the host belongs
- **protocol**: The protocol that the corresponding flow was using
- **hostRole**: The role of the hosts selected in the Filter
- **hostBytes**: The number of bytes used by the host
- **hostPackets**: The number of packets used by the host
- **peerBytes**: The number of bytes used by "Other Hosts"(peers)
- **peerPackets**: The number of packets used by "Other Hosts(peers)
- **peerRole**: The role of the "Other Hosts"(peers)
- **hostBytesRatio**: The ratio, in percent, of host bytes to the bytes used
- **peerBytesRatio**: The ratio, in percent, of peer bytes to the bytes used
- **clientBytesRatio**: The ratio, in percent, of client bytes to the bytes used
- **serverBytesRatio**: The ratio, in percent, of server bytes to the bytes used

#### Body

Type: application/json

#### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "others": {
          "type": "object",
          "properties": {
            "bytes": {
              "type": "integer"
            },
            "clientBytesRatio": {
              "type": "number"
            },
            "connections": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "deviceId": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "hostBytes": {
              "type": "integer"
            },
            "hostBytesRatio": {
              "type": "number"
            },
            "hostClientBytes": {
              "type": "integer"
            },
            "hostClientPackets": {
              "type": "integer"
            },
            "hostClients": {
              "type": "integer"
            },
            "hostConnections": {
```

```
    "type": "integer"
  },
  "hostFlows": {
    "type": "integer"
  },
  "hostPackets": {
    "type": "integer"
  },
  "hostRole": {
    "type": "string"
  },
  "hostServerBytes": {
    "type": "integer"
  },
  "hostServerPackets": {
    "type": "integer"
  },
  "hostServers": {
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "number"
  },
  "packetRateAvg": {
    "type": "number"
  },
  "packetRateMax": {
    "type": "number"
  },
  "packetRateMin": {
    "type": "number"
  },
  "packets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerBytesRatio": {
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "records": {
    "type": "integer"
  }
}
```

```

    },
    "serverBytesRatio": {
      "type": "number"
    },
    "trafficRate95th": {
      "type": "number"
    },
    "trafficRateAvg": {
      "type": "number"
    },
    "trafficRateMax": {
      "type": "number"
    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "count",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ]
},
"results": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "bytes": {
        "type": "integer"
      },
      "clientBytesRatio": {
        "type": "number"
      },
      "connections": {
        "type": "integer"
      },
      "deviceId": {
        "type": "integer"
      },
      "flows": {

```

```
flows : {
  "type": "integer"
},
"host": {
  "type": "object",
  "properties": {
    "country": {
      "type": "string"
    },
    "hostGroupIds": {
      "type": "array",
      "items": {
        "type": "integer"
      }
    },
    "ipAddress": {
      "type": "string"
    },
    "name": {
      "type": "string"
    }
  },
  "required": [
    "country",
    "ipAddress",
    "name",
    "hostGroupIds"
  ]
},
"hostBytes": {
  "type": "integer"
},
"hostBytesRatio": {
  "type": "number"
},
"hostClientBytes": {
  "type": "integer"
},
"hostClientPackets": {
  "type": "integer"
},
"hostClients": {
  "type": "integer"
},
"hostConnections": {
  "type": "integer"
},
"hostFlows": {
  "type": "integer"
},
"hostPackets": {
  "type": "integer"
},
"hostRole": {
  "type": "string"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
,
```

```
    },
    "peerBytes": {
      "type": "integer"
    },
    "peerBytesRatio": {
      "type": "number"
    },
    "peerClientBytes": {
      "type": "integer"
    },
    "peerClientPackets": {
      "type": "integer"
    },
    "peerClients": {
      "type": "integer"
    },
    "peerConnections": {
      "type": "integer"
    },
    "peerFlows": {
      "type": "integer"
    },
    "peerPackets": {
      "type": "integer"
    },
    "peerRole": {
      "type": "string"
    },
    "peerServerBytes": {
      "type": "integer"
    },
    "peerServerPackets": {
      "type": "integer"
    },
    "peerServers": {
      "type": "integer"
    },
    "peers": {
      "type": "integer"
    },
    "percent": {
      "type": "number"
    },
    "rank": {
      "type": "integer"
    },
    "records": {
      "type": "integer"
    },
    "serverBytesRatio": {
      "type": "number"
    },
    "trafficRate95th": {
      "type": "number"
    },
    "trafficRateAvg": {
      "type": "number"
    },
    "trafficRateMax": {
      "type": "number"
    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "rank",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
```

```
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "host",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ]
}
},
"summary": {
  "type": "object",
  "properties": {
    "bytes": {
      "type": "integer"
    },
    "clientBytesRatio": {
      "type": "number"
    },
    "connections": {
      "type": "integer"
    },
    "count": {
      "type": "integer"
    },
    "deviceId": {
      "type": "integer"
    },
    "flows": {
      "type": "integer"
    },
    "hostBytes": {
      "type": "integer"
    },
    "hostBytesRatio": {
      "type": "number"
    },
    "hostClientBytes": {
      "type": "integer"
    },
    "hostClientPackets": {
      "type": "integer"
    },
    "hostClients": {
      "type": "integer"
    },
    "hostConnections": {
      "type": "integer"
    },
    "hostFlows": {
      "type": "integer"
    },
    "hostPackets": {
      "type": "integer"
    },
    "hostRole": {
      "type": "string"
    },
    "hostServerBytes": {
      "type": "integer"
    }
  }
}
```

```
},
"hostServerPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
"peerFlows": {
  "type": "integer"
},
"peerPackets": {
  "type": "integer"
},
"peerRole": {
  "type": "string"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"percent": {
  "type": "number"
},
"records": {
  "type": "integer"
},
"serverBytesRatio": {
  "type": "number"
},
"trafficRate95th": {
  "type": "number"
},
"trafficRateAvg": {
  "type": "number"
},
"trafficRateMax": {
  "type": "number"
}
}
```

```

    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "count",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ]
}
},
"required": [
  "others",
  "results",
  "summary"
]
}
},
"required": [
  "data"
]
}
}

```

**Example:**

```

{
  "data" : {
    "summary": {
      "deviceId": 622,
      "percent": 100.0,
      "records": 4161574,
      "count": 237759,
      "bytes": 2377479875358,
      "packets": 1649544432,
      "flows": 1240477,
      "connections": 3151300,
      "trafficRateMin": 0.0,
      "trafficRateMax": 0.0,
      "trafficRateAvg": 0.0,
      "trafficRate95th": 0.0,

```



```
"packetRateMin": 0.0,
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 237759,
"hostFlows": 620261,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 451431872505,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 737308065174,
"hostServerPackets": 0,
"peers": 644617,
"peerFlows": 620261,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 451431872505,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 737308065174,
"peerServerPackets": 0,
"hostBytes": 1188739937679,
"hostPackets": 0,
"peerBytes": 1188739937679,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 37.97566298533009,
"serverBytesRatio": 62.02433701466991,
"hostRole": "CLIENT_AND_SERVER"
},
"others": {
  "deviceId": 622,
  "percent": 8.953520673942462,
  "records": 3433332,
  "count": 237733,
  "bytes": 212868152159,
  "packets": 185557066,
  "flows": 957319,
  "connections": 3020024,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 237733,
  "hostFlows": 343269,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 54755140776,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 34553342244,
  "hostServerPackets": 0,
  "peers": 489948,
  "peerFlows": 614095,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 118072738050,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 5486931089,
  "peerServerPackets": 0,
  "hostBytes": 89308483020,
  "hostPackets": 0,
  "peerBytes": 123559669139,
  "peerPackets": 0,
  "peerRole": "CLIENT_AND_SERVER",
  "hostBytesRatio": 41.9548354764182,
  "peerBytesRatio": 58.0451645235818,
  "clientBytesRatio": 81.19010621039625,
  "serverBytesRatio": 18.80989378960375,
  "hostRole": "CLIENT_AND_SERVER"
},
"results": [{
  "deviceId": 622,
  "rank": 1,
```

```
"percent": 19.515433006646784,
"records": 2437,
"bytes": 463975492322,
"packets": 403537191,
"flows": 54,
"connections": 0,
"trafficRateMin": 0.0,
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 1,
"host": {
  "ipAddress": "10.203.7.10",
  "country": "XR",
  "hostGroupIds": [65534],
  "name": "Catch All"
},
"hostFlows": 41,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 2922061722,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 1843321,
"hostServerPackets": 0,
"peers": 8,
"peerFlows": 13,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 770579,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 461050816700,
"peerServerPackets": 0,
"hostBytes": 2923905043,
"hostPackets": 0,
"peerBytes": 461051587279,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 0.6301852342172426,
"peerBytesRatio": 99.36981476578276,
"clientBytesRatio": 0.6299540276087574,
"serverBytesRatio": 99.37004597239124,
"hostRole": "CLIENT_AND_SERVER"
}, {
  "deviceId": 622,
  "rank": 2,
  "percent": 19.514754852347053,
  "records": 2145,
  "bytes": 463959369340,
  "packets": 403516018,
  "flows": 24,
  "connections": 402,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 1,
  "host": {
    "ipAddress": "10.203.7.10",
    "country": "XR",
    "hostGroupIds": [65534],
    "name": "Catch All"
  },
  "hostFlows": 10,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 1764216,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 461048838949,
  "hostServerPackets": 0,
  "peers": 5,
  "peerFlows": 14.
```

```

    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 2905101992,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 3664183,
    "peerServerPackets": 0,
    "hostBytes": 461050603165,
    "hostPackets": 0,
    "peerBytes": 2908766175,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 99.37305583910552,
    "peerBytesRatio": 0.6269441608944833,
    "clientBytesRatio": 0.6265346493886154,
    "serverBytesRatio": 99.37346535061138,
    "hostRole": "CLIENT_AND_SERVER"
  }, {
    "deviceId": 622,
    "rank": 3,
    "percent": 11.84168661371347,
    "records": 1161,
    "bytes": 281533716144,
    "packets": 89807482,
    "flows": 134,
    "connections": 328,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 1,
    "host": {
      "ipAddress": "10.203.7.10",
      "country": "XR",
      "hostGroupIds": [65534],
      "name": "Catch All"
    },
    "hostFlows": 104,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 73678643668,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 101668268769,
    "hostServerPackets": 0,
    "peers": 5,
    "peerFlows": 30,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 1465251170,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 104721552537,
    "peerServerPackets": 0,
    "hostBytes": 175346912437,
    "hostPackets": 0,
    "peerBytes": 106186803707,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 62.282739999536275,
    "peerBytesRatio": 37.717260000463725,
    "clientBytesRatio": 26.69090433188652,
    "serverBytesRatio": 73.30909566811347,
    "hostRole": "CLIENT_AND_SERVER"
  }
]
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/flow-reports/top-peers/queries

POST

Initiate Search

POST

/tenants/{tenantId}/flow-reports/top-peers/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included
      - **excludes**: List of predefined application ids to be excluded
    - **direction**: Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols**: Port Protocols used or unused in the flow.
      - **includes**: List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes**: List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp

The following protocols are allowed to include or exclude

      - tcp: Should have a valid port number
      - udp: Should have a valid port number
      - icmp: Should not have a port number
  - **subject**:
    - **tags**: Specify the tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: Specify the IP addresses or range of IP addresses involved or not involved in the flow.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **peer**:
    - **tags**: Specify the peer tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: List of IP addresses to be included or excluded.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **orientation**: Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
    - EITHER (Default)
    - CLIENT
    - SERVER
  - **maxRows**: The maximum no. of records to be returned. The maximum value for this field is 5000.
    - Default: 50
  - **flowCollectors**: List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.



```
"direction": {
  "type": "string",
  "enum": ["INBOUND_PLUS_OUTBOUND", "INBOUND", "OUTBOUND", "WITHIN"],
  "default": "INBOUND_PLUS_OUTBOUND"
},
"portProtocols": {
  "type": "object",
  "properties": {
    "includes": {
      "type": "array",
      "items": {}
    },
    "excludes": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "includes",
    "excludes"
  ]
},
"required": [
  "direction"
]
},
"subject": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    }
  },
  "required": []
},
"peer": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      }
    }
  },
  "required": []
},
```

```

    "required": [
      "includes",
      "excludes"
    ]
  },
  "ipAddresses": {
    "type": "object",
    "properties": {
      "includes": {
        "type": "array",
        "items": {}
      },
      "excludes": {
        "type": "array",
        "items": {}
      }
    },
    "required": [
      "includes",
      "excludes"
    ]
  },
  "required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 400000
},
"flowCollectors": {
  "type": "array",
  "items": {}
},
"orderBy": {
  "type": "string",
  "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
  "default": "TOTAL_BYTES"
},
"excludeBpsPps": {
  "type": "boolean",
  "default": true
},
"excludeOthers": {
  "type": "boolean",
  "default": true
},
"excludeCounts": {
  "type": "boolean",
  "default": false
},
"defaultColumns": {
  "type": "boolean",
  "default": true
}
},
"required": [
  "startTime",
  "endTime"
]
}

```

**Example:**

```
{
  "searchName": "Top Reports on 3/1/2017 at 12:36 PM",
  "startTime": "2017-03-10T00:00:00.000",
  "endTime": "2017-03-10T00:05:00.000",
  "connection": {
    "applications": {
      "includes": [103119],
      "excludes": [2000]
    },
    "direction": "INBOUND_PLUS_OUTBOUND",
    "portProtocols": {
      "includes": ["70/tcp"],
      "excludes": ["80/tcp"]
    }
  },
  "subject": {
    "tags": {
      "includes": [63000],
      "excludes": [63006]
    },
    "ipAddresses": {
      "includes": ["10.20.11.11"],
      "excludes": []
    }
  },
  "peer": {
    "tags": {
      "includes": [47],
      "excludes": [60000]
    },
    "ipAddresses": {
      "includes": [],
      "excludes": ["10.20.11.12"]
    }
  },
  "orientation": "either",
  "maxRows": 50,
  "flowCollectors": ["162"],
  "orderBy": "TOTAL_BYTES",
  "excludeBpsPps": true,
  "excludeOthers": true,
  "excludeCounts": false,
  "defaultColumns": true
}
```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

**Type:** application/json

**Schema:**



```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      }
    },
    "required": [
      "queryId",
      "status"
    ]
  }
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/flow-reports/top-peers/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-peers/queries/{queryId}

Get the status of the search

**Request**

**URI Parameters**

- **tenantId:** *required (string)*
- **queryId:** *required (string)*

**Headers**

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

**Example:**





- **name**: The name assigned to the peer
  - **hostGroupIds**: The ids of the host group(s) to which the peer belongs
- **protocol**: The protocol that the corresponding flow was using
  - **hostRole**: The role of the hosts selected in the Filter
  - **hostBytes**: The number of bytes used by the host
  - **hostPackets**: The number of packets used by the host
  - **peerBytes**: The number of bytes used by "Other Hosts"(peers)
  - **peerPackets**: The number of packets used by "Other Hosts(peers)
  - **peerRole**: The role of the "Other Hosts"(peers)
  - **hostBytesRatio**: The ratio, in percent, of host bytes to the bytes used
  - **peerBytesRatio**: The ratio, in percent, of peer bytes to the bytes used
  - **clientBytesRatio**: The ratio, in percent, of client bytes to the bytes used
  - **serverBytesRatio**: The ratio, in percent, of server bytes to the bytes used

## Body

Type: application/json

## Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "others": {
          "type": "object",
          "properties": {
            "bytes": {
              "type": "integer"
            },
            "clientBytesRatio": {
              "type": "number"
            },
            "connections": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "deviceId": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "hostBytes": {
              "type": "integer"
            },
            "hostBytesRatio": {
              "type": "number"
            },
            "hostClientBytes": {
              "type": "integer"
            },
            "hostClientPackets": {
              "type": "integer"
            },
            "hostClients": {
              "type": "integer"
            },
            "hostConnections": {
              "type": "integer"
            },
            "hostFlows": {
              "type": "integer"
            },
            "hostPackets": {
              "type": "integer"
            },
            "hostRole": {
              "type": "string"
            },
            "hostServerBytes": {
              "type": "integer"
            },
            "hostServerPackets": {
              "type": "integer"
            }
          }
        }
      }
    }
  }
}
```

```
,
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
"peerFlows": {
  "type": "integer"
},
"peerPackets": {
  "type": "integer"
},
"peerRole": {
  "type": "string"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"percent": {
  "type": "number"
},
"records": {
  "type": "integer"
},
"serverBytesRatio": {
  "type": "number"
},
"trafficRate95th": {
  "type": "number"
},
"trafficRateAvg": {
  "type": "number"
},
"trafficRateMax": {
  "type": "number"
},
"trafficRateMin": {
  "type": "number"
}
}
```

```

    },
    "required": [
      "hostPackets",
      "hostServerBytes",
      "peerFlows",
      "packetRateMin",
      "packets",
      "connections",
      "trafficRateMin",
      "hostBytes",
      "hostClientPackets",
      "hostBytesRatio",
      "hostClientBytes",
      "percent",
      "peerClientPackets",
      "trafficRateMax",
      "trafficRate95th",
      "clientBytesRatio",
      "hostFlows",
      "packetRateMax",
      "peerBytes",
      "serverBytesRatio",
      "peerConnections",
      "flows",
      "records",
      "hostServerPackets",
      "peerPackets",
      "deviceId",
      "hostServers",
      "peerServers",
      "count",
      "peers",
      "packetRate95th",
      "trafficRateAvg",
      "peerBytesRatio",
      "peerServerBytes",
      "hostConnections",
      "peerClients",
      "bytes",
      "peerClientBytes",
      "peerServerPackets",
      "hosts",
      "hostClients",
      "peerRole",
      "hostRole",
      "packetRateAvg"
    ]
  },
  "results": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "bytes": {
          "type": "integer"
        },
        "clientBytesRatio": {
          "type": "number"
        },
        "connections": {
          "type": "integer"
        },
        "deviceId": {
          "type": "integer"
        },
        "flows": {
          "type": "integer"
        },
        "hostBytes": {
          "type": "integer"
        },
        "hostBytesRatio": {
          "type": "number"
        },
        "hostClientBytes": {
          "type": "integer"
        },
        "hostClientPackets": {
          "type": "integer"
        },
        "hostClients": {
          "type": "integer"
        },

```

```
},
"hostConnections": {
  "type": "integer"
},
"hostFlows": {
  "type": "integer"
},
"hostPackets": {
  "type": "integer"
},
"hostRole": {
  "type": "string"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
"peer": {
  "type": "object",
  "properties": {
    "country": {
      "type": "string"
    },
    "hostGroupIds": {
      "type": "array",
      "items": {
        "type": "integer"
      }
    },
    "ipAddress": {
      "type": "string"
    },
    "name": {
      "type": "string"
    }
  },
  "required": [
    "country",
    "ipAddress",
    "name",
    "hostGroupIds"
  ]
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerConnections": {
```

```
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "rank": {
    "type": "integer"
  },
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "trafficRate95th": {
    "type": "number"
  },
  "trafficRateAvg": {
    "type": "number"
  },
  "trafficRateMax": {
    "type": "number"
  },
  "trafficRateMin": {
    "type": "number"
  }
},
"required": [
  "hostPackets",
  "hostServerBytes",
  "peerFlows",
  "packetRateMin",
  "packets",
  "rank",
  "connections",
  "trafficRateMin",
  "hostBytes",
  "hostClientPackets",
  "hostBytesRatio",
  "hostClientBytes",
  "percent",
  "peerClientPackets",
  "trafficRateMax",
  "trafficRate95th",
  "clientBytesRatio",
  "hostFlows",
  "packetRateMax",
  "peerBytes",
  "serverBytesRatio",
  "peerConnections",
  "flows",
  "records",
  "hostServerPackets",
  "peerPackets",
  "deviceId",
  "peer",
  "hostServers",
  "peerServers",
  "peers",
  "packetRate95th",
  "trafficRateAvg",
```



```
        "peerBytesRatio",
        "peerServerBytes",
        "hostConnections",
        "peerClients",
        "bytes",
        "peerClientBytes",
        "peerServerPackets",
        "hosts",
        "hostClients",
        "peerRole",
        "hostRole",
        "packetRateAvg"
    ]
}
},
"summary": {
    "type": "object",
    "properties": {
        "bytes": {
            "type": "integer"
        },
        "clientBytesRatio": {
            "type": "number"
        },
        "connections": {
            "type": "integer"
        },
        "count": {
            "type": "integer"
        },
        "deviceId": {
            "type": "integer"
        },
        "flows": {
            "type": "integer"
        },
        "hostBytes": {
            "type": "integer"
        },
        "hostBytesRatio": {
            "type": "number"
        },
        "hostClientBytes": {
            "type": "integer"
        },
        "hostClientPackets": {
            "type": "integer"
        },
        "hostClients": {
            "type": "integer"
        },
        "hostConnections": {
            "type": "integer"
        },
        "hostFlows": {
            "type": "integer"
        },
        "hostPackets": {
            "type": "integer"
        },
        "hostRole": {
            "type": "string"
        },
        "hostServerBytes": {
            "type": "integer"
        },
        "hostServerPackets": {
            "type": "integer"
        },
        "hostServers": {
            "type": "integer"
        },
        "hosts": {
            "type": "integer"
        },
        "packetRate95th": {
            "type": "number"
        },
        "packetRateAvg": {
            "type": "number"
        },
        "packetRateMax": {
```

```
    "packetRateMax": {
      "type": "number"
    },
    "packetRateMin": {
      "type": "number"
    },
    "packets": {
      "type": "integer"
    },
    "peerBytes": {
      "type": "integer"
    },
    "peerBytesRatio": {
      "type": "number"
    },
    "peerClientBytes": {
      "type": "integer"
    },
    "peerClientPackets": {
      "type": "integer"
    },
    "peerClients": {
      "type": "integer"
    },
    "peerConnections": {
      "type": "integer"
    },
    "peerFlows": {
      "type": "integer"
    },
    "peerPackets": {
      "type": "integer"
    },
    "peerRole": {
      "type": "string"
    },
    "peerServerBytes": {
      "type": "integer"
    },
    "peerServerPackets": {
      "type": "integer"
    },
    "peerServers": {
      "type": "integer"
    },
    "peers": {
      "type": "integer"
    },
    "percent": {
      "type": "number"
    },
    "records": {
      "type": "integer"
    },
    "serverBytesRatio": {
      "type": "number"
    },
    "trafficRate95th": {
      "type": "number"
    },
    "trafficRateAvg": {
      "type": "number"
    },
    "trafficRateMax": {
      "type": "number"
    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "peerBytes",
    "peerClientBytes",
    "peerConnections",
    "peerFlows",
    "peerPackets",
    "peerRole",
    "peerServerBytes",
    "peerServerPackets",
    "peerServers",
    "peers",
    "percent",
    "records",
    "serverBytesRatio",
    "trafficRate95th",
    "trafficRateAvg",
    "trafficRateMax",
    "trafficRateMin"
  ]
}
```

```

        "percent",
        "peerClientPackets",
        "trafficRateMax",
        "trafficRate95th",
        "clientBytesRatio",
        "hostFlows",
        "packetRateMax",
        "peerBytes",
        "serverBytesRatio",
        "peerConnections",
        "flows",
        "records",
        "hostServerPackets",
        "peerPackets",
        "deviceId",
        "hostServers",
        "peerServers",
        "count",
        "peers",
        "packetRate95th",
        "trafficRateAvg",
        "peerBytesRatio",
        "peerServerBytes",
        "hostConnections",
        "peerClients",
        "bytes",
        "peerClientBytes",
        "peerServerPackets",
        "hosts",
        "hostClients",
        "peerRole",
        "hostRole",
        "packetRateAvg"
    ]
}
},
"required": [
    "others",
    "results",
    "summary"
]
}
},
"required": [
    "data"
]
}
}

```

**Example:**

```

{
  "data" : {
    "summary": {
      "deviceId": 622,
      "percent": 100.0,
      "records": 4152672,
      "count": 237702,
      "bytes": 2374122533094,
      "packets": 1646930926,
      "flows": 1237059,
      "connections": 3149908,
      "trafficRateMin": 0.0,
      "trafficRateMax": 0.0,
      "trafficRateAvg": 0.0,
      "trafficRate95th": 0.0,
      "packetRateMin": 0.0,
      "packetRateMax": 0.0,
      "packetRateAvg": 0.0,
      "packetRate95th": 0.0,
      "hosts": 643957,
      "hostFlows": 618552,
      "hostConnections": 0,
      "hostClients": 0,
      "hostClientBytes": 450593175961,
      "hostClientPackets": 0,
      "hostServers": 0,
      "hostServerBytes": 736468090586,
      "hostServerPackets": 0,
      "peers": 237702,
      "peerFlows": 618552,
      "peerConnections": 0,
      "peerClients": 0,

```

```
"peerClientBytes": 450593175961,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 736468090586,
"peerServerPackets": 0,
"hostBytes": 1187061266547,
"hostPackets": 0,
"peerBytes": 1187061266547,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 37.95871271848625,
"serverBytesRatio": 62.04128728151375,
"hostRole": "CLIENT_AND_SERVER"
},
"others": {
  "deviceId": 622,
  "percent": 8.952497330035014,
  "records": 3426395,
  "count": 237676,
  "bytes": 212543256387,
  "packets": 185240239,
  "flows": 955221,
  "connections": 3018752,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 489612,
  "hostFlows": 612394,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 117898594467,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 5469862151,
  "hostServerPackets": 0,
  "peers": 237676,
  "peerFlows": 342872,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 54666982807,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 34507816962,
  "peerServerPackets": 0,
  "hostBytes": 123368456618,
  "hostPackets": 0,
  "peerBytes": 89174799769,
  "peerPackets": 0,
  "peerRole": "CLIENT_AND_SERVER",
  "hostBytesRatio": 58.043928899522456,
  "peerBytesRatio": 41.956071100477544,
  "clientBytesRatio": 81.1908033251319,
  "serverBytesRatio": 18.809196674868108,
  "hostRole": "CLIENT_AND_SERVER"
},
"results": [{
  "deviceId": 622,
  "rank": 1,
  "percent": 19.52122756322157,
  "records": 2432,
  "bytes": 463457862315,
  "packets": 403024824,
  "flows": 54,
  "connections": 0,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 8,
  "hostFlows": 13,
  "hostConnections": 0
```

```
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 769621,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 460539990536,
    "hostServerPackets": 0,
    "peers": 1,
    "peer": {
      "ipAddress": "10.203.7.10",
      "country": "XR",
      "hostGroupIds": [65534],
      "name": "Catch All"
    },
    "peerFlows": 41,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 2915260277,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 1841881,
    "peerServerPackets": 0,
    "hostBytes": 460540760157,
    "hostPackets": 0,
    "peerBytes": 2917102158,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 99.37057877421068,
    "peerBytesRatio": 0.6294212257893088,
    "clientBytesRatio": 0.6291898649500204,
    "serverBytesRatio": 99.37081013504998,
    "hostRole": "CLIENT_AND_SERVER"
  }, {
    "deviceId": 622,
    "rank": 2,
    "percent": 19.52054924642134,
    "records": 2141,
    "bytes": 463441758243,
    "packets": 403003678,
    "flows": 24,
    "connections": 402,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 5,
    "hostFlows": 14,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 2898322624,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 3656300,
    "hostServerPackets": 0,
    "peers": 1,
    "peer": {
      "ipAddress": "10.203.7.10",
      "country": "XR",
      "hostGroupIds": [65534],
      "name": "Catch All"
    },
    "peerFlows": 10,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 1760435,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 460538018884,
    "peerServerPackets": 0,
    "hostBytes": 2901978924,
    "hostPackets": 0,
    "peerBytes": 460539779319,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 0.6261798537537877,
    "peerBytesRatio": 99.37382014624622,
    "clientBytesRatio": 0.6257707699873211,
    "serverBytesRatio": 99.37422923001267,
```

```

    "hostRole": "CLIENT_AND_SERVER"
  }, {
    "deviceId": 622,
    "rank": 3,
    "percent": 11.850052326210786,
    "records": 1159,
    "bytes": 281334762460,
    "packets": 89765875,
    "flows": 134,
    "connections": 328,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 5,
    "hostFlows": 30,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 1465248250,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 104717561615,
    "hostServerPackets": 0,
    "peers": 1,
    "peer": {
      "ipAddress": "10.203.7.10",
      "country": "XR",
      "hostGroupIds": [65534],
      "name": "Catch All"
    },
    "peerFlows": 104,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 73483754254,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 101668198341,
    "peerServerPackets": 0,
    "hostBytes": 106182809865,
    "hostPackets": 0,
    "peerBytes": 175151952595,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 37.74251320261107,
    "peerBytesRatio": 62.25748679738893,
    "clientBytesRatio": 26.64050537112569,
    "serverBytesRatio": 73.35949462887432,
    "hostRole": "CLIENT_AND_SERVER"
  }]
}
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

POST

/tenants/{tenantId}/flow-reports/top-conversations/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included
      - **excludes**: List of predefined application ids to be excluded
    - **direction**: Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols**: Port Protocols used or unused in the flow.
      - **includes**: List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes**: List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp

The following protocols are allowed to include or exclude

      - tcp: Should have a valid port number
      - udp: Should have a valid port number
      - icmp: Should not have a port number
  - **subject**:
    - **tags**: Specify the tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: Specify the IP addresses or range of IP addresses involved or not involved in the flow.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **peer**:
    - **tags**: Specify the peer tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: List of IP addresses to be included or excluded.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **orientation**: Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
    - EITHER (Default)
    - CLIENT
    - SERVER
  - **maxRows**: The maximum no. of records to be returned. The maximum value for this field is 5000.
    - Default: 50
  - **flowCollectors**: List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.
  - **orderBy**: The order based on which the records will be retrieved and sorted by (i.e. Bytes or Packets or Flows or TCP Connection). The value must be one of the following:
    - TOTAL\_BYTES (Default)
    - TOTAL\_PACKETS
    - TOTAL\_FLOWS
    - TOTAL\_CONNECTIONS
  - **standardOptions**: Flag to return the default columns.
    - Value should be either true or false
    - In case if the value is set as "true" or set to default, then implicitly fields part of AdvancedOptions will be set to its default value.
    - Default value: true
  - **advanceOptions**
    - **excludeBpsPps**: Flag to excludes bps/pps values.
      - Value should be either true or false





```
    },
    "required": [
      "includes",
      "excludes"
    ]
  }
},
"required": [
  "direction"
]
},
"subject": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    }
  },
  "required": []
},
"peer": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      }
    }
  }
}
```

```

    },
    "required": [
      "includes",
      "excludes"
    ]
  },
  "required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 400000
},
"flowCollectors": {
  "type": "array",
  "items": {}
},
"orderBy": {
  "type": "string",
  "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
  "default": "TOTAL_BYTES"
},
"excludeBpsPps": {
  "type": "boolean",
  "default": true
},
"excludeOthers": {
  "type": "boolean",
  "default": true
},
"excludeCounts": {
  "type": "boolean",
  "default": false
},
"defaultColumns": {
  "type": "boolean",
  "default": true
}
},
"required": [
  "startTime",
  "endTime"
]
}

```

**Example:**

```
{
  "searchName":"Top Reports on 3/1/2017 at 12:36 PM",
  "startTime":"2017-03-10T00:00:00.000",
  "endTime":"2017-03-10T00:05:00.000",
  "connection":{
    "applications":{
      "includes":[103119],
      "excludes":[2000]
    },
    "direction":"INBOUND_PLUS_OUTBOUND",
    "portProtocols":{
      "includes":["70/tcp"],
      "excludes":["80/tcp"]
    }
  },
  "subject":{
    "tags":{
      "includes":[63000],
      "excludes":[63006]
    },
    "ipAddresses":{
      "includes":["10.20.11.11"],
      "excludes":[]
    }
  },
  "peer":{
    "tags":{
      "includes":[47],
      "excludes":[60000]
    },
    "ipAddresses":{
      "includes":[],
      "excludes":["10.20.11.12"]
    }
  },
  "orientation":"either",
  "maxRows":50,
  "flowCollectors":["162"],
  "orderBy":"TOTAL_BYTES",
  "excludeBpsPps":true,
  "excludeOthers":true,
  "excludeCounts":false,
  "defaultColumns": true
}
```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

Type: application/json

Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      }
    },
    "required": [
      "queryId",
      "status"
    ]
  }
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/flow-reports/top-conversations/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-conversations/queries/{queryId}

Get the status of the search

**Request**

**URI Parameters**

- **tenantId:** *required (string)*
- **queryId:** *required (string)*

**Headers**

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

**Example:**





- **peerFlows**: The number of flows used by "Other Hosts"(peers)
- **peerConnections**: Number of TCP connections used by the "Other Hosts"(peers)
- **peerClients**: The number of client hosts among the "Other Hosts" in the corresponding conversation
- **peerClientBytes**:The number of bytes used by client hosts among the "Other Hosts" in the corresponding conversation
- **peerClientPackets**: The number of packets used by client hosts among the "Other Hosts" in the corresponding conversation
- **peerServers**: The number of "Other Hosts"acting as servers, in the corresponding conversation
- **peerServerBytes**: The number of bytes used by server hosts among the "Other Hosts" in the corresponding conversation
- **peerServerPackets**: The number of packets used by server hosts among the "Other Hosts" in the corresponding conversation
- **protocolNumber**: Provides number of the protocol
- **port**: The port that the corresponding conversation was using
- **protocol**: The protocol that the corresponding conversation was using
- **hostRole**: The role of the hosts selected in the Filter used in the corresponding conversation
- **hostBytes**: The number of bytes for the hosts selected in the Filter used in the corresponding conversation
- **hostPackets**: The number of packets for the hosts selected in the Filter used in the corresponding conversation
- **peerRole**: The role of the "Other Hosts"(peers)
- **peerBytes**: The number of bytes used by "Other Hosts" in the corresponding conversation
- **peerPackets**: The number of packets used by "Other Hosts" in the corresponding conversation
- **hostBytesRatio**: The ratio, in percent, of host bytes to the bytes used in the corresponding conversation
- **peerBytesRatio**: The ratio, in percent, of peer bytes to the bytes used in the corresponding conversation
- **clientBytesRatio**: The ratio, in percent, of client bytes to the bytes used in the corresponding conversation
- **serverBytesRatio**: The ratio, in percent, of server bytes to the bytes used in the corresponding conversation

### Body

**Type:** application/json

### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "others": {
          "type": "object",
          "properties": {
            "bytes": {
              "type": "integer"
            },
            "clientBytesRatio": {
              "type": "number"
            },
            "connections": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "deviceId": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "hostBytes": {
              "type": "integer"
            },
            "hostBytesRatio": {
              "type": "number"
            },
            "hostClientBytes": {
              "type": "integer"
            },
            "hostClientPackets": {
              "type": "integer"
            },
            "hostClients": {
              "type": "integer"
            },
            "hostConnections": {
              "type": "integer"
            },
            "hostFlows": {
              "type": "integer"
            },
            "hostPackets": {
```

```
    "type": "integer"
  },
  "hostRole": {
    "type": "string"
  },
  "hostServerBytes": {
    "type": "integer"
  },
  "hostServerPackets": {
    "type": "integer"
  },
  "hostServers": {
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "number"
  },
  "packetRateAvg": {
    "type": "number"
  },
  "packetRateMax": {
    "type": "number"
  },
  "packetRateMin": {
    "type": "number"
  },
  "packets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerBytesRatio": {
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerPackets": {
    "type": "integer"
  },
  "peerRole": {
    "type": "string"
  },
  "peerServerBytes": {
    "type": "integer"
  },
  "peerServerPackets": {
    "type": "integer"
  },
  "peerServers": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "percent": {
    "type": "number"
  },
  "records": {
    "type": "integer"
  },
  "serverBytesRatio": {
    "type": "number"
  },
  "trafficRate95th": {
    "type": "number"
  }
```



```
    },
    "trafficRateAvg": {
      "type": "number"
    },
    "trafficRateMax": {
      "type": "number"
    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "count",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
  ]
},
"results": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "bytes": {
        "type": "integer"
      },
      "clientBytesRatio": {
        "type": "number"
      },
      "connections": {
        "type": "integer"
      },
      "deviceId": {
        "type": "integer"
      },
      "flows": {
        "type": "integer"
      },
      "host": {
        "type": "object",
        "properties": {
          "country": {
```

```
    "type": "string"
  },
  "hostGroupIds": {
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "ipAddress": {
    "type": "string"
  },
  "name": {
    "type": "string"
  }
},
"required": [
  "country",
  "ipAddress",
  "name",
  "hostGroupIds"
]
},
"hostBytes": {
  "type": "integer"
},
"hostBytesRatio": {
  "type": "number"
},
"hostClientBytes": {
  "type": "integer"
},
"hostClientPackets": {
  "type": "integer"
},
"hostClients": {
  "type": "integer"
},
"hostConnections": {
  "type": "integer"
},
"hostFlows": {
  "type": "integer"
},
"hostPackets": {
  "type": "integer"
},
"hostRole": {
  "type": "string"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"packetRate95th": {
  "type": "number"
},
"packetRateAvg": {
  "type": "number"
},
"packetRateMax": {
  "type": "number"
},
"packetRateMin": {
  "type": "number"
},
"packets": {
  "type": "integer"
},
"peer": {
  "type": "object",
  "properties": {
    "country": {
      "type": "string"
    }
  }
}
}
```

```
,
  "hostGroupIds": {
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "ipAddress": {
    "type": "string"
  },
  "name": {
    "type": "string"
  }
},
"required": [
  "country",
  "ipAddress",
  "name",
  "hostGroupIds"
]
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
"peerFlows": {
  "type": "integer"
},
"peerPackets": {
  "type": "integer"
},
"peerRole": {
  "type": "string"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"percent": {
  "type": "number"
},
"port": {
  "type": "integer"
},
"portProtocol": {
  "type": "object",
  "properties": {
    "port": {
      "type": "integer"
    },
    "protocol": {
      "type": "string"
    }
  }
},
"required": [
  "protocol",
  "port"
]
},
"protocol": {
```

```

        "type": "string"
    },
    "protocolNumber": {
        "type": "integer"
    },
    "rank": {
        "type": "integer"
    },
    "records": {
        "type": "integer"
    },
    "serverBytesRatio": {
        "type": "number"
    },
    "trafficRate95th": {
        "type": "number"
    },
    "trafficRateAvg": {
        "type": "number"
    },
    "trafficRateMax": {
        "type": "number"
    },
    "trafficRateMin": {
        "type": "number"
    }
},
"required": [
    "hostPackets",
    "hostServerBytes",
    "protocol",
    "peerFlows",
    "packetRateMin",
    "packets",
    "rank",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "port",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "host",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "portProtocol",
    "peer",
    "protocolNumber",
    "hostServers",
    "peerServers",
    "peers",
    "packetRate95th",
    "trafficRateAvg",
    "peerBytesRatio",
    "peerServerBytes",
    "hostConnections",
    "peerClients",
    "bytes",
    "peerClientBytes",
    "peerServerPackets",
    "hosts",
    "hostClients",
    "peerRole",
    "hostRole",
    "packetRateAvg"
]
}
},
"summary": {

```

```
"type": "object",
"properties": {
  "bytes": {
    "type": "integer"
  },
  "clientBytesRatio": {
    "type": "number"
  },
  "connections": {
    "type": "integer"
  },
  "count": {
    "type": "integer"
  },
  "deviceId": {
    "type": "integer"
  },
  "flows": {
    "type": "integer"
  },
  "hostBytes": {
    "type": "integer"
  },
  "hostBytesRatio": {
    "type": "number"
  },
  "hostClientBytes": {
    "type": "integer"
  },
  "hostClientPackets": {
    "type": "integer"
  },
  "hostClients": {
    "type": "integer"
  },
  "hostConnections": {
    "type": "integer"
  },
  "hostFlows": {
    "type": "integer"
  },
  "hostPackets": {
    "type": "integer"
  },
  "hostRole": {
    "type": "string"
  },
  "hostServerBytes": {
    "type": "integer"
  },
  "hostServerPackets": {
    "type": "integer"
  },
  "hostServers": {
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "number"
  },
  "packetRateAvg": {
    "type": "number"
  },
  "packetRateMax": {
    "type": "number"
  },
  "packetRateMin": {
    "type": "number"
  },
  "packets": {
    "type": "integer"
  },
  "peerBytes": {
    "type": "integer"
  },
  "peerBytesRatio": {
    "type": "number"
  },
  "peerClientBytes": {
    "type": "integer"
  }
}
```

```
    },
    "peerClientPackets": {
      "type": "integer"
    },
    "peerClients": {
      "type": "integer"
    },
    "peerConnections": {
      "type": "integer"
    },
    "peerFlows": {
      "type": "integer"
    },
    "peerPackets": {
      "type": "integer"
    },
    "peerRole": {
      "type": "string"
    },
    "peerServerBytes": {
      "type": "integer"
    },
    "peerServerPackets": {
      "type": "integer"
    },
    "peerServers": {
      "type": "integer"
    },
    "peers": {
      "type": "integer"
    },
    "percent": {
      "type": "number"
    },
    "records": {
      "type": "integer"
    },
    "serverBytesRatio": {
      "type": "number"
    },
    "trafficRate95th": {
      "type": "number"
    },
    "trafficRateAvg": {
      "type": "number"
    },
    "trafficRateMax": {
      "type": "number"
    },
    "trafficRateMin": {
      "type": "number"
    }
  },
  "required": [
    "hostPackets",
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "connections",
    "trafficRateMin",
    "hostBytes",
    "hostClientPackets",
    "hostBytesRatio",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "clientBytesRatio",
    "hostFlows",
    "packetRateMax",
    "peerBytes",
    "serverBytesRatio",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "peerPackets",
    "deviceId",
    "hostServers",
    "peerServers"
```

```

        peerServers ,
        "count",
        "peers",
        "packetRate95th",
        "trafficRateAvg",
        "peerBytesRatio",
        "peerServerBytes",
        "hostConnections",
        "peerClients",
        "bytes",
        "peerClientBytes",
        "peerServerPackets",
        "hosts",
        "hostClients",
        "peerRole",
        "hostRole",
        "packetRateAvg"
    ]
}
},
"required": [
    "others",
    "results",
    "summary"
]
}
},
"required": [
    "data"
]
}
}

```

**Example:**

```

{
  "data" : {
    "summary": {
      "deviceId": 362,
      "percent": 100.0,
      "records": 40094,
      "count": 19859,
      "bytes": 29854345956,
      "packets": 24072222,
      "flows": 20035,
      "connections": 12274,
      "trafficRateMin": 0.0,
      "trafficRateMax": 0.0,
      "trafficRateAvg": 0.0,
      "trafficRate95th": 0.0,
      "packetRateMin": 0.0,
      "packetRateMax": 0.0,
      "packetRateAvg": 0.0,
      "packetRate95th": 0.0,
      "hosts": 19859,
      "hostFlows": 10018,
      "hostConnections": 0,
      "hostClients": 0,
      "hostClientBytes": 9162438024,
      "hostClientPackets": 0,
      "hostServers": 0,
      "hostServerBytes": 5764734954,
      "hostServerPackets": 0,
      "peers": 19859,
      "peerFlows": 10018,
      "peerConnections": 0,
      "peerClients": 0,
      "peerClientBytes": 9162438024,
      "peerClientPackets": 0,
      "peerServers": 0,
      "peerServerBytes": 5764734954,
      "peerServerPackets": 0,
      "hostRole": "CLIENT_AND_SERVER",
      "hostBytes": 14927172978,
      "hostPackets": 0,
      "peerBytes": 14927172978,
      "peerPackets": 0,
      "peerRole": "CLIENT_AND_SERVER",
      "hostBytesRatio": 50.0,
      "peerBytesRatio": 50.0,
      "clientBytesRatio": 61.38093286320059,
      "serverBytesRatio": 38.61906713679941
    },
  },
}

```

```
"others": {
  "deviceId": 362,
  "percent": 35.016615763103005,
  "records": 39996,
  "count": 19833,
  "bytes": 10453981612,
  "packets": 9742228,
  "flows": 20009,
  "connections": 12274,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 19833,
  "hostFlows": 10005,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 4046007836,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 1180982970,
  "hostServerPackets": 0,
  "peers": 19833,
  "peerFlows": 10005,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 4046007836,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 1180982970,
  "peerServerPackets": 0,
  "hostRole": "CLIENT_AND_SERVER",
  "hostBytes": 5226990806,
  "hostPackets": 0,
  "peerBytes": 5226990806,
  "peerPackets": 0,
  "peerRole": "CLIENT_AND_SERVER",
  "hostBytesRatio": 50.0,
  "peerBytesRatio": 50.0,
  "clientBytesRatio": 77.40606375958488,
  "serverBytesRatio": 22.59393624041511
},
"results": [{
  "deviceId": 362,
  "rank": 1,
  "percent": 13.782706055809733,
  "records": 4,
  "bytes": 4114736748,
  "packets": 3622849,
  "flows": 1,
  "connections": 0,
  "trafficRateMin": 0.0,
  "trafficRateMax": 0.0,
  "trafficRateAvg": 0.0,
  "trafficRate95th": 0.0,
  "packetRateMin": 0.0,
  "packetRateMax": 0.0,
  "packetRateAvg": 0.0,
  "packetRate95th": 0.0,
  "hosts": 1,
  "hostFlows": 0,
  "hostConnections": 0,
  "hostClients": 0,
  "hostClientBytes": 0,
  "hostClientPackets": 0,
  "hostServers": 0,
  "hostServerBytes": 4085092052,
  "hostServerPackets": 0,
  "peers": 1,
  "peerFlows": 1,
  "peerConnections": 0,
  "peerClients": 0,
  "peerClientBytes": 29644696,
  "peerClientPackets": 0,
  "peerServers": 0,
  "peerServerBytes": 0,
  "peerServerPackets": 0,
  "host": {
```



```
"ipAddress": "10.203.0.243",
"country": "XR",
"hostGroupIds": [65534],
"name": "Catch All"
},
"peer": {
"ipAddress": "10.203.7.10",
"country": "XR",
"hostGroupIds": [65534],
"name": "Catch All"
},
"protocolNumber": 6,
"port": 443,
"portProtocol": {
"protocol": "TCP",
"port": 443
},
},
"protocol": "TCP",
"hostRole": "SERVER",
"hostBytes": 4085092052,
"hostPackets": 0,
"peerBytes": 29644696,
"peerPackets": 0,
"peerRole": "CLIENT",
"hostBytesRatio": 99.27954817487634,
"peerBytesRatio": 0.7204518251236616,
"clientBytesRatio": 0.7204518251236616,
"serverBytesRatio": 99.27954817487634
}, {
"deviceId": 362,
"rank": 2,
"percent": 13.782706055809733,
"records": 4,
"bytes": 4114736748,
"packets": 3622849,
"flows": 1,
"connections": 0,
"trafficRateMin": 0.0,
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 1,
"hostFlows": 1,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 29644696,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 0,
"hostServerPackets": 0,
"peers": 1,
"peerFlows": 0,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 0,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 4085092052,
"peerServerPackets": 0,
"host": {
"ipAddress": "10.203.7.10",
"country": "XR",
"hostGroupIds": [65534],
"name": "Catch All"
},
"peer": {
"ipAddress": "10.203.0.243",
"country": "XR",
"hostGroupIds": [65534],
"name": "Catch All"
},
"protocolNumber": 6,
"port": 443,
"portProtocol": {
"protocol": "TCP",
"port": 443
},
},
"protocol": "TCP"
```

```

    "protocol": "TCP",
    "hostRole": "CLIENT",
    "hostBytes": 29644696,
    "hostPackets": 0,
    "peerBytes": 4085092052,
    "peerPackets": 0,
    "peerRole": "SERVER",
    "hostBytesRatio": 0.7204518251236616,
    "peerBytesRatio": 99.27954817487634,
    "clientBytesRatio": 0.7204518251236616,
    "serverBytesRatio": 99.27954817487634
  }, {
    "deviceId": 362,
    "rank": 3,
    "percent": 3.042928859801145,
    "records": 4,
    "bytes": 908446509,
    "packets": 646509,
    "flows": 1,
    "connections": 0,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 1,
    "hostFlows": 1,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 908446509,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 0,
    "hostServerPackets": 0,
    "peers": 1,
    "peerFlows": 0,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 0,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 0,
    "peerServerPackets": 0,
    "host": {
      "ipAddress": "10.203.0.227",
      "country": "XR",
      "hostGroupIds": [65534],
      "name": "Catch All"
    },
    "peer": {
      "ipAddress": "10.203.1.114",
      "country": "XR",
      "hostGroupIds": [65534],
      "name": "Catch All"
    },
    "protocolNumber": 17,
    "port": 2055,
    "portProtocol": {
      "protocol": "UDP",
      "port": 2055
    },
    "protocol": "UDP",
    "hostRole": "CLIENT",
    "hostBytes": 908446509,
    "hostPackets": 0,
    "peerBytes": 0,
    "peerPackets": 0,
    "peerRole": "SERVER",
    "hostBytesRatio": 100.0,
    "peerBytesRatio": 0.0,
    "clientBytesRatio": 100.0,
    "serverBytesRatio": 0.0
  }
}
}
}

```

#### HTTP status code 400

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/flow-reports/top-services/queries

POST

Initiate Search

POST /tenants/{tenantId}/flow-reports/top-services/queries

- Initiate a search
- All the properties in the request body are optional except **startTime**, **endTime**.
  - **searchName**: Name for the search.
  - **startTime**: Start Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **endTime**: End Time in UTC for flow search
    - Format: yyyy-MM-dd'T'HH:mm:ss.SSS
    - Required field
  - **connection**:
    - **applications**: Specify the applications involved or not involved in the flow.
      - **includes**: List of predefined application ids to be included
      - **excludes**: List of predefined application ids to be excluded
    - **direction**: Specify the direction of the flow. The value must be one of the following:
      - INBOUND\_PLUS\_OUTBOUND (Default)
      - INBOUND
      - OUTBOUND
      - WITHIN
    - **portProtocols**: Port Protocols used or unused in the flow.
      - **includes**: List of Port Protocols to be included
        - Ex: 80/tcp or 75/udp or icmp
      - **excludes**: List of Port Protocols to be excluded
        - Ex: 80/tcp or 75/udp or icmp

The following protocols are allowed to include or exclude

      - tcp: Should have a valid port number
      - udp: Should have a valid port number
      - icmp: Should not have a port number
  - **subject**:
    - **tags**: Specify the tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: Specify the IP addresses or range of IP addresses involved or not involved in the flow.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **peer**:
    - **tags**: Specify the peer tags involved or not involved in the flow
      - **includes**: List of tag ids to be included.
      - **excludes**: List of tag ids to be excluded.
    - **ipAddresses**: List of IP addresses to be included or excluded.
      - **includes**: List of IP addresses to be included.
        - Ex: 192.168.10.10 or 192.168.10.10-15
      - **excludes**: List of IP addresses to be excluded.
        - Ex: 192.168.10.10 or 192.168.10.10-15
  - **orientation**: Whether the subject information is considered to be part of client or server or either. The value must be one of the following:
    - EITHER (Default)
    - CLIENT

- SERVER
- **maxRows**: The maximum no. of records to be returned. The maximum value for this field is 5000.
  - Default: 50
- **flowCollectors**: List of Flow Collectors that the system will search. If no Flow Collector ID is specified, the system will search all Flow Collectors.
- **orderBy**: The order based on which the records will be retrieved and sorted by (i.e. Bytes or Packets or Flows or TCP Connection). The value must be one of the following:
  - TOTAL\_BYTES (Default)
  - TOTAL\_PACKETS
  - TOTAL\_FLOWS
  - TOTAL\_CONNECTIONS
- **standardOptions**: Flag to return the default columns.
  - Value should be either true or false
  - Incase If the value is set as "true" or set to default, then implicitly fields part of AdvancedOptions will be set to its default value.
  - Default value: true
- **advanceOptions**
  - **excludeBpsPps**: Flag to excludes bps/pps values.
    - Value should be either true or false
    - Default value: true
  - **excludeOthers**: Flag to excludes Other Records
    - Value should be either true or false
    - Default value: true
  - **excludeCounts**: Flag to excludes Counts
    - Value should be either true or false
    - Default value: false

## Request

### URI Parameters

- **tenantId**: *required (string)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIiXm0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsImV4cCI6MTYwMjUwMDAwfQ.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIiXm0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhbiIsImV4cCI6MTYwMjUwMDAwfQ.
```

### Body

**Type:** application/json

### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "searchName": {
      "type": "string"
    },
    "startTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "endTime": {
      "type": "date",
      "format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
      "required": true
    },
    "connection": {
      "type": "object",
      "properties": {
        "applications": {
          "type": "object",
          "properties": {
            "includes": {
              "type": "array",
              "items": {}
            },
            "excludes": {
              "type": "array",
              "items": {}
            }
          }
        }
      }
    }
  }
}
```

```

    "required": [
      "includes",
      "excludes"
    ]
  },
  "direction": {
    "type": "string",
    "enum": ["INBOUND_PLUS_OUTBOUND", "INBOUND", "OUTBOUND", "WITHIN"],
    "default": "INBOUND_PLUS_OUTBOUND"
  },
  "portProtocols": {
    "type": "object",
    "properties": {
      "includes": {
        "type": "array",
        "items": {}
      },
      "excludes": {
        "type": "array",
        "items": {}
      }
    },
    "required": [
      "includes",
      "excludes"
    ]
  },
  "required": [
    "direction"
  ]
},
"subject": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    },
    "ipAddresses": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        },
        "excludes": {
          "type": "array",
          "items": {}
        }
      },
      "required": [
        "includes",
        "excludes"
      ]
    }
  },
  "required": []
},
"peer": {
  "type": "object",
  "properties": {
    "tags": {
      "type": "object",
      "properties": {
        "includes": {
          "type": "array",
          "items": {}
        }
      }
    }
  }
}

```

```

    },
    "excludes": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "includes",
    "excludes"
  ]
},
"ipAddresses": {
  "type": "object",
  "properties": {
    "includes": {
      "type": "array",
      "items": {}
    },
    "excludes": {
      "type": "array",
      "items": {}
    }
  },
  "required": [
    "includes",
    "excludes"
  ]
}
},
"required": []
},
"orientation": {
  "type": "string",
  "enum": ["EITHER", "CLIENT", "SERVER"],
  "default": "EITHER"
},
"maxRows": {
  "type": "integer",
  "default": 50,
  "max": 400000
},
"flowCollectors": {
  "type": "array",
  "items": {}
},
"orderBy": {
  "type": "string",
  "enum": ["TOTAL_BYTES", "TOTAL_PACKETS", "TOTAL_FLOWS", "TOTAL_CONNECTIONS"],
  "default": "TOTAL_BYTES"
},
"excludeBpsPps": {
  "type": "boolean",
  "default": true
},
"excludeOthers": {
  "type": "boolean",
  "default": true
},
"excludeCounts": {
  "type": "boolean",
  "default": false
},
"defaultColumns": {
  "type": "boolean",
  "default": true
}
},
"required": [
  "startTime",
  "endTime"
]
}

```

**Example:**

```
{
  "searchName":"Top Reports on 3/1/2017 at 12:36 PM",
  "startTime":"2017-03-10T00:00:00.000",
  "endTime":"2017-03-10T00:05:00.000",
  "connection":{
    "applications":{
      "includes":[103119],
      "excludes":[2000]
    },
    "direction":"INBOUND_PLUS_OUTBOUND",
    "portProtocols":{
      "includes":["70/tcp"],
      "excludes":["80/tcp"]
    }
  },
  "subject":{
    "tags":{
      "includes":[63000],
      "excludes":[63006]
    },
    "ipAddresses":{
      "includes":["10.20.11.11"],
      "excludes":[]
    }
  },
  "peer":{
    "tags":{
      "includes":[47],
      "excludes":[60000]
    },
    "ipAddresses":{
      "includes":[],
      "excludes":["10.20.11.12"]
    }
  },
  "orientation":"either",
  "maxRows":50,
  "flowCollectors": ["162"],
  "orderBy":"TOTAL_BYTES",
  "excludeBpsPps":true,
  "excludeOthers":true,
  "excludeCounts":false,
  "defaultColumns": true
}
```

## Response

### HTTP status code 200

It has a **data** element which has following fields.

- **queryId**: Query id.
- **status**: The current status of the search. It will be one of the follows.
  - **PENDING**: The search job has been created, but may not have been started yet.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It's been started, but is not yet complete.
  - **WAITING**: The search job is WAITING. It will run at some point, but hasn't started.
  - **COMPLETED**: The search job has been COMPLETED. Results should be available.
  - **FAILED**: The search job has FAILED. Some part of the job may have finished, but not all of it. It cannot be restarted.

### Body

**Type:** application/json

**Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "queryId": {
          "type": "string"
        },
        "status": {
          "type": "string"
        }
      }
    },
    "required": [
      "queryId",
      "status"
    ]
  }
}
```

**Example:**

```
{
  "data": {
    "queryId": "58b9394b60b2998c00cd63f7",
    "status": "IN_PROGRESS"
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/flow-reports/top-services/queries/{queryId}

GET

Get Search Status.

GET

/tenants/{tenantId}/flow-reports/top-services/queries/{queryId}

Get the status of the search

**Request**

**URI Parameters**

- **tenantId:** *required (string)*
- **queryId:** *required (string)*

**Headers**

- **Cookie:** *required (string)*  
JSON Web Token for the authenticated user.

**Example:**







- **hostRole**: The role of the hosts selected in the Filter that are used in the corresponding service.
- **hostBytes**: The number of bytes for the hosts selected in the Filter that are used in the corresponding service.
- **hostPackets**: The number of packets for the hosts selected in the Filter that are used in the corresponding service.
- **peerRole**: The role of the "Other Hosts"(peers).
- **peerBytes**: The number of bytes used by "Other Hosts" in the corresponding service.
- **peerPackets**: The number of packets used by "Other Hosts" in the corresponding service.
- **hostBytesRatio**: The ratio (represented by a percentage) of host bytes to the bytes used in the corresponding service.
- **peerBytesRatio**: The ratio (represented by a percentage) of peer bytes to the bytes used in the corresponding service.
- **clientBytesRatio**: The ratio (represented by a percentage) of client bytes to the bytes used in the corresponding service.
- **serverBytesRatio**: The ratio (represented by a percentage) of server bytes to the bytes used in the corresponding service.

## Body

**Type:** application/json

## Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "summary": {
          "type": "object",
          "properties": {
            "deviceId": {
              "type": "integer"
            },
            "percent": {
              "type": "integer"
            },
            "records": {
              "type": "integer"
            },
            "count": {
              "type": "integer"
            },
            "bytes": {
              "type": "integer"
            },
            "packets": {
              "type": "integer"
            },
            "flows": {
              "type": "integer"
            },
            "connections": {
              "type": "integer"
            },
            "trafficRateMin": {
              "type": "integer"
            },
            "trafficRateMax": {
              "type": "integer"
            },
            "trafficRateAvg": {
              "type": "integer"
            },
            "trafficRate95th": {
              "type": "integer"
            },
            "packetRateMin": {
              "type": "integer"
            },
            "packetRateMax": {
              "type": "integer"
            },
            "packetRateAvg": {
              "type": "integer"
            },
            "packetRate95th": {
              "type": "integer"
            },
            "hosts": {
              "type": "integer"
            },
            "hostFlows": {
              "type": "integer"
            }
          }
        }
      }
    }
  }
}
```

```
,
"hostConnections": {
  "type": "integer"
},
"hostClients": {
  "type": "integer"
},
"hostClientBytes": {
  "type": "integer"
},
"hostClientPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"peerFlows": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"hostBytes": {
  "type": "integer"
},
"hostBytesRatio": {
  "type": "number"
},
"hostPackets": {
  "type": "integer"
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
  "type": "number"
},
"peerRole": {
  "type": "string"
},
"peerPackets": {
  "type": "integer"
},
"clientBytesRatio": {
  "type": "number"
},
"serverBytesRatio": {
  "type": "number"
},
"hostRole": {
  "type": "string"
}
},
"required": [
  "deviceId",
  ..
  ..
```

```
    "percent",
    "records",
    "count",
    "bytes",
    "packets",
    "flows",
    "connections",
    "trafficRateMin",
    "trafficRateMax",
    "trafficRate95th",
    "trafficRateAvg",
    "packetRateMin",
    "packetRateMax",
    "packetRateAvg",
    "packetRate95th",
    "hostServerBytes",
    "hosts",
    "hostFlows",
    "hostConnections",
    "hostClients",
    "hostClientBytes",
    "hostClientPackets",
    "hostServers",
    "hostServerPackets",
    "peers",
    "peerFlows",
    "peerConnections",
    "peerClients",
    "peerClientBytes",
    "peerClientPackets",
    "peerServers",
    "peerServerBytes",
    "peerServerPackets",
    "hostBytes",
    "hostBytesRatio",
    "hostPackets",
    "peerBytes",
    "peerBytesRatio",
    "peerRole",
    "peerPackets",
    "clientBytesRatio",
    "serverBytesRatio",
    "hostRole"
  ],
  "type": "object"
},
"others": {
  "type": "object",
  "properties": {
    "deviceId": {
      "type": "integer"
    },
    "percent": {
      "type": "number"
    },
    "records": {
      "type": "integer"
    },
    "count": {
      "type": "integer"
    },
    "bytes": {
      "type": "integer"
    },
    "packets": {
      "type": "integer"
    },
    "flows": {
      "type": "integer"
    },
    "connections": {
      "type": "integer"
    },
    "trafficRateMin": {
      "type": "integer"
    },
    "trafficRateMax": {
      "type": "integer"
    },
    "trafficRateAvg": {
      "type": "integer"
    }
  },
}
```

```
"trafficRate95th": {
  "type": "integer"
},
"packetRateMin": {
  "type": "integer"
},
"packetRateMax": {
  "type": "integer"
},
"packetRateAvg": {
  "type": "integer"
},
"packetRate95th": {
  "type": "integer"
},
"hosts": {
  "type": "integer"
},
"hostFlows": {
  "type": "integer"
},
"hostConnections": {
  "type": "integer"
},
"hostClients": {
  "type": "integer"
},
"hostClientBytes": {
  "type": "integer"
},
"hostClientPackets": {
  "type": "integer"
},
"hostServers": {
  "type": "integer"
},
"hostServerBytes": {
  "type": "integer"
},
"hostServerPackets": {
  "type": "integer"
},
"peers": {
  "type": "integer"
},
"peerFlows": {
  "type": "integer"
},
"peerConnections": {
  "type": "integer"
},
"peerClients": {
  "type": "integer"
},
"peerClientBytes": {
  "type": "integer"
},
"peerClientPackets": {
  "type": "integer"
},
"peerServers": {
  "type": "integer"
},
"peerServerBytes": {
  "type": "integer"
},
"peerServerPackets": {
  "type": "integer"
},
"hostBytes": {
  "type": "integer"
},
"hostBytesRatio": {
  "type": "number"
},
"hostPackets": {
  "type": "integer"
},
"peerBytes": {
  "type": "integer"
},
"peerBytesRatio": {
```

```
        "type": "number"
      },
      "peerRole": {
        "type": "string"
      },
      "peerPackets": {
        "type": "integer"
      },
      "clientBytesRatio": {
        "type": "number"
      },
      "serverBytesRatio": {
        "type": "number"
      },
      "hostRole": {
        "type": "string"
      }
    },
    "required": [
      "hostServerBytes",
      "peerFlows",
      "packetRateMin",
      "packets",
      "connections",
      "trafficRateMin",
      "hostClientPackets",
      "hostClientBytes",
      "percent",
      "peerClientPackets",
      "trafficRateMax",
      "trafficRate95th",
      "hostFlows",
      "packetRateMax",
      "peerConnections",
      "flows",
      "records",
      "hostServerPackets",
      "deviceId",
      "hostServers",
      "peerServers",
      "count",
      "peers",
      "packetRate95th",
      "trafficRateAvg",
      "peerServerBytes",
      "hostConnections",
      "peerClients",
      "bytes",
      "peerClientBytes",
      "peerServerPackets",
      "hosts",
      "hostClients",
      "packetRateAvg",
      "hostBytes",
      "hostBytesRatio",
      "hostPackets",
      "peerBytes",
      "peerBytesRatio",
      "peerRole",
      "peerPackets",
      "clientBytesRatio",
      "serverBytesRatio",
      "hostRole"
    ],
    "type": "object"
  },
  "results": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "deviceId": {
          "type": "integer"
        },
        "percent": {
          "type": "number"
        },
        "rank": {
          "type": "integer"
        },
        "records": {
          "type": "integer"
        }
      }
    }
  }
}
```

```
    type: "integer"
  },
  "bytes": {
    "type": "integer"
  },
  "packets": {
    "type": "integer"
  },
  "flows": {
    "type": "integer"
  },
  "connections": {
    "type": "integer"
  },
  "trafficRateMin": {
    "type": "integer"
  },
  "trafficRateMax": {
    "type": "integer"
  },
  "trafficRateAvg": {
    "type": "integer"
  },
  "trafficRate95th": {
    "type": "integer"
  },
  "packetRateMin": {
    "type": "integer"
  },
  "packetRateMax": {
    "type": "integer"
  },
  "packetRateAvg": {
    "type": "integer"
  },
  "packetRate95th": {
    "type": "integer"
  },
  "hosts": {
    "type": "integer"
  },
  "hostFlows": {
    "type": "integer"
  },
  "hostConnections": {
    "type": "integer"
  },
  "hostClients": {
    "type": "integer"
  },
  "hostClientBytes": {
    "type": "integer"
  },
  "hostClientPackets": {
    "type": "integer"
  },
  "hostServers": {
    "type": "integer"
  },
  "hostServerBytes": {
    "type": "integer"
  },
  "hostServerPackets": {
    "type": "integer"
  },
  "peers": {
    "type": "integer"
  },
  "peerFlows": {
    "type": "integer"
  },
  "peerConnections": {
    "type": "integer"
  },
  "peerClients": {
    "type": "integer"
  },
  "peerClientBytes": {
    "type": "integer"
  },
  "peerClientPackets": {
    "type": "integer"
  },
  ,
```



```
    },
    "peerServers": {
      "type": "integer"
    },
    "peerServerBytes": {
      "type": "integer"
    },
    "peerServerPackets": {
      "type": "integer"
    },
    "service": {
      "type": "object",
      "properties": {
        "id": {
          "type": "integer"
        },
        "name": {
          "type": "string"
        }
      },
      "required": [
        "id",
        "name"
      ]
    },
    "hostBytes": {
      "type": "integer"
    },
    "hostBytesRatio": {
      "type": "number"
    },
    "hostPackets": {
      "type": "integer"
    },
    "peerBytes": {
      "type": "integer"
    },
    "peerBytesRatio": {
      "type": "number"
    },
    "peerRole": {
      "type": "string"
    },
    "peerPackets": {
      "type": "integer"
    },
    "clientBytesRatio": {
      "type": "number"
    },
    "serverBytesRatio": {
      "type": "number"
    },
    "hostRole": {
      "type": "string"
    }
  },
  "required": [
    "hostServerBytes",
    "peerFlows",
    "packetRateMin",
    "packets",
    "rank",
    "connections",
    "trafficRateMin",
    "hostClientPackets",
    "service",
    "hostClientBytes",
    "percent",
    "peerClientPackets",
    "trafficRateMax",
    "trafficRate95th",
    "hostFlows",
    "packetRateMax",
    "peerConnections",
    "flows",
    "records",
    "hostServerPackets",
    "deviceId",
    "hostServers",
    "peerServers",
    "peers",
    "packetRate95th",
```

```

        "trafficRateAvg",
        "peerServerBytes",
        "hostConnections",
        "peerClients",
        "bytes",
        "peerClientBytes",
        "peerServerPackets",
        "hosts",
        "hostClients",
        "packetRateAvg",
        "hostBytes",
        "hostBytesRatio",
        "hostPackets",
        "peerBytes",
        "peerBytesRatio",
        "peerRole",
        "peerPackets",
        "clientBytesRatio",
        "serverBytesRatio",
        "hostRole"
    ]
}
},
"required": [
    "others",
    "results",
    "summary"
]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data" : {
    "summary": {
      "deviceId": 622,
      "percent": 100.0,
      "records": 4161574,
      "count": 237759,
      "bytes": 2377479875358,
      "packets": 1649544432,
      "flows": 1240477,
      "connections": 3151300,
      "trafficRateMin": 0.0,
      "trafficRateMax": 0.0,
      "trafficRateAvg": 0.0,
      "trafficRate95th": 0.0,
      "packetRateMin": 0.0,
      "packetRateMax": 0.0,
      "packetRateAvg": 0.0,
      "packetRate95th": 0.0,
      "hosts": 237759,
      "hostFlows": 620261,
      "hostConnections": 0,
      "hostClients": 0,
      "hostClientBytes": 451431872505,
      "hostClientPackets": 0,
      "hostServers": 0,
      "hostServerBytes": 737308065174,
      "hostServerPackets": 0,
      "peers": 644617,
      "peerFlows": 620261,
      "peerConnections": 0,
      "peerClients": 0,
      "peerClientBytes": 451431872505,
      "peerClientPackets": 0,
      "peerServers": 0,
      "peerServerBytes": 737308065174,
      "peerServerPackets": 0,
      "hostBytes": 1188739937679,
      "hostPackets": 0,
      "peerBytes": 1188739937679,
      "peerPackets": 0,
      "peerRole": "CLIENT_AND_SERVER",
      "hostBytesRatio": 50.0
    }
  }
}

```

```
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 37.97566298533009,
    "serverBytesRatio": 62.02433701466991,
    "hostRole": "CLIENT_AND_SERVER"
  },
  "others": {
    "deviceId": 622,
    "percent": 8.953520673942462,
    "records": 3433332,
    "count": 237733,
    "bytes": 212868152159,
    "packets": 185557066,
    "flows": 957319,
    "connections": 3020024,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 237733,
    "hostFlows": 343269,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 54755140776,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 34553342244,
    "hostServerPackets": 0,
    "peers": 489948,
    "peerFlows": 614095,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 118072738050,
    "peerClientPackets": 0,
    "peerServers": 0,
    "peerServerBytes": 5486931089,
    "peerServerPackets": 0,
    "hostBytes": 1188739937679,
    "hostPackets": 0,
    "peerBytes": 1188739937679,
    "peerPackets": 0,
    "peerRole": "CLIENT_AND_SERVER",
    "hostBytesRatio": 50.0,
    "peerBytesRatio": 50.0,
    "clientBytesRatio": 37.97566298533009,
    "serverBytesRatio": 62.02433701466991,
    "hostRole": "CLIENT_AND_SERVER"
  },
  "results": [{
    "deviceId": 622,
    "rank": 1,
    "percent": 19.515433006646784,
    "records": 2437,
    "bytes": 463975492322,
    "packets": 403537191,
    "flows": 54,
    "connections": 0,
    "trafficRateMin": 0.0,
    "trafficRateMax": 0.0,
    "trafficRateAvg": 0.0,
    "trafficRate95th": 0.0,
    "packetRateMin": 0.0,
    "packetRateMax": 0.0,
    "packetRateAvg": 0.0,
    "packetRate95th": 0.0,
    "hosts": 1,
    "hostFlows": 41,
    "hostConnections": 0,
    "hostClients": 0,
    "hostClientBytes": 2922061722,
    "hostClientPackets": 0,
    "hostServers": 0,
    "hostServerBytes": 1843321,
    "hostServerPackets": 0,
    "peers": 8,
    "peerFlows": 13,
    "peerConnections": 0,
    "peerClients": 0,
    "peerClientBytes": 720570
```

```
"peerClientBytes": 110519,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 461050816700,
"peerServerPackets": 0,
"service": {
  "id": 49,
  "name": "xwindows"
},
"hostBytes": 1188739937679,
"hostPackets": 0,
"peerBytes": 1188739937679,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 37.97566298533009,
"serverBytesRatio": 62.02433701466991,
"hostRole": "CLIENT_AND_SERVER"
}, {
"deviceId": 622,
"rank": 2,
"percent": 19.514754852347053,
"records": 2145,
"bytes": 463959369340,
"packets": 403516018,
"flows": 24,
"connections": 402,
"trafficRateMin": 0.0,
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 1,
"hostFlows": 10,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 1764216,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 461048838949,
"hostServerPackets": 0,
"peers": 5,
"peerFlows": 14,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 2905101992,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 3664183,
"peerServerPackets": 0,
"service": {
  "id": 4,
  "name": "https"
},
"hostBytes": 1188739937679,
"hostPackets": 0,
"peerBytes": 1188739937679,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 37.97566298533009,
"serverBytesRatio": 62.02433701466991,
"hostRole": "CLIENT_AND_SERVER"
}, {
"deviceId": 622,
"rank": 3,
"percent": 11.84168661371347,
"records": 1161,
"bytes": 281533716144,
"packets": 89807482,
"flows": 134,
"connections": 328,
"trafficRateMin": 0.0,
"trafficRateMax": 0.0,
"trafficRateAvg": 0.0,
"trafficRate95th": 0.0,
"packetRateMin": 0.0,
```

```
"packetRateMax": 0.0,
"packetRateAvg": 0.0,
"packetRate95th": 0.0,
"hosts": 1,
"hostFlows": 104,
"hostConnections": 0,
"hostClients": 0,
"hostClientBytes": 73678643668,
"hostClientPackets": 0,
"hostServers": 0,
"hostServerBytes": 101668268769,
"hostServerPackets": 0,
"peers": 5,
"peerFlows": 30,
"peerConnections": 0,
"peerClients": 0,
"peerClientBytes": 1465251170,
"peerClientPackets": 0,
"peerServers": 0,
"peerServerBytes": 104721552537,
"peerServerPackets": 0,
"service": {
  "id": 38,
  "name": "netflow"
},
"hostBytes": 1188739937679,
"hostPackets": 0,
"peerBytes": 1188739937679,
"peerPackets": 0,
"peerRole": "CLIENT_AND_SERVER",
"hostBytesRatio": 50.0,
"peerBytesRatio": 50.0,
"clientBytesRatio": 37.97566298533009,
"serverBytesRatio": 62.02433701466991,
"hostRole": "CLIENT_AND_SERVER"
}]
}
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

## Internal Hosts

`/tenants/{tenantId}/internalHosts/tags`

GET

Retrieves all Internal Host Tags for the specific Tenant (tenantId).

GET `/tenants/{tenantId}/internalHosts/tags`

### Request

#### URI Parameters

- **tenantId**: required (string)

#### Headers

- **Cookie**: required (string)

JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZSIsImNpdWUiOiJhcnRpdWUiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpcyI6ImVudCIsIm91dG8iOiJhcnRpdWUiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

Response

**HTTP status code 200**

**Body**

Type: application/json

**Schema:**

```
{
  "type": "object",
  "$schema": "http://json-schema.org/draft-03/schema",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": {
          "$ref": "tag"
        }
      }
    }
  }
}
```

**Example:**

```
{
  "data": [
    {
      "id": 27,
      "displayName": "Tag Name 1"
    },
    {
      "id": 28,
      "displayName": "Tag Name 2"
    }
  ]
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/internalHosts/tags/{tagId}

GET

Retrieves a single Internal Host Tag (tagId) given an ID for the specific Tenant (tenantId).

GET

/tenants/{tenantId}/internalHosts/tags/{tagId}

Request

**URI Parameters**

- **tenantId:** required (string)







```

        type: integer
      },
      "withinByteCount": {
        "type": "integer"
      },
      "granularity": {
        "type": "integer"
      }
    },
    "required": [
      "sentByteCount",
      "receivedByteCount",
      "withinByteCount"
    ]
  }
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



- **targetCategoryEvents:**

- **typeid:** The category event type ID where this host is the target of an alarm.
- **severity:** The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- **typeid:** The security event type ID where this host is the target of a security event.
- **severity:** The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- **alwaysBadCount:** The number of times an "always bad" alarm occurred for this security event with the host as the target.

### Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            }
          },
          "required": [
            "typeid",
            "severity",
            "alwaysBadCount"
          ]
        }
      }
    }
  }
}
```

```

    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
}
},
"required": [
  "hostname"

```

```

        header ,
        "data"
    ]
}
},
"required": [
    "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [

```

```
    "typeId": 47,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 56,
    "severity": 8.01,
    "alwaysBadCount": 0
  }
],
"sourceSecurityEvents": [
  {
    "typeId": 63,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 276,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
],
"targetCategoryEvents": [
  {
    "typeId": 46,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 32,
    "severity": 300.0,
    "alwaysBadCount": 0
  }
],
"targetSecurityEvents": [
  {
    "typeId": 286,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 267,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
]
}
]
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/internalHosts/tags/{tagId}/traffic/hourly

GET

Retrieves the hourly traffic trend for an Internal Host Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/internalHosts/tags/{tagId}/traffic/hourly





```
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
```

**Example:**



- **filter[startAbsolute]:** (*integer*)  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      }
    }
  }
}
```

```

    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
}
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  }
},
"data": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "string"
      },
      "value": {
        "type": "array",
        "items": {
          "type": {
            "oneOf": [
              {
                "$ref": "#/definitions/alarmTrend"
              },
              {
                "$ref": "#/definitions/trafficTrend"
              }
            ]
          }
        }
      }
    }
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}

```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/internalHosts/tags/{tagId}/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for an Internal Host Tag (tagId) of a Tenant (tenantId).

GET

/tenants/{tenantId}/internalHosts/tags/{tagId}/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)

- **tagId**: *required (integer)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiMjAxNjA5MjYyMjE0LjE2InQ=
```

#### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trends for 25 hours (current hour and past 24 hours).

The response is an array of traffic trends with each element representing a trend per application.

A trend has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.
- **applicationId**: ID of the application.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning

of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.

- **value**: For traffic trends, this element has the following aggregated values:

- **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not

classified by the tag.

- **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not

classified by the tag.

- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

#### Body

Type: application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    }
  },
  "required": [
    "sourceCount",
    "targetCount",
    "severity"
  ]
}
```

```

    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      },
      "required": [
        "timestamp",
        "value"
      ]
    }
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.





**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```

```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request



- **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

## Body

Type: application/json

## Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  }
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

```

    },
    {
      "$ref": "#/definitions/trafficTrend"
    }
  ]
},
{
  "required": [
    "timestamp",
    "value"
  ]
}
},
{
  "required": [
    "header",
    "data"
  ]
}
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

Retrieves the raw traffic trend of an application (applicationId) for an Internal Host Tag (tagId) of a Tenant (tenantId).

**GET** /tenants/{tenantId}/internalHosts/tags/{tagId}/applications/{applicationId}/traffic/raw

## Request

### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (integer)*
- **applicationId**: *required (integer)*

### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG96IiwiaWF0IjoiYXNjaWkiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjEiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
```

### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity**: This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

#### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
```



```

        "type": "integer"
    },
    "targetCount": {
        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [

```

```
required : [
  "header",
  "data"
]
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

`/tenants/{tenantId}/internalHosts/tags/tree`

GET

Retrieves all Internal Host Tags for a specific Tenant (tenantId) organized in a hierarchy.





```

        "sourceCount",
        "targetCount",
        "severity"
    ]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



- o **typeid**: The category event type ID where this host is the target of an alarm.
- o **severity**: The maximum severity level for this type of alarm category with the host as the target.

Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this alarm category with the host as the target.

- **targetSecurityEvents:**

- o **typeid**: The security event type ID where this host is the target of a security event.
- o **severity**: The maximum severity level for this type of security event with the host as the target.

Severity per security event is the ratio of the total number of points accumulated divided by the threshold.

- o **alwaysBadCount**: The number of times an "always bad" alarm occurred for this security event with the host as the target.

## Body

Type: application/json

### Schema:

```
{
  "id": "topHosts",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "header": {
          "type": "object",
          "properties": {
            "startTime": {
              "type": "string"
            },
            "endTime": {
              "type": "string"
            }
          }
        },
        "required": [
          "startTime",
          "endTime"
        ]
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "hostGroupIds": {
            "type": "array"
          },
          "sourceCategoryEvents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "typeid": {
                  "type": "integer"
                },
                "severity": {
                  "type": "number"
                },
                "alwaysBadCount": {
                  "type": "integer"
                }
              }
            },
            "required": [
              "typeid",
              "severity",
              "alwaysBadCount"
            ]
          }
        }
      }
    }
  }
}
```



```
    },
    "sourceSecurityEvents": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "typeId": {
            "type": "integer"
          },
          "severity": {
            "type": "number"
          },
          "alwaysBadCount": {
            "type": "integer"
          }
        }
      },
      "required": [
        "typeId",
        "severity",
        "alwaysBadCount"
      ]
    }
  },
  "targetCategoryEvents": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        },
        "alwaysBadCount": {
          "type": "integer"
        }
      }
    },
    "required": [
      "typeId",
      "severity",
      "alwaysBadCount"
    ]
  }
},
"targetSecurityEvents": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "typeId": {
        "type": "integer"
      },
      "severity": {
        "type": "number"
      },
      "alwaysBadCount": {
        "type": "integer"
      }
    }
  },
  "required": [
    "typeId",
    "severity",
    "alwaysBadCount"
  ]
}
}
},
"required": [
  "ipAddress",
  "hostGroupIds",
  "sourceCategoryEvents",
  "sourceSecurityEvents",
  "targetCategoryEvents",
  "targetSecurityEvents"
]
}
}
},
"required": [
  "header",
  "data"
```

```

    data
  }
},
"required": [
  "data"
]
}

```

**Example:**

```

{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "ipAddress": "10.205.20.70",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 56,
            "severity": 8.01,
            "alwaysBadCount": 0
          }
        ],
        "sourceSecurityEvents": [
          {
            "typeId": 63,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 276,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ],
        "targetCategoryEvents": [
          {
            "typeId": 46,
            "severity": 0.0,
            "alwaysBadCount": 2
          },
          {
            "typeId": 32,
            "severity": 300.0,
            "alwaysBadCount": 0
          }
        ],
        "targetSecurityEvents": [
          {
            "typeId": 286,
            "severity": 300.0,
            "alwaysBadCount": 0
          },
          {
            "typeId": 267,
            "severity": 140.0,
            "alwaysBadCount": 0
          }
        ]
      },
      {
        "ipAddress": "10.205.30.123",
        "hostGroupIds": [
          1,
          65534
        ],
        "sourceCategoryEvents": [
          {
            "typeId": 47,

```

```
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 56,
    "severity": 8.01,
    "alwaysBadCount": 0
  }
],
"sourceSecurityEvents": [
  {
    "typeId": 63,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 276,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
],
"targetCategoryEvents": [
  {
    "typeId": 46,
    "severity": 0.0,
    "alwaysBadCount": 1
  },
  {
    "typeId": 32,
    "severity": 300.0,
    "alwaysBadCount": 0
  }
],
"targetSecurityEvents": [
  {
    "typeId": 286,
    "severity": 300.0,
    "alwaysBadCount": 0
  },
  {
    "typeId": 267,
    "severity": 140.0,
    "alwaysBadCount": 0
  }
]
}
]
}
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

`/tenants/{tenantId}/internalHosts/traffic/hourly`

GET

Retrieves the hourly traffic trend for Internal Host Tags of a Tenant (tenantId).

GET

`/tenants/{tenantId}/internalHosts/traffic/hourly`

Request



```

"trafficTrend": {
  "type": "object",
  "properties": {
    "sentByteCount": {
      "type": "integer"
    },
    "receivedByteCount": {
      "type": "integer"
    },
    "withinByteCount": {
      "type": "integer"
    },
    "granularity": {
      "type": "integer"
    }
  },
  "required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
  ]
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**



Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.

- **filter[endAbsolute]:** (*integer*)  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]:** (*integer*)  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]:** (*integer*)  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

## Response

### HTTP status code 200

If no *query parameters* are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime:** start time for the traffic trend.
- **endTime:** end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp:** For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the *\*interval\** (5 minute, 1 hour or 1 day) which is specified in the *\*value\**.
- **value:** For traffic trends, this element has the following aggregated values:
  - **outboundByteCount:** The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount:** The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount:** The number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - **granularity:** This value is provided only for the raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour, or 1 day intervals, so this field could have the value of 300, 3600, or 86400.

### Body

Type: application/json

### Schema:

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      }
    },
    "required": [
      "sourceCount",
      "targetCount",
      "severity"
    ]
  },
  "trafficTrend": {
    "type": "object",
    "properties": {
      "sentByteCount": {
        "type": "integer"
      },
      "receivedByteCount": {
        "type": "integer"
      },
      "withinByteCount": {
```

```

        "type": "integer"
    },
    "granularity": {
        "type": "integer"
    }
},
"required": [
    "sentByteCount",
    "receivedByteCount",
    "withinByteCount"
]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    }
},
"data": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string"
            },
            "value": {
                "type": "array",
                "items": {
                    "type": {
                        "oneOf": [
                            {
                                "$ref": "#/definitions/alarmTrend"
                            },
                            {
                                "$ref": "#/definitions/trafficTrend"
                            }
                        ]
                    }
                }
            }
        }
    },
    "required": [
        "timestamp",
        "value"
    ]
}
},
"required": [
    "header",
    "data"
]
}
}

```

**Example:**



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

#### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/internalHosts/applications/traffic/hourly

GET

Retrieves the hourly traffic trends of all applications for Internal Host Tags of a Tenant (tenantId).

GET

/tenants/{tenantId}/internalHosts/applications/traffic/hourly

#### Request

##### URI Parameters

- **tenantId**: required (string)



```

"type": "object",
"properties": {
  "sentByteCount": {
    "type": "integer"
  },
  "receivedByteCount": {
    "type": "integer"
  },
  "withinByteCount": {
    "type": "integer"
  },
  "granularity": {
    "type": "integer"
  }
},
"required": [
  "sentByteCount",
  "receivedByteCount",
  "withinByteCount"
]
},
"properties": {
  "header": {
    "type": "object",
    "properties": {
      "startTime": {
        "type": "string"
      },
      "endTime": {
        "type": "string"
      }
    }
  },
  "data": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "string"
        },
        "value": {
          "type": "array",
          "items": {
            "type": {
              "oneOf": [
                {
                  "$ref": "#/definitions/alarmTrend"
                },
                {
                  "$ref": "#/definitions/trafficTrend"
                }
              ]
            }
          }
        }
      }
    },
    "required": [
      "timestamp",
      "value"
    ]
  }
},
"required": [
  "header",
  "data"
]
}

```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}

```

---

#### HTTP status code 400

Bad or invalid request

---

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

---

#### HTTP status code 403

Forbidden.



**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {

```

```
        "$ref": "#/definitions/trafficTrend"
      }
    ]
  },
  "required": [
    "timestamp",
    "value"
  ]
}
},
"required": [
  "header",
  "data"
]
}
```

**Example:**

```

{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}

```

#### HTTP status code 400

Bad or invalid request



### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

## /tenants/{tenantId}/internalHosts/applications/{applicationId}/traffic/hourly

GET

Retrieves the hourly traffic trend of an application (applicationId) for Internal Host Tags of a Tenant (tenantId).

GET /tenants/{tenantId}/internalHosts/applications/{applicationId}/traffic/hourly

### Request

#### URI Parameters

- **tenantId**: *required (string)*
- **applicationId**: *required (integer)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiNjU5MjA0MDAwMCJ9
```

#### Query Parameters

- **filter[startAbsolute]**: *(integer)*  
Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- **filter[endAbsolute]**: *(integer)*  
End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- **filter[startRelative]**: *(integer)*  
Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- **filter[intervalLength]**: *(integer)*  
Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

### Response

#### HTTP status code 200

If no *query parameters* are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a **header** element with the following fields:

- **startTime**: start time for the traffic trend.
- **endTime**: end time for the traffic trend.

It has a **data** element which represents the time series data. Each element in the series has the following fields:

- **timestamp**: For hourly trend, this is the beginning of the hour. For the raw trend, this is the

```
beginning of the *interval* (5 minute, 1 hour or 1 day) which is specified in the *value*.
```

- **value**: For traffic trends, this element has the following aggregated values:
  - **outboundByteCount**: The number of bytes sent by the hosts (classified by the tag) to hosts not classified by the tag.
  - **inboundByteCount**: The number of bytes received by the hosts (classified by the tag) from hosts not classified by the tag.
  - **withinByteCount**: The number of bytes transmitted (sent/received) within the hosts classified by the tag.

**Body****Type:** application/json**Schema:**

```
{
  "id": "timeSeries",
  "$schema": "http://json-schema.org/draft-03/schema",
  "type": "object",
  "definitions": {
    "alarmTrend": {
      "type": "object",
      "properties": {
        "sourceCount": {
          "type": "integer"
        },
        "targetCount": {
          "type": "integer"
        },
        "severity": {
          "type": "number"
        }
      },
      "required": [
        "sourceCount",
        "targetCount",
        "severity"
      ]
    },
    "trafficTrend": {
      "type": "object",
      "properties": {
        "sentByteCount": {
          "type": "integer"
        },
        "receivedByteCount": {
          "type": "integer"
        },
        "withinByteCount": {
          "type": "integer"
        },
        "granularity": {
          "type": "integer"
        }
      },
      "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
      ]
    }
  },
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "startTime": {
          "type": "string"
        },
        "endTime": {
          "type": "string"
        }
      }
    },
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "string"
          },
          "value": {
            "type": "array",
            "items": {
              "type": {
                "oneOf": [
                  {
                    "$ref": "#/definitions/alarmTrend"
                  },
                  {
                    "$ref": "#/definitions/trafficTrend"
                  }
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

```
GET /tenants/{tenantId}/internalHosts/applications/{applicationId}/traffic/raw

}
]
}
},
"required": [
  "timestamp",
  "value"
]
}
},
"required": [
  "header",
  "data"
]
}
}
```

**Example:**

```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900
        }
      }
    ]
  }
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.



```

        "type": "integer"
    },
    "severity": {
        "type": "number"
    }
},
"required": [
    "sourceCount",
    "targetCount",
    "severity"
]
},
"trafficTrend": {
    "type": "object",
    "properties": {
        "sentByteCount": {
            "type": "integer"
        },
        "receivedByteCount": {
            "type": "integer"
        },
        "withinByteCount": {
            "type": "integer"
        },
        "granularity": {
            "type": "integer"
        }
    },
    "required": [
        "sentByteCount",
        "receivedByteCount",
        "withinByteCount"
    ]
}
},
"properties": {
    "header": {
        "type": "object",
        "properties": {
            "startTime": {
                "type": "string"
            },
            "endTime": {
                "type": "string"
            }
        }
    },
    "data": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "timestamp": {
                    "type": "string"
                },
                "value": {
                    "type": "array",
                    "items": {
                        "type": {
                            "oneOf": [
                                {
                                    "$ref": "#/definitions/alarmTrend"
                                },
                                {
                                    "$ref": "#/definitions/trafficTrend"
                                }
                            ]
                        }
                    }
                }
            }
        },
        "required": [
            "timestamp",
            "value"
        ]
    }
}
},
"required": [
    "header",
    "data"
]
}

```

```
}  
}
```

**Example:**

```
{  
  "data": {  
    "header": {  
      "startTime": "2016-04-28T04:00:00Z",  
      "endTime": "2016-04-28T05:30:00Z"  
    },  
    "data": [  
      {  
        "timestamp": "2016-04-28T05:25:00Z",  
        "value": {  
          "outboundByteCount": 72294547625,  
          "inboundByteCount": 89717837575,  
          "withinByteCount": 12122355650,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T05:05:00Z",  
        "value": {  
          "outboundByteCount": 6103355775,  
          "inboundByteCount": 82136226125,  
          "withinByteCount": 11125132250,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T05:00:00Z",  
        "value": {  
          "outboundByteCount": 72294547625,  
          "inboundByteCount": 89717837575,  
          "withinByteCount": 12122355650,  
          "granularity": 300  
        }  
      },  
      {  
        "timestamp": "2016-04-28T04:00:00Z",  
        "value": {  
          "outboundByteCount": 5069793300,  
          "inboundByteCount": 88965886200,  
          "withinByteCount": 8615046900,  
          "granularity": 3600  
        }  
      }  
    ]  
  }  
}
```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

**Interface Status**

This is used to get Interface Status based on tenant to retrieve statistics for the interfaces on routers, switches (i.e., exporters) etc., that sends flow data to a Stealthwatch Flow Collector.



```

{
  "interfaceList": [
    {
      "swaId": 129,
      "exporterIpAddress": "10.100.100.7",
      "id": 1
    },
    {
      "swaId": 12,
      "exporterIpAddress": "10.100.100.7",
      "id": 2
    }
  ],
  "date": "2015-05-23",
  "direction": "INBOUND"
}

```

## Response

### HTTP status code 200

It has a **data** element with array of objects which has following fields.

- **domainId**: The domain in which the Stealthwatch Flow Collector exists.
- **deviceId**: Stealth watch appliance id.
- **retention**: Data retention period.
- **exporterIp**: Exporter IP Address.
- **interfaceId**: Interface id.
- **direction**: The direction of the traffic.
- **speed**: The speed (i.e., line speed) for traffic (direction is indicated by the Direction column) on the interface.
- **threshold**: Threshold limit.
- **currentTraffic**: The rate (in bits per second) at which the interface is currently receiving or sending (indicated by the Direction column) traffic..
- **maximumTraffic**: The peak rate (in bits per second) at which traffic has been received or sent (indicated by the Direction column) by the interface since the archive hour.
- **averageTraffic**: The average rate (in bits per second) at which traffic has been received or sent (indicated by the Direction column) by the interface since the archive hour.
- **currentPacketRate**: The rate (in packets per second) at which the interface is currently receiving or sending (indicated by the Direction column) packets.
- **maximumPacketRate**: The peak rate (in packets per second) at which packets have been received or sent (indicated by the Direction column) by the interface since the archive hour.
- **averagePacketRate**: The average rate (in packets per second) at which packets have been received or sent (indicated by the Direction column) by the interface since the archive hour.
- **trafficThreshold**: Traffic Threshold.
- **currentUtilization**: The current percent of available bandwidth consumed by inbound or outbound traffic.
- **maximumUtilization**: The peak percent of available bandwidth consumed by inbound or outbound traffic.
- **averageUtilization**: The average percent of available bandwidth consumed by inbound or outbound traffic.
- **date**: Date Time in ISO format.
- **name**: Interface Name.

### Body

Type: application/json

### Schema:

```

{
  "id": "interfaceStatusResponse",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "domainId": {
            "type": "integer"
          },
          "deviceId": {
            "type": "integer"
          },
          "retention": {
            "type": "integer"
          },
          "exporterIp": {
            "type": "string"
          },
          "interfaceId": {

```



```
    "type": "integer"
  },
  "direction": {
    "enum": "[INBOUND, OUTBOUND]"
  },
  "name": {
    "type": "string"
  },
  "speed": {
    "type": "number",
    "format": "long"
  },
  "threshold": {
    "type": "number",
    "format": "long"
  },
  "currentTraffic": {
    "type": "number",
    "format": "long"
  },
  "maximumTraffic": {
    "type": "number",
    "format": "long"
  },
  "averageTraffic": {
    "type": "number",
    "format": "long"
  },
  "currentPacketRate": {
    "type": "number",
    "format": "long"
  },
  "maximumPacketRate": {
    "type": "number",
    "format": "long"
  },
  "averagePacketRate": {
    "type": "number",
    "format": "long"
  },
  "trafficThreshold": {
    "type": "number",
    "format": "long"
  },
  "currentUtilization": {
    "type": "number",
    "format": "float"
  },
  "maximumUtilization": {
    "type": "number",
    "format": "float"
  },
  "averageUtilization": {
    "type": "number",
    "format": "float"
  },
  "date": {
    "type": "date",
    "example": "2017-02-27T16:29:00.000+0000",
    "format": "rfc3339"
  }
},
"required": [
  "domainId",
  "deviceId",
  "retention",
  "exporterIp",
  "interfaceId",
  "direction",
  "name",
  "speed",
  "threshold",
  "currentTraffic",
  "maximumTraffic",
  "averageTraffic",
  "currentPacketRate",
  "maximumPacketRate",
  "averagePacketRate",
  "trafficThreshold",
  "currentUtilization",
  "maximumUtilization",
  "averageUtilization",
```

```
        "date"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

**Example:**

```
{
  "data": [
    {
      "domainId": 123,
      "deviceId": 129,
      "retention": -1,
      "exporterIp": "10.203.0.1",
      "interfaceId": 0,
      "direction": "INBOUND",
      "speed": 1000000000,
      "threshold": 90,
      "currentTraffic": 0,
      "maximumTraffic": 0,
      "averageTraffic": 0,
      "currentPacketRate": 0,
      "maximumPacketRate": 0,
      "averagePacketRate": 0,
      "trafficThreshold": 900000000,
      "currentUtilization": 0,
      "maximumUtilization": 0,
      "averageUtilization": 0,
      "date": "2017-02-24T16:06:00.000+0000",
      "name": "V1"
    },
    {
      "domainId": 123,
      "deviceId": 129,
      "retention": -1,
      "exporterIp": "10.203.0.1",
      "interfaceId": 0,
      "direction": "OUTBOUND",
      "speed": 1000000000,
      "threshold": 90,
      "currentTraffic": 534187,
      "maximumTraffic": 886264,
      "averageTraffic": 554483,
      "currentPacketRate": 84,
      "maximumPacketRate": 116,
      "averagePacketRate": 86,
      "trafficThreshold": 900000000,
      "currentUtilization": 0.0534187,
      "maximumUtilization": 0.0886264,
      "averageUtilization": 0.055448300000000006,
      "date": "2017-02-24T16:06:00.000+0000",
      "name": "V1"
    }
  ]
}
```

**HTTP status code 400**

Invalid parameters or a token is missing (in which case the client needs to authenticate).

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

`/tenants/{tenantId}/netops/exporters/details/{isDetailsRequired}`

GET

Fetches all the exporters for a domain.

GET `/tenants/{tenantId}/netops/exporters/details/{isDetailsRequired}`

Request

#### URI Parameters

- **tenantId**: *required (string)*
- **isDetailsRequired**: *required (string)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZSIsImNpdWUiOiJkaW50b3R5IiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZSIsImNpdWUiOiJkaW50b3R5IiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.
```

## Response

### HTTP status code 200

It has a **data** element with array of exporters

- **swald**: Flow Collector id.
- **flowCollectorName**: Flow Collector Name.
- **name**: Exporter name.
- **ipAddress**: Exporter IP Address.
- **id**: Exporter Id.
- **type**: Exporter Type.
  - FLOW\_SENSOR\_AE
  - FLOW\_SENSOR\_VE
  - ASA\_FIREWALL
  - FIREWALL
  - EXPORTER
  - ASA\_ENH\_FIREWALL
- **detail**: An object contains additional detail. Optional field.
  - **alarms**: An array contains list of objects with type and severity, sorted in descending order by severity.
    - **type**: Alarm Type
      - **4020**: Interface utilization exceeded inbound
      - **4030**: Interface utilization exceeded outbound
      - **4010**: Flow Collector Flow Data Lost
      - **5999**: FlowSensor Management Channel Down
      - **5998**: FlowSensor Time Mismatch
      - **5011**: FlowSensor Traffic Lost
    - **severity**: Alarm Severity.
      - 1: Informational
      - 2: Trivial
      - 3: Minor
      - 4: Major
      - 5: Critical

#### Body

Type: application/json

Schema:

```

{
  "id": "exportersResponse",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "swaId": {
            "type": "integer"
          },
          "ipAddress": {
            "type": "string"
          },
          "flowCollectorName": {
            "type": "integer"
          },
          "type": {
            "enum": "[FLOW_SENSOR_VE,
FLOW_SENSOR_VE,ASA_FIREWALL,FIREWALL,EXPORTER,ASA_ENH_FIREWALL]"
          },
          "name": {
            "type": "string"
          },
          "detail": {
            "type": "object"
          },
          "id": {
            "type": "integer"
          }
        }
      },
      "required": [
        "swaId",
        "ipAddress",
        "flowCollectorName",
        "type",
        "name",
        "id"
      ]
    }
  },
  "required": [
    "data"
  ]
}

```

**Example:**

```

{
  "data": [
    {
      "swaId": 292,
      "ipAddress": "10.100.100.7",
      "flowCollectorName": "fc2",
      "name": "10.100.100.7",
      "type": "EXPORTER",
      "id": -1,
      "detail": {
        alarms: [{type: 4020, severity: 3}, {type: 4030, severity: 2}]
      }
    },
    {
      "swaId": 292,
      "ipAddress": "10.203.0.1",
      "flowCollectorName": "fc2",
      "name": "10.203.0.1",
      "type": "FLOW_SENSOR_VE",
      "id": -1,
      "detail": {
        alarms: [{type: 4020, severity: 3}, {type: 4030, severity: 2}]
      }
    }
  ]
}

```

**HTTP status code 400**

Invalid parameters or a token is missing (in which case the client needs to authenticate).



```

{
  "id": "interfacesResponse",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "swaId": {
            "type": "integer"
          },
          "exporterIpAddress": {
            "type": "string"
          },
          "flowCollectorName": {
            "type": "string"
          },
          "exporterName": {
            "type": "string"
          },
          "name": {
            "type": "string"
          },
          "interfaceId": {
            "type": "integer"
          }
        }
      },
      "required": [
        "swaId",
        "exporterIpAddress",
        "name",
        "interfaceId",
        "flowCollectorName",
        "exporterName"
      ]
    }
  },
  "required": [
    "data"
  ]
}

```

**Example:**

```

{
  "data": [
    {
      "swaId": 292,
      "flowCollectorName": "fcnf-01",
      "exporterIpAddress": "10.100.100.7",
      "exporterName": "exporter-1",
      "name": "Vl1",
      "interfaceId": 147
    },
    {
      "swaId": 292,
      "flowCollectorName": "fcnf-01",
      "exporterIpAddress": "10.100.100.7",
      "exporterName": "exporter-2",
      "name": "Vl2",
      "interfaceId": 146
    }
  ]
}

```

**HTTP status code 400**

Invalid parameters or a token is missing (in which case the client needs to authenticate).

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.



```

{
  "id": "security-events-templates",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "integer"
          },
          "name": {
            "type": "string"
          },
          "description": {
            "type": "string"
          }
        }
      },
      "required": [
        "id",
        "name",
        "description"
      ]
    }
  },
  "required": [
    "data"
  ]
}

```

**Example:**

```

{
  "data": [
    {
      "id": 30,
      "name": "High Traffic",
      "description": "The host traffic rate averaged over a 5-minute period has exceeded the limit of acceptable traffic values."
    },
    {
      "id": 19,
      "name": "SYNs Received",
      "description": "The host has received an excessive number of unanswered TCP connection requests (i.e., SYN packets) in a 5-minute period."
    }
  ]
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

**HTTP status code 500**

Internal Server error.

/tenants/{tenantId}/security-events/templates/{templateId}

POST

POST /tenants/{tenantId}/security-events/templates/{templateId}





```
{
  "details": [
    {
      "key": "baseline",
      "value": "100"
    },
    {
      "key": "tolerance",
      "value": "100"
    },
    {
      "key": "threshold",
      "value": "200"
    }
  ]
}
```

## Response

### HTTP status code 200

- The response has a **data** element that contains the following sections:
  - detailsText**: The populated security event details string for the related event and values.

#### Body

Type: application/json

#### Schema:

```
{
  "id": "security-events-templates",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "detailsText": {
          "type": "string"
        }
      },
      "required": [
        "detailsText"
      ]
    }
  },
  "required": [
    "data"
  ]
}
```

#### Example:

```
{
  "data": {
    "detailsText": "Expected 100 points, tolerance of 100 allows up to 200 points."
  }
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

**POST** /tenants/{tenantId}/security-events/queries

- Creates and starts a search for security events.
- All inputs are optional except **timeRange**.
  - **hosts**: The included host or hosts as well as the direction of each host.
    - **ipAddress**: Specify an IP address to be included.
      - Multiple IPs can be requested, but they need to be individual arrays.
    - **type**: indicate each host as "Source" or "Target".
      - If requesting an IP, this is required.
      - If needing to request same IP as a Source AND as a Target, include individual IP arrays for each.
  - **alarmCategoryId**: The alarm category ID to be included in the search.
  - **securityEventTypelds**: List of security event IDs to be included.
  - **timeRange**: Required active time range for the search. Results will be active within some portion of the time range, but they will not always be entirely contained within it.
    - **from**: Start time for search.
      - **Format**: rfc3339
      - Required field
    - **to**: End time for search.
      - **Format**: rfc3339
      - Required field
- **Note**: If any portion of the search is invalid, then that portion of the request will be dropped.

Request

**URI Parameters**

- **tenantId**: *required (string)*

**Headers**

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

**Example:**

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9hIiwiaWF0IjoiMTYxMjM0NTY3ODkwIn0.
```

**Body**

**Type:** application/json

**Schema:**

```

{
  "id": "security-events-templates",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "hosts": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "ipAddress": {
            "type": "string"
          },
          "type": {
            "type": "string"
          }
        }
      },
      "required": [
        "ipAddress"
      ]
    },
    "alarmCategoryId": {
      "type": "integer"
    },
    "securityEventTypeId": {
      "type": "array",
      "items": {
        "type": "integer"
      }
    },
    "timeRange": {
      "type": "object",
      "properties": {
        "from": {
          "type": "date",
          "format": "rfc3339"
        },
        "to": {
          "type": "date",
          "format": "rfc3339"
        }
      },
      "required": [
        "from",
        "to"
      ]
    }
  },
  "required": [
    "timeRange"
  ]
}

```

**Example:**

```

{
  "hosts": [
    {
      "ipAddress": "10.0.1.56",
      "type": "source"
    },
    {
      "ipAddress": "10.0.2.45",
      "type": "target"
    }
  ],
  "securityEventTypeId": [30, 31],
  "timeRange": {
    "from": "2017-03-17T17:54:41Z",
    "to": "2017-03-21T17:54:41Z"
  }
}

```

**Response**

**HTTP status code 200**

- The response has a data element that contains the following fields:
  - **searchJob:**

- **id**: Query ID of the triggered search job.
- **searchJobStatus**: The current status of the search. It will be one of the following:
  - **PENDING**: The search job has been created but may not have started.
  - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It has started but is not complete.
  - **WAITING**: The search job is WAITING. It will start to run at some point.
  - **COMPLETED**: The search job has been COMPLETED. Results will be available.
  - **FAILED**: The search job has FAILED. A portion of the job may have finished, but the entire job is not complete. You cannot restart this job.
- **percentComplete**: The percentage of the search that has completed.

#### Body

Type: application/json

#### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "searchJob": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "searchJobStatus": {
              "type": "string"
            },
            "percentComplete": {
              "type": "integer"
            }
          }
        },
        "required": [
          "id",
          "searchJobStatus",
          "percentComplete"
        ]
      }
    },
    "required": [
      "searchJob"
    ]
  },
  "required": [
    "data"
  ]
}
```

#### Example:

```
{
  "data": {
    "searchJob": {
      "id": "58b9394b60b2998c00cd63f7",
      "searchJobStatus": "IN_PROGRESS",
      "percentComplete": 0
    }
  }
}
```

#### HTTP status code 400

Bad or invalid request

#### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

#### HTTP status code 403

Forbidden.

#### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

## HTTP status code 500

Internal Server error.

## /tenants/{tenantId}/security-events/queries/{queryId}

GET

Gets the search status.

GET /tenants/{tenantId}/security-events/queries/{queryId}

- Gets the search status for the Security Events API.
- The request body is simply **queryId**, which is required.
  - **queryId**: Specific security event query job ID received from a /queries or /top request.

### Request

#### URI Parameters

- **tenantId**: *required (string)*
- **queryId**: *required (string)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZCIsImNpdWUiOiJhbnR5bWVudC5kaXN0aW50Lm9yZyIsImV4cCI6MTYxMjM0NTY3ODkwfQ.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZCIsImNpdWUiOiJhbnR5bWVudC5kaXN0aW50Lm9yZyIsImV4cCI6MTYxMjM0NTY3ODkwfQ.
```

### Response

#### HTTP status code 200

- The response has a **data** element that contains the following sections:
  - **status**: The current status of the search. It will be one of the following:
    - **PENDING**: The search job has been created but may not have started.
    - **IN\_PROGRESS**: The search job has been initiated and is IN\_PROGRESS. It has started but is not complete.
    - **WAITING**: The search job is WAITING. It will run at some point.
    - **COMPLETED**: The search job has been COMPLETED. Results will be available.
    - **FAILED**: The search job has FAILED. A portion of the job may have finished, but the entire job is not complete. You cannot restart this job.

#### Body

Type: application/json

#### Schema:

```
{
  "id": "security-events-templates",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "inProgress": {
          "type": "double"
        },
        "status": {
          "enum": "[IN_PROGRESS, COMPLETE]"
        }
      }
    },
    "required": ["status"]
  },
  "required": [
    "data"
  ]
}
```

#### Example:



- **target:** Information regarding the target host for this security event.
  - **ipAddress:** IP address of the related host. A value of 0.0.0.0 indicates multiple hosts.
  - **port:** Port location of the related host.
  - **protocol:** Protocol used by the related host.
  - **tags:** Indicates the host group(s) for which the given host is included. If there are multiple host groups, the array will include multiple host group arrays.
    - **type:** Name of a host group for which the related host is included.
    - **values:** Integer representation of the host group. This may be a string.
- **details:** This section will contain the additional details on this specific security event. Details will contain varying "key" types per security event type. Multiple Key/Value pairs will be listed as individual arrays.
  - **key:** A single security event attribute captured for the specific instance of the related security event (e.g., tolerance).
  - **value:** Related value for the specific key (e.g., 34).
- **hitCount:** Number of times this related event was seen.

## Body

Type: application/json

## Schema:

```
{
  "id": "security-events-templates",
  "$schema": "http://json-schema.org/draft-03/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "number",
            "format": "long"
          },
          "domainId": {
            "type": "integer"
          },
          "deviceId": {
            "type": "integer"
          },
          "securityEventType": {
            "type": "integer"
          },
          "firstActiveTime": {
            "type": "date",
            "format": "RFC3339"
          },
          "lastActiveTime": {
            "type": "date",
            "format": "RFC3339"
          },
          "source": {
            "type": "object",
            "properties": {
              "ipAddress": {
                "type": "string"
              },
              "ipRange": {
                "type": "string"
              },
              "port": {
                "type": "integer"
              },
              "protocol": {
                "type": "string"
              },
              "tags": {
                "type": "array",
                "items": {
                  "type": "object",
                  "properties": {
                    "name": {
                      "type": "string"
                    },
                    "id": {
                      "type": "integer"
                    }
                  }
                }
              }
            }
          }
        }
      }
    },
    "required": [
```



```
        "name",
        "id"
    ]
    }
},
"required": [
    "ipAddress",
    "port",
    "protocol",
    "hostGroup"
]
},
"target": {
    "type": "object",
    "properties": {
        "ipAddress": {
            "type": "string"
        },
        "ipRange": {
            "type": "string"
        },
        "port": {
            "type": "integer"
        },
        "protocol": {
            "type": "string"
        },
        "tags": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "type": {
                        "type": "string"
                    },
                    "values": {
                        "type": "array",
                        "items": {
                            "type": "integer"
                        }
                    }
                }
            },
            "required": [
                "type",
                "values"
            ]
        }
    },
    "required": [
        "ipAddress",
        "port",
        "protocol",
        "hostGroup"
    ]
},
"details": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "key": {
                "type": "string"
            },
            "value": {
                "type": "string"
            }
        },
        "required": [
            "key",
            "value"
        ]
    }
},
"hitCount": {
    "type": "integer"
},
"required": [
    "id",
    "domainId",
```

```

    "deviceId",
    "securityEventTypeId",
    "firstActiveTime",
    "lastActiveTime",
    "source",
    "target",
    "details",
    "hitCount"
  ]
}
},
"required": [
  "data"
]
}

```

**Example:**

```

{
  "data": {
    "results": [
      {
        "id": 23746,
        "domainId": 120,
        "deviceId": 129,
        "securityEventTypeId": 30,
        "firstActiveTime": "2017-03-17T17:54:41.375Z",
        "lastActiveTime": "2017-03-21T17:54:41.375Z",
        "source": {
          "ipAddress": "10.0.1.56",
          "port": 21,
          "protocol": "tcp",
          "tags": [
            {
              "name": "Catch All",
              "id": 65534
            }
          ]
        },
        "target": {
          "ipAddress": "10.0.2.0",
          "ipRange": "10.0.2.0/24",
          "port": 21,
          "protocol": "tcp",
          "tags": [
            {
              "name": "Catch All",
              "id": 65534
            }
          ]
        },
        "details": [
          {
            "key": "tolerance",
            "value": "34"
          },
          {
            "key": "serviceName",
            "value": "Service"
          }
        ],
        "hitCount": 10
      }
    ]
  }
}

```

**HTTP status code 400**

Bad or invalid request

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

**HTTP status code 403**

Forbidden.

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.



```
{
  "startDate": "2017-09-21T03:20:33Z",
  "endDate": "2017-09-21T15:20:33Z",
  "tagLimit": 10,
  "rootTagIds": [1, 0]
}
```

## Response

### HTTP status code 201

Successfully Created.

#### Body

Type: application/json

#### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "queryId": {
      "type": "string"
    }
  }
}
```

#### Example:

```
{
  "data": {
    "queryId": "58c163a760b25025de707e8b"
  }
}
```

### HTTP status code 400

Bad or invalid request

### HTTP status code 401

Expired or invalid token. Client should re-authenticate.

### HTTP status code 403

Forbidden.

### HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

### HTTP status code 500

Internal Server error.

/tenants/{tenantId}/tags/{tagId}/traffic/queries/{queryId}

GET

Get the status of the specified query ID.

GET

/tenants/{tenantId}/tags/{tagId}/traffic/queries/{queryId}

Returns the status of given query ID.

## Request

### URI Parameters

- **tenantId**: required (string)
- **tagId**: required (string)
- **queryId**: required (string)

### Headers

- **Cookie**: required (string)  
JSON Web Token for the authenticated user.



#### URI Parameters

- **tenantId**: *required (string)*
- **tagId**: *required (string)*
- **queryId**: *required (string)*

#### Headers

- **Cookie**: *required (string)*  
JSON Web Token for the authenticated user.

#### Example:

```
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRGVhZCIsInR5cCI6IkpXVCJ9.eyJ1b290aW50IjoiIn0=
```

## Response

### HTTP status code 200

Success.

#### Body

Type: application/json

#### Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "searchId": {
        "type": "string"
      },
    },
    "searchJobId": {
      "type": "string"
    },
    "resultDto": {
      "type": "array",
      "rootTagId": {
        "type": "integer"
      },
    },
    "tagTraffic": {
      "type": "array",
      "tagId": {
        "type": "integer"
      },
    },
    "totalTrafficBytes": {
      "type": "long"
    },
  },
  "remainderTagCount": {
    "type": "integer"
  },
  "remainderTagTrafficPercentage": {
    "type": "integer"
  },
  "createdOn": {
    "type": "date",
    "example": "yyyy-MM-dd'T'HH:mm:ss.SSS'ZZZZ'"
  },
  "accessTime": {
    "type": "date",
    "example": "yyyy-MM-dd'T'HH:mm:ss.SSS'ZZZZ'"
  },
  "completedTime": {
    "type": "date",
    "example": "yyyy-MM-dd'T'HH:mm:ss.SSS'ZZZZ'"
  },
  "id": {
    "type": "string"
  }
}
```

#### Example:

```
{
```

```
"data": [
  {
    "searchId": "58ef800660b237b646569c54",
    "searchJobId": "58ef800660b237b646569c55",
    "resultDto": [
      {
        "rootTagId": 1,
        "tagTraffic": [
          {
            "tagId": 58,
            "totalTrafficBytes": 1024250122556
          },
          {
            "tagId": 53,
            "totalTrafficBytes": 209832878292
          },
          {
            "tagId": 54,
            "totalTrafficBytes": 192742643200
          },
          {
            "tagId": 55,
            "totalTrafficBytes": 97934826030
          },
          {
            "tagId": 56,
            "totalTrafficBytes": 14970999833
          },
          {
            "tagId": 64,
            "totalTrafficBytes": 9369307430
          },
          {
            "tagId": 65534,
            "totalTrafficBytes": 25963182
          },
          {
            "tagId": 14,
            "totalTrafficBytes": 15237051
          },
          {
            "tagId": 59,
            "totalTrafficBytes": 7666922
          },
          {
            "tagId": 62,
            "totalTrafficBytes": 2136249
          }
        ],
        "remainderTagCount": 4,
        "remainderTagTrafficPercentage": 0
      },
      {
        "rootTagId": 0,
        "tagTraffic": [
          {
            "tagId": 61755,
            "totalTrafficBytes": 5157362366228
          },
          {
            "tagId": 61627,
            "totalTrafficBytes": 1002908924449
          },
          {
            "tagId": 61758,
            "totalTrafficBytes": 4731872
          },
          {
            "tagId": 61097,
            "totalTrafficBytes": 840216
          },
          {
            "tagId": 61008,
            "totalTrafficBytes": 300771
          },
          {
            "tagId": 60835,
            "totalTrafficBytes": 243320
          },
          {
            "tagId": 60769,
            "totalTrafficBytes": 130110
          }
        ]
      }
    ]
  }
]
```

```
    },
    {
      "tagId": 61319,
      "totalTrafficBytes": 58938
    },
    {
      "tagId": 61151,
      "totalTrafficBytes": 46839
    },
    {
      "tagId": 61527,
      "totalTrafficBytes": 41187
    }
  ],
  "remainderTagCount": 2,
  "remainderTagTrafficPercentage": 0
}
],
"createdOn": "2017-04-13T13:41:30.706+0000",
"accessTime": "2017-04-13T13:41:30.706+0000",
"completedTime": "2017-04-12T14:41:30.706+0000",
"id": "58ef800a60b237b646569c58"
}
]
```

---

**HTTP status code 400**

Bad or invalid request

---

**HTTP status code 401**

Expired or invalid token. Client should re-authenticate.

---

**HTTP status code 403**

Forbidden.

---

**HTTP status code 404**

Not Found. Invalid or inaccessible path parameters.

---

**HTTP status code 500**

Internal Server error.



# ABOUT THE STEALTHWATCH SYSTEM APIS

## Overview

---

The Stealthwatch System REST API consists of a collection of resources for developers, administrators, or partners that enable the functionality of the Stealthwatch System to be accessed programmatically. Since the Stealthwatch REST API is based on open standards, you can use any programming or scripting language you wish as long as it supports HTTP.

This document assumes you have some knowledge of using REST services and a basic understanding of the Hyper Text Transfer Protocol (HTTP). Experience using either curl or wget is highly beneficial. To access the REST API resource reference from a running SMC, go to <https://<smcip>/smc/restapi-docs/index.html>.

You may also be interested in the "SMC Web Services Programming Guide" which documents the SOAP interface to Stealthwatch. With the Web Services API that is compliant with the Simple Object Access Protocol (SOAP), administrators can use external applications, such as Security Information and Event Management (SIEM) systems, trouble-ticketing systems, and third-party reporting systems, to access data from the SMC. For more information, refer to the SMC Web Services Programming Guide.

## Using the Stealthwatch REST API

---

### Authentication

Before you can use the Stealthwatch REST API, you need to authenticate. The same credentials (login/password pair) you use to login to the user interface for Stealthwatch can be used for accessing the Stealthwatch REST API. If you do not have credentials, the first step is to contact your Stealthwatch administrator.

You authenticate by sending a POST request containing the password to Stealthwatch. Assuming your Stealthwatch Management Console (SMC) is at "smcaddress", the username is "jim" and the password is "password123", an example of using curl to authenticate is shown below:

```
curl -s -k -c cookies.txt -d  
"username=jim&password=password123"  
https://smcaddress/token/v2/authenticate Assuming the credentials  
are good, a user session is created and a cookie is returned in the file cookies.txt.
```

You will need to reference the cookie in subsequent calls. Note that the user session will expire after a period of inactivity - this is no different to user sessions initiated through logging in via the browser.

## HTTP Methods

The standard HTTP methods are used to access the Stealthwatch System REST API. The verbs are applied consistently across all resources and should be fairly familiar to any other REST APIs you may have used:

Desired Option	HTTP Verb to Use
Read	GET
Create	POST
Update	PUT
Delete	DELETE

Please note that not all resources support all HTTP methods. For example, some resources may only support GET.

## HTTP Status Codes

If the operation succeeded, expect a standard 200 OK response. You should always check the Status-Code field in the HTTP header. If it is anything other than 200, then the operation failed.

The error codes are in the following table:

HTTP Status Code	Description
400 Bad Request	General error when fulfilling the request that would cause an invalid state, for example, domain validation errors or missing data.
401 Unauthorized	Error code response for missing or invalid authentication token. It indicates that the request can be retried once a valid authentication token has been acquired.
404 Not Found	Used when the requested resource is not found, whether because it does not exist, or for security reasons, the service wants to mask its existence.

405 Method Not Allowed	A request was made of a resource using a request method not supported by that resource, for example, using GET on a write-only resource or using PUT on a read-only resource.
500 Internal Server Error	The general catch-all error when the server-side has an exception. In the event of a failure, the response body will contain a JSON response that should contain more information about what caused the operation to fail.

## Media Types

The Stealthwatch REST API returns HTTP responses in JSON format only.

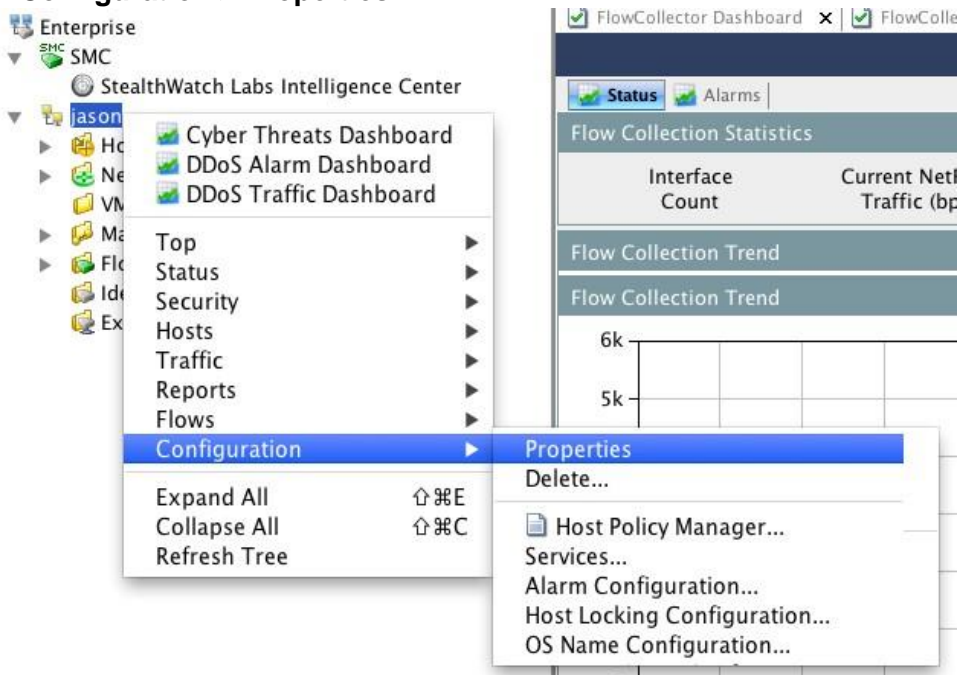
## Parameters

Several of the REST API calls may require identifiers of various objects in the system, e.g. domainId. This section describes how to obtain these identifiers by two methods.

### SMC Client

One way to obtain these identifiers is via the Stealthwatch Management Console (SMC) client as follows:

1. In the SMC client interface, elect your domain in the enterprise tree and then click **Configuration > Properties.**



*The Properties dialog opens.*



2. Click **Export** and select the "Export All configuration" option.
3. Save the XML configuration file.
4. After it is downloaded, open it with a text editor.
5. Locate the domainId by search for "<domain id".
6. Locate the hostGroupId by searching for "<host-group".
7. Locate the interface if-index by searching for "<interface if-index=".
8. Locate the exporterIp by searching for "<exporter ip=".

### **Command Line Interface**

You can also acquire parameter information from a CLI. Type this command to get a list of the host\_id from a Flow Collector:

```
grep id= /lancope/var/sw/today/config/groups.xml | awk  
-F\" '{print $2, $4}' |awk '$1<60000'|sort -k1,1n |less
```

To get the domain number for an SMC, type this command:

```
ls /lancope/var/smc/config/ | grep domain
```

## Copyrights and Trademarks

© 2017 Cisco Systems, Inc. All rights reserved.

### NOTICE

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

