



## Firehose Supporting APIs

---

- [Firehose Supporting APIs, on page 1](#)
- [Location Info, on page 1](#)
- [Map Image Info, on page 3](#)
- [Map Image , on page 3](#)
- [Devices Lookup, on page 4](#)
- [Create or Update MAC Filters, on page 4](#)
- [Get MAC Filters, on page 5](#)
- [Update Device Profile, on page 5](#)
- [Get Device Profile Data, on page 8](#)
- [Healthcheck Info, on page 9](#)
- [Network Topology, on page 10](#)

## Firehose Supporting APIs

Cisco Spaces - Data Firehose API provides various event types and their corresponding event data over the Cisco Spaces - Firehose Stream. If you wish to get data for specific entities such as a location, map, or even to address other requirements, you can utilize and leverage the power of the available supporting REST APIs described below.

### Location Info

**Supported App Types:** On-Prem and Cloud.

**Description:** Get location information by passing the location ID.

**Path:** /api/partners/v1/locations/:locationId

**Method:** GET

**Query Params:** partnerTenantId=<PARTNER\_TENANT\_ID>

**Headers:** X-API-KEY=<X-API-KEY>

**Response Codes**

- **404** – When location is not found.

- **403** – When the supplied partnerTenantId is not authorized to access the location.
- **200** – Success

### Response Headers

**CONTENT\_TYPE:** application/json

### Sample JSON:

```
{
  "location": {
    "locationId": "location-d827508f",
    "name": "Location Level 3",
    "inferredLocationTypes": [
      "FLOOR"
    ],
    "parent": {
      "locationId": "location-3e306fd1",
      "name": "Location Level 2",
      "inferredLocationTypes": [
        "NETWORK"
      ],
      "parent": {
        "locationId": "location-d4a2b651",
        "name": "Location Level 2",
        "inferredLocationTypes": [
          "CAMPUS"
        ],
        "parent": {
          "locationId": "location-da34853c",
          "name": "Location Level 1",
          "inferredLocationTypes": [
            "MSE"
          ],
          "parent": {
            "locationId": "location-91cdc6dc",
            "name": "Location Level Root",
            "inferredLocationTypes": [
              "ROOT"
            ]
          }
        }
      ],
      "locationDetails": {
        "timeZone": "Europe/London",
        "city": "",
        "state": "",
        "country": "",
        "category": "",
        "metadata": [
        ]
      }
    ],
    "latitude": 0.0,
    "longitude": 0.0
  }
}
```

---

## Map Image Info

**Supported App Types:** On-Premise, Cloud

**Description:** Get Map Image Info by passing map ID.

**Path:** /api/partners/v1/maps/:mapId

**Method:** GET

**Query Params:** partnerTenantId=<PARTNER\_TENANT\_ID>

**Headers:** X-API-KEY=<X-API-KEY>

### Response Codes

- **404** – When map ID is not found
- **200** – Success

### Response Headers

**CONTENT\_TYPE:** application/json

### Sample JSON:

```
{
  "mapId": "26752a276ac412620baf2822def1f523",
  "imageWidth": 856,
  "imageHeight": 591
}
```

---

## Map Image

**Supported App Types:** On-Premise, Cloud

**Description:** Get Map Image by passing map ID.

**Path:** /api/partners/v1/maps/:mapId/image

**Method:** GET

**Query Params:** partnerTenantId=<PARTNER\_TENANT\_ID>

**Headers:** X-API-KEY=<X-API-KEY>

### Response Codes

- **404** – When map ID is not found
- **200** – Success

### Response Headers

**CONTENT\_MD5:** md5 of an image

**ETAG :** eTag of an image

**LAST\_MODIFIED :** last modified date of an image

**CONTENT\_TYPE:** content type of an image

---

## Devices Lookup

**Supported App Types:** On-Premise, Cloud

**Description:** Get Device ID by passing remote IP and clientIP

**Path:** /api/partners/v1/devices/lookup

**Method:** GET

**Query Params:** partnerTenantId=<PARTNER\_TENANT\_ID>&remoteIP=<IP\_V4 of the remote server>&clientIP=<IP\_V4/IP\_V6 of the client>

**Headers:** X-API-KEY=<X-API-KEY>

### Response Codes

- **400** – When either remoteIP/clientIP is not supplied or is invalid
- **404** – When device not found.
- **403** – When partnerTenantId supplied is not authorized to access the device at the location
- **200** – Success

### Response Headers

**CONTENT\_TYPE:** application/json

**Sample JSON:**

```
{
  "deviceId": "device-kqtBjbyJS619EoqIVRBSF"
}
```

---

## Create or Update MAC Filters

To create or update MAC filters:

**URL:** /api/partners/v1/firehose/filters

**Method:** PUT

**URL Params Required:** partnerTenantId=[PARTNER\_TENANT\_ID]

**Body:**

```
{
  "macFilter": [
    "<Mac_Address/Pattern1>",
    "<Mac_Address/Pattern2>"
  ]
}
```

**Success Response****Code: 200** – Success**Sample Response****Error Response**

- **Code: 401 UNAUTHORIZED** – [API Key/Partner Tenant Id is not authorized ].
  - **Code: 404 BAD REQUEST** – [Invalid body json].
- 

## Get MAC Filters

To get MAC filters:

**URL:** /api/partners/v1/firehose/filters**Method:** GET**URL Params Required:** partnerTenantId=[PARTNER\_TENANT\_ID>**Success Response****Code: 200** – Success**Sample Response:**

```
{
  "macFilter": [
    "<Mac_Address/Pattern1>",
    "<Mac_Address/Pattern2>"
  ]
}
```

**Error Response**

- **Code: 401 UNAUTHORIZED** – [ API Key/Partner Tenant Id is not authorized ].
- 

## Update Device Profile

To update device profile data:

**URL:** /api/partners/v1/device**Method:** POST**URL Params Required:** partnerTenantId=[PARTNER\_TENANT\_ID>,  
macAddress=[MAC\_ADDRESS\_OF\_DEVICE] OR deviceId=[DEVICE\_ID]

**Body:**

```

{
  "firstName": "XXXX",
  "lastName": "YYYY",
  "gender": "male",
  "mobileInfo": {
    "number": "+917567387290",
    "verified": true,
    "optedIn": true
  },
  "emailInfo": {
    "address": "xyz@abc.com",
    "verified": true,
    "optedIn": true
  },
  "deviceType": "mobile",
  "addTags": [
    "tag1",
    "tag2",
    "tag3"
  ],
  "removeTags": [
    "tag4",
    "tag5",
    "tag6"
  ],
  "attributes": [
    {
      "name": "Device attribute 1",
      "values": [
        "device attr1",
        "device attr2"
      ]
    },
    {
      "name": "Device attribute 2",
      "values": [
        "device attr3",
        "device attr4"
      ]
    }
  ],
  "socialNetworkInfo": [
    {
      "socialNetwork": "facebook",
      "socialHandle": "facebook-handle-123",
      "socialInfo": {
        "facebook": {
          "id": "facebook-handle-123",
          "firstName": "xxxx",
          "lastName": "yyyy",
          "middleName": "zzzz",
          "name": "XYZ",
          "shortName": "xyz",
          "nameFormat": "xxx",
          "picture": "http://xyz.com",
          "email": "xyzfacebook@gmail.com",
          "attributes": [
            {
              "name": "Facebook attribute 1",
              "values": [
                "facebook attr1",
                "facebook attr2"
              ]
            }
          ]
        }
      }
    }
  ]
}

```

```

    ]
  },
  {
    "name": "Facebook attribute 2",
    "values": [
      "facebook attr3",
      "facebook attr4"
    ]
  }
]
}
},
{
  "socialNetwork": "twitter",
  "socialHandle": "twitter-handle-123",
  "socialInfo": {
    "twitter": {
      "id": "twitter-handle-123",
      "name": "XYZ",
      "screenName": "twitter screen name",
      "friendsCount": 10,
      "followersCount": 20,
      "profileImageUrl": "http://xyz.com",
      "profileBannerUrl": "http://xyz.com",
      "location": "location X",
      "statusesCount": 30,
      "email": "xyztwitter@gmail.com",
      "profileVerified": true,
      "utcOffset": "+5:30",
      "timeZone": "UTC",
      "geoEnabled": false,
      "lang": "EN",
      "attributes": [
        {
          "name": "Twitter attribute 1",
          "values": [
            "twitter attr1",
            "twitter attr2"
          ]
        },
        {
          "name": "Twitter attribute 2",
          "values": [
            "twitter attr3",
            "twitter attr4"
          ]
        }
      ]
    }
  }
},
{
  "socialNetwork": "linkedIn",
  "socialHandle": "linkedin-handle-123",
  "socialInfo": {
    "linkedIn": {
      "id": "linkedin-handle-123",
      "firstName": "xxxx",
      "lastName": "yyyy",
      "profilePicture": "http://xyz.com",
      "email": "xyzlinkedin@gmail.com",
      "attributes": [
        {

```



**Method:** GET

**URL Params Required:** partnerTenantId=[PARTNER\_TENANT\_ID>, macAddress=[MAC\_ADDRESS\_OF\_DEVICE] OR deviceId=[DEVICE\_ID]

**Success Response**

**Code: 200** – Success

**Sample Response:**

```
{
  "deviceId": "",
  "userId": "",
  "tags": [],
  "mobile": "",
  "email": "",
  "gender": "",
  "firstName": "",
  "lastName": "",
  "postalCode": "",
  "optIns": [],
  "otherFields": [],
  "macAddress": "",
  "manufacturer": "",
  "os": "",
  "osVersion": "",
  "type": "NOT_AVAILABLE",
  "socialNetworkInfo": []
}
```

**Error Response**

- **Code: 401 UNAUTHORIZED** – [API Key/Partner Tenant Id is not authorized].
- **Code: 404 NOT FOUND** – [Device not found for given macAddress or deviceId].

## Healthcheck Info

To get firehose health info:

**URL:** /api/partners/v1/firehose/health

**Method:** GET

**URL Params Required:** partnerTenantId=[PARTNER\_TENANT\_ID>

**Success Response**

**Code: 200** – Success

**Sample Response:**

```
{
  "timestamp": ,
  "activeConnections": [
  ],
  "protocolVersion": "",
  "appActivations": [
    {
```

```

        "spacesTenantName": "",
        "spacesTenantId": "",
        "partnerTenantId": "",
        "name": "",
        "referenceId": "",
        "instanceName": "",
        "macFilters": [
            ]
        }
    ]
}

```

#### Error Response

- **Code: 401 UNAUTHORIZED** – [API Key/Partner Tenant Id is not authorized].
  - **Code: 404 NOT FOUND** – [Device not found for given macAddress or deviceId].
- 

## Network Topology

**Supported App Types:** On-Premise

**Description:** Get Network Topology of a Tenant.

**Path:** /api/partners/v1/networks/topology

**Method:** GET

#### Response Codes

- **403** – When request is not authorized to access the location.

#### Response Headers

**CONTENT\_TYPE:** "application/json"

---