Configure and Troubleshoot AppDynamics API Client

Contents

Introduction

This document describes how to create an AppDynamics API client, generate Tokens, and troubleshoot issues.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- To create API Client, a User must have Account Owner (Default) role or a custom role with Administration, Agents, Getting Started Wizard permission.

Components Used

The information in this document is based on these software and hardware versions:

- AppDynamics Controller

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

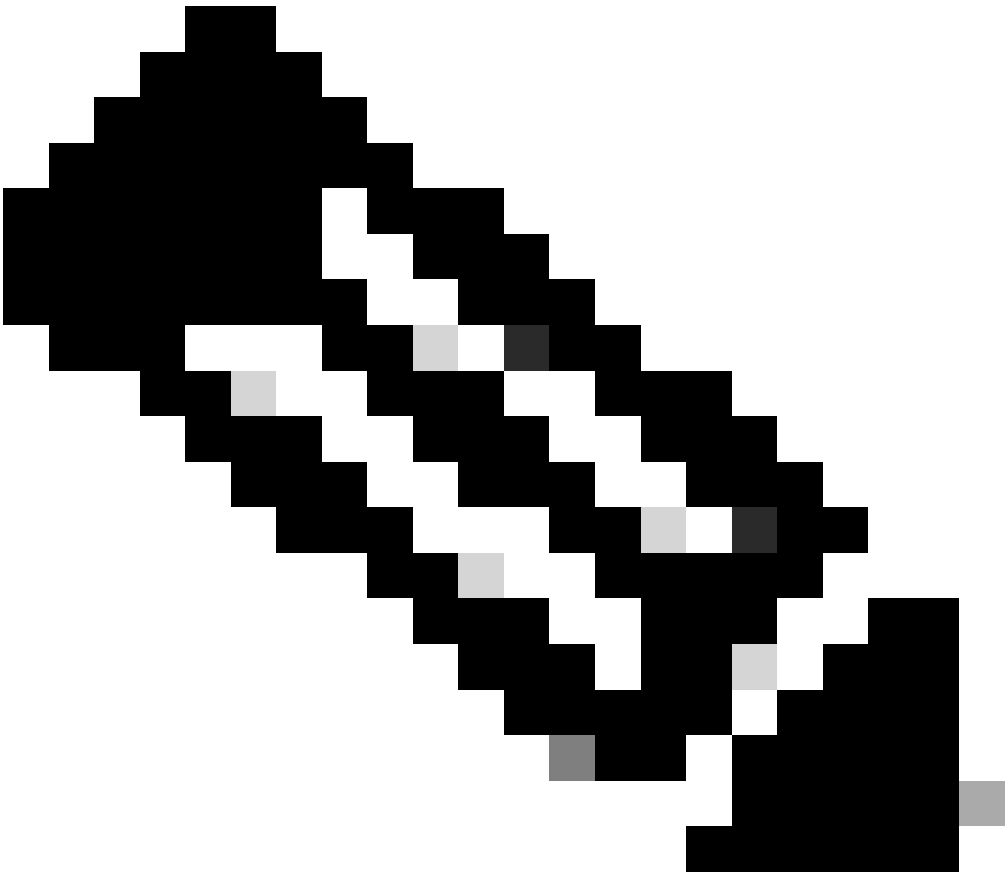
Background Information

This document describes the process for creating API Clients to securely access the data from the AppDynamics Controller using Representational State Transfer (REST) and Application Programming Interface (API) calls. The API Clients utilize Open Authorization (OAuth) token-based authentication. OAuth allows third-party services to access an end user account information without exposing the user credentials. It acts as an intermediary, providing the third-party service with an access token that authorizes the sharing of specific account information. Users can generate the OAuth token after setting up the API Client. Additionally, this document covers troubleshooting common issues encountered while using API Clients.

Configure

Create an API Client

1. Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
2. Click **User Name** (top right) **> Administration.**
3. Click API Client Tab.
4. Click **+ Create**.
5. Enter the **Client Name** and **Description**.
6. Click **Generate Secret** to populate the **Client Secret.**



Note: The client secret is generated and displayed only once. Copy and securely store this information.

7. Set the Default Token Expiration.

8. Click **+ Add** in Roles section to add the role.
9. Click **Save** at the top right.

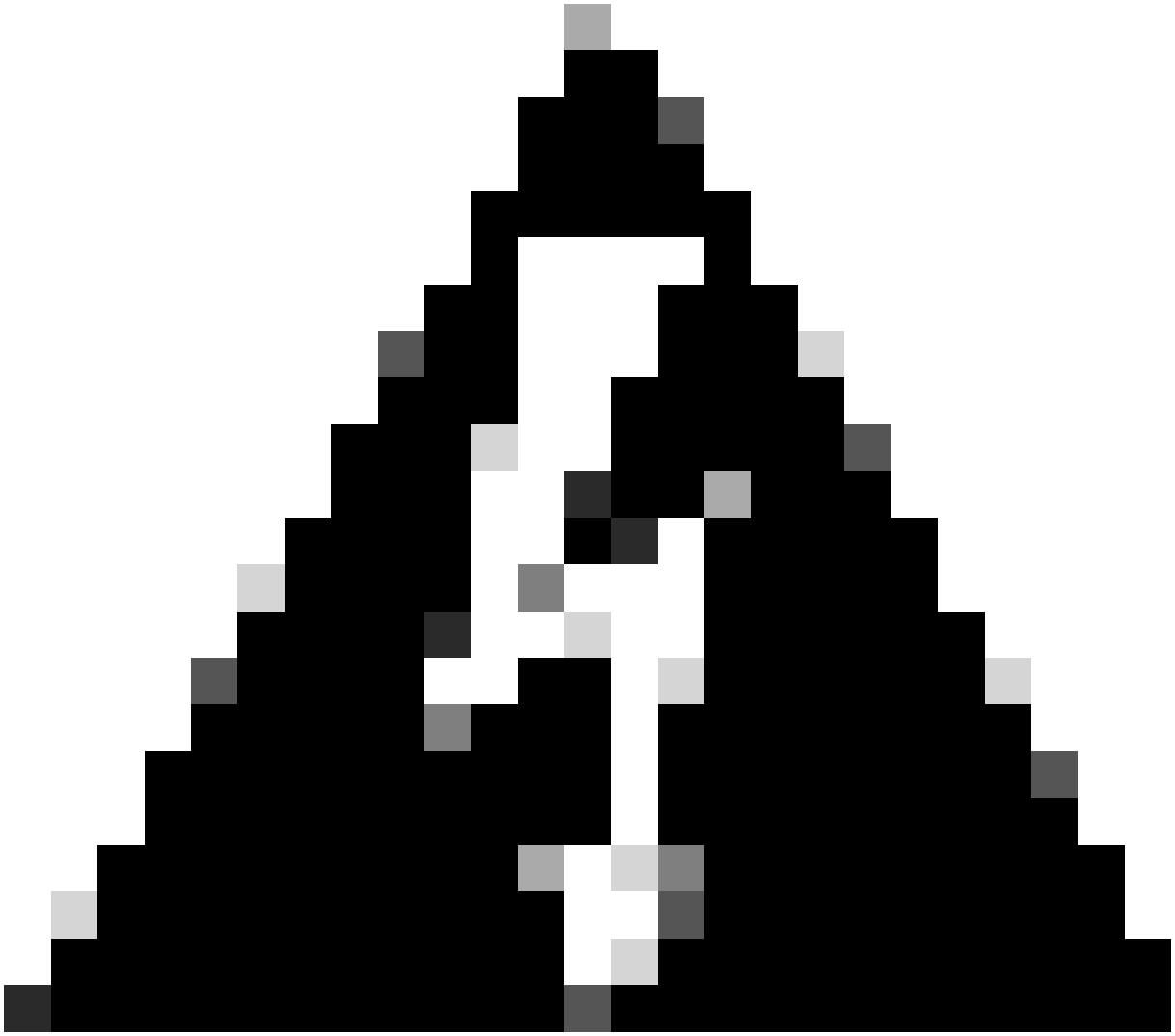
View Existing API Client

1. Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
2. Click **User Name** (top right corner) **> Administration.**
3. Click **API Client Tab** to view existing **API Clients.**

Delete Existing API Client

1. Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
2. Click your **User Name** (top right corner) **> Administration > API Clients.**
3. Find the specific API clients you want to delete and select them.

4. Click **Delete** icon or **Right Click** on the selected **API Client(s)** and Select **Delete API Client(s)** to delete the existing API Client(s).



Warning: Deleting the API Client invalidates the token.

Generate Access Token

The Access Token can be generated through the Administrator UI or the OAuth API. The UI provides long-lived tokens, while the OAuth API generates short-lived, regularly refreshed tokens.

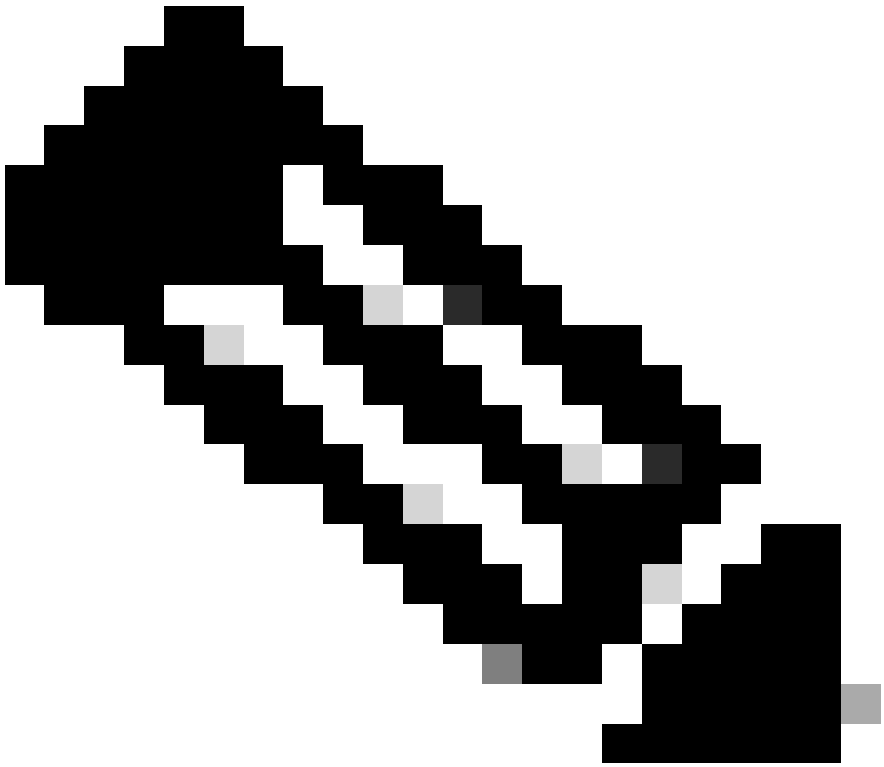
- **Administrator UI (Long-lived tokens)**
    - Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
    - Click your **User Name** (top right corner) **> Administration > API Clients.**
    - Select the API Client for which you want to generate the Access Token and click **Generate Temporary Access Token**.
    - The Access Tokens generated from the UI have a longer expiration time.

- **OAuth API (Short-lived tokens)**
  - `You can use REST APIs to generate a short-lived Access Token.

```
curl -X POST -H "Content-Type: application/x-www-form-urlencoded" "https://<controller addre
```

Replace:

- <apiClientName> with the Client Name that you entered while creating the API Client or as shared by your administrator.
- <accountName> with the Account Name.
- <clientSecret> with the Client Secret that you generated while creating the API Client or as shared by your administrator.



Note: On-demand token is not tracked on the UI.

Example Response:

```
{
"access_token": "<token>",
"expires_in": 300
}
```

Manange Access Tokens

- Access Tokens generated from the REST API can only be invalidated by deleting the associated API Client.
- Access Tokens generated through the Controller UI can be Revoked or Regenerated.
- Regenerating an Access Token does not invalidate the previous tokens. The older tokens remains active until its expiration.
- There is no way to retrieve previous or currently valid tokens. Therefore, only the current token can be revoked.

    - **Regenerate Access Token**

        - Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
        - Click your **User Name** (top right corner) > **Administration** > **API Clients.**
        - Select the API Client for which you want to regenerate the Access Token, Click **Regenerate** > **Save** (top right corner).

    - **Revoke Access Token**

        - Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
        - Click your **User Name** (Top Right Corner) > **Administration** > **API Clients.**
        - Select the API Client for which you want to Revoke the Access Token, Click **Revoke** > **Save** (top right corner).

Use Access Token to make Rest API

- From Splunk AppDynamics APIs determine the specific endpoint you need to interact with.
- Construct the Request:
    - Method: Select the HTTP method (GET, POST, PUT, DELETE) based on the action you want to perform.
    - Headers: Add the access token in the Authorization header.
    - Body (If any): Add the request body in JavaScript Object Notation (JSON) format.

- Example Request

```
curl --location --request GET 'https://<controller_address>/controller/<endpoint>' --header 'Autho
```

Replace:

- <controller_address> with the Controller URL.
- <endpoint> with the Rest Endpoint you need to interact with.
- <access_token> with the Access Token generated using the Client Name and Client Secret.

Common Problems and Solution

- **401 Unauthorized**
  - Issue: Users encounter a 401 Unauthorized error when attempting to generate an access token.
  - Sample Response:

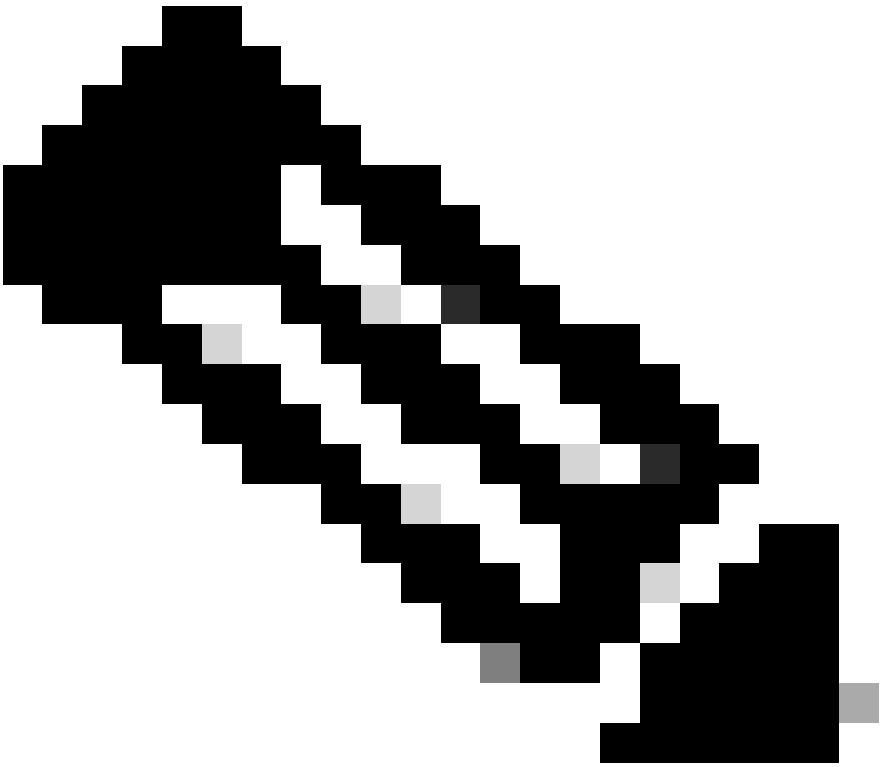
```
<body>
HTTP Error 401 Unauthorized
<p/>
This request requires HTTP authentication
</body>
```

  - Root Cause:The issue typically arises due to the client secret associated with the client name is invalid. This often happens when the client secret is generated but not saved
  - Solution:
    - Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
    - Click **User Name** (top right corner) **> Administration.**
    - Click **API Client Tab** to view existing API Clients.
    - Select the API Client for which you are getting the error.
    - Click **Generate Secret** to generate new Client Secret and Click **Save** (top right corner)

- **Empty Response.**
  - Issue: Users encounter an empty response when query a REST endpoint, even after generating an Access Token successfully.
  - Sample Response:

```
<applications></applications>
```

  - Root Cause: The issue typically arises due to insufficient roles or permissions assigned to the API client. Without the necessary roles, the API client cannot retrieve the expected data from the endpoint.
  - Solution:
    - Log in to the Controller UI as an Account Owner Role or a role with Administration, Agents, Getting Started Wizard permission.
    - Click **User Name** (top right corner) **> Administration.**
    - Click **API Client Tab** to view existing Api Clients.
    - Select the API Client for which you want to assign the Role
    - Click **+ Add** in Roles section to add the role.
    - Click **Save** at the top right.



Note: Ensure that the API Client has the appropriate roles assigned. Roles must align with the data access requirements of the REST endpoint.

- **Invalid Content Type**
  - Issue: User encounter a 500 Internal Server error when attempting to generate an Access Token.
  - Sample Error:

```
<h2>HTTP ERROR 500 javax.servlet.ServletException: java.lang.IllegalStateException: The @For
```

  - Root Cause: The issue arises due to content type header. In the Controller Version 24.10 the content type was changed from application/vnd.appd.cntrl+json;v=1 to application/x-www-form-urlencoded
  - Solution:
    - Modify the request and set content type header to application/x-www-form-urlencoded Example:

```
curl -X POST -H "Content-Type: application/x-www-form-urlencoded" "https://<controller
```

Related Information

AppDynamics Documentation

Splunk AppDynamics APIs

API Clients

Manage Access Tokens

Need Further Assistance?

If you have a question or experiencing issues, please create a support ticket with these details:

- **Error Details or Screenshot:** Provide specific error message or a screenshot of the problem.
- **Command Used:** Specify the exact command you were running when the issue occurred.
- **Controller Server.log (On-Prem only):** If applicable, provide the controller server logs from <controller-install-dir>/logs/server.log*