API Clients

You can create and use the identity type, API Clients, to provide secure access to the Controller through AppDynamics Controller REST API calls. These calls use Open Authorization (OAuth) token-based authentication.

About OAuth Mechanisms

OAuth is an open protocol to allow secure authorization in a simple and standard method from web, mobile, and desktop applications. See https://oauth.net/ for more information.

OAuth acts as the intermediary on your behalf, providing third-party applications with an access token that authorizes specific account information to be shared. Using the OAuth protocol with AppDynamics Controller REST APIs is the best way to securely grant access to your Controller information.

The OAuth authentication process works by first authenticating a request token. This request token is used to obtain an encrypted access token from your Controller. Once the access token is available, you can use it to make authenticated requests to your Controller until the token expires or is revoked.

The tokens are based on JSON Web Tokens (JWT) authentication format, which is the industry standard RFC 7519 method for representing claims securely between two parties.

Accessing Authentication Provider Settings
Creating API Clients
You can create new API Client identity types that can be used to generate OAuth tokens.

To create or edit an API Client

1. While logged in to the Controller UI as an Account Owner (or other role with the Administer users, groups, roles ...permission), click **Administer users, groups, roles** > Administration.
2. Click the **API Clients** tab to view the list of existing clients. From the tab, you can create new clients and modify or delete older ones.
3. Click + Create to create an API Client or select an existing client to edit.

   **Administration**

<table>
<thead>
<tr>
<th>Users</th>
<th>API Clients</th>
<th>Groups</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Create" /></td>
<td><img src="image" alt="API Clients" /></td>
<td><img src="image" alt="Groups" /></td>
<td><img src="image" alt="Roles" /></td>
</tr>
</tbody>
</table>

4. Enter the Client Name and Description.
5. Click Generate Secret to populate the Client Secret. This will generate a UUID as the secret of the API Client.

   - **This API Client secret acts as a password. It does not generate the authentication token.**

6. Set the Default API-generated Token Expiration. This expiration only applies to authentication tokens generated through the '/controller/api/oauth/access_token' REST API, not to Temporary Access Tokens generated from the UI. See Using the Access Token for more information.

   - **Every API-generated access token has an expiration. The default is 5 minutes, but you can set it to any second, minute, or hour limit. The Default API-generated Token has a shorter expiration than the authentication token generated through the Administration UI.**

7. Add the Roles you would like to associate with this API Client. You can add or remove roles at any time. See Roles and Permissions for more information.

   - **The REST APIs will use the identity which the access token represents to pull up RBAC permissions and check those permissions at the underlying API level.**
8. Click **Save** at the top right of the panel to save your API Client.

Using the Access Token

You can now generate and use the access token for each API access call into your Controller by generating the token through one of the following:

- **Administration UI**: These tokens are usually long-lived and have a longer expiration time. They can be generated by the account administrator and distributed to parties/teams who need to access the Controller but do not want to refresh such tokens too frequently.
- **API**: These tokens are usually short-lived and have a relatively short expiration time. They are typically generated and refreshed regularly by a program before expiration. These tokens are visible from the UI, and are not individually tracked and managed.

Generate the Token Through the UI

To generate the access token through the Administration UI:

1. Click **Generate Temporary Access Token** when you are ready to use the access token.

Remember that temporary token expirations are usually longer than the Default API-generated Token Expiration.

2. Copy and paste the access token from the Temporary Access Token field into the TOKEN portion of your curl command. See the following example:

   ```
   ```

3. Check that your call is successful. It should look like this:

   ```xml
   <Application>
     <ApplicationId>5</ApplicationId>
     <ApplicationName>Customer-Sales-MyClient</ApplicationName>
   </Application>
   ```

Generate the Token Through API
You can use REST API to generate a short-lived access token. This default, on-demand token is not tracked on the UI.

**To generate the access token through API:**

Use POST /controller/api/oauth/access_token. You will need the API Client name and Client Secret to generate the token:

**Example**

```bash
curl -X POST -H "Content-Type: application/vnd.appd.cntrl+protobuf;v=1" "https://<controller address>/controller/api/oauth/access_token" -d 'grant_type=client_credentials&client_id=<userName>@<accountName>&client_secret=face10d5-573e-4a75-8396-afa006f6b8f19'
```

**Example**

Use the generated token to list all applications:

```bash
```

In the case of success: all applications are returned.

In the case of failure: HTTP 401 Unauthorized "Failed to authenticate: invalid access token." is returned.

**Managing Access Tokens**

It is to be noted that:

- For controllers before 4.5.9, authentication header is required in the request as shown below:

```bash
curl -X POST -H "Content-Type: application/vnd.appd.cntrl+protobuf;v=1" -u 'TestApiClient@customer:face10d5-573e-4a75-8396-afa006f6b8f19' "https://<controller address>/controller/api/oauth/access_token" -d 'grant_type=client_credentials&client_id=TestApiClient@customer&client_secret=face10d5-573e-4a75-8396-afa006f6b8f19'
```

- The token generation API requires both an authentication header with username and password as well as the OAuth request body to successfully request a token.

1. Use the generated token to list all applications:

**Example**

```bash
```

- In the case of success: all applications are returned.
- In the case of failure: HTTP 401 Unauthorized "Failed to authenticate: invalid access token." is returned.
Access tokens are based on JWT, so even if you decode them, you will not be able to see any sensitive information.

However, if for any reason you believe that your token has become compromised, then you can revoke it by clicking **Revoke**. Deleting the API Client will also invalidate the token. Calls using revoked access tokens fail to authenticate with a 401 Unauthorized error HTTP status code. You can also click **Regenerate** to refresh a token.

Regenerated tokens do not disable older tokens. The older tokens will remain active until they expire.

When you regenerate a token, you can set the Temporary Token Expiration.

The default is 1 day, but you can set it to any hour, day, or year limit.

Regenerated tokens do not disable older tokens. The older tokens will remain active until they expire.

There is currently no way to retrieve all previous or currently valid tokens. Therefore, only the current token can be revoked.

Also, API generated tokens, which have Default API-generated Token Expiration, cannot be viewed nor revoked through the UI or REST API.