Use REST APIs with Azure Digital Twins

7 minutes

In this unit, learn about the Azure Digital Twins REST APIs: what they're for and how to use them.

Why REST APIs?

Some application architectures don't support the Azure Digital Twins SDK. In such cases, you can use the Azure Digital Twins REST APIs to perform both control-plane operations (for example, to create Azure Digital Twins instances) and data-plane operations (for example, ingesting telemetry or modifying Azure Digital Twins instance properties).

For more information, see the Azure Digital Twins REST APIs documentation.

API authentication

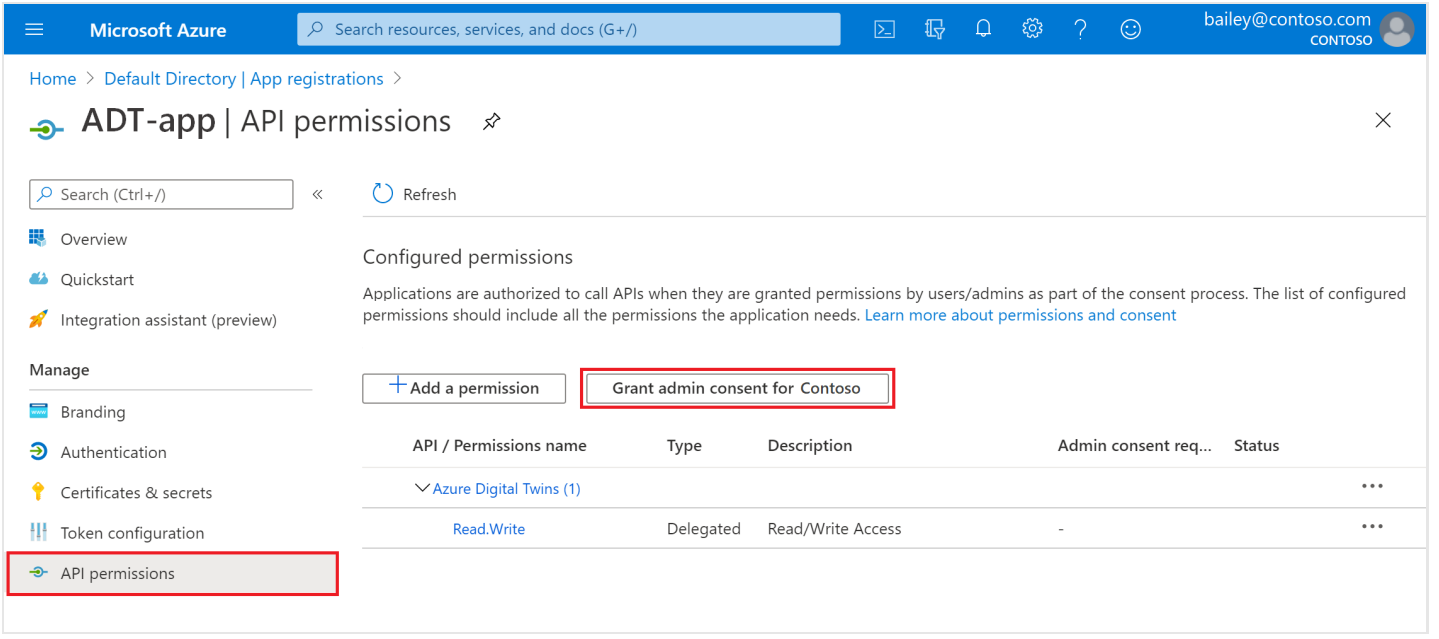
Use the OAuth 2.0 client-credentials grant, specified in RFC 6749 and sometimes called "two-legged OAuth," to access web-hosted resources by using the identity of an application. This type of grant is commonly used for server-to-server interactions that must run in the background, without immediate interaction with a user. These types of applications are often called daemons or service accounts.

For more information on different authentication patterns, see the Azure Active Directory documentation.

Grant admin consent

Your organization might require extra actions from subscription owners or administrators to successfully set up an app registration. The steps required might vary depending on your organization's specific settings. Admins perform such operations on the Azure Active Directory App registrations page in the Azure portal.

The **Grant admin consent for [*company*]** action is often required for app registration. Your organization might have **Admin Consent Required** turned on globally in Azure Active Directory (Azure AD) for all app registrations within your subscription. If so, for the app registration to be valid, the owner or administrator must select the button for your company on the API permissions page for that app registration:



Set up App Registration

In order to leverage the Azure Digital Twins APIs, an application will need to be defined with permissions to use the ADT APIs.

1. Create the manifest.json file for later use:

   ```
   touch manifest.json
   cat > manifest.json
   ```

   Now you're editing manifest.json.

2. Paste the following JSON code into PowerShell and then press `Ctrl+C` to close the file.

   ```
   [{
       "resourceAppId": "0b07f429-9f4b-4714-9392-
   cc5e8e80c8b0",
       "resourceAccess": [
        {
          "id": "4589bd03-58cb-4e6c-b17f-b580e39652f8",
          "type": "Scope"
        }
       ]
   }]
   ```

3. Create an Azure Active Directory (Azure AD) application with permissions to connect to Azure Digital Twins. You'll use this application in later units.

```
az ad app create --display-name $aaddtapp --native-app
--required-resource-accesses ./manifest.json --reply-
urls http://localhost -o json
```

4. The following command will output the application ID. Save this ID for later use.

```

$appid = $(az ad app list --display-name $aaddtapp --
query '[0].appId' -o json)
```

5. Create a service principal for the application by using the application ID from the preceding step:

```

az ad sp create --id $appid
```

6. The next command gives the application you just created permissions to the Azure Digital Twins instance. Add the application ID from the preceding steps to the command before you run it:

```

az dt role-assignment create --dt-name $dtname --as-
signee $appid --role "Azure Digital Twins Data Owner"
```

7. Create a password for the application:

> **Note**

Make sure to copy the password from the output because you can't retrieve it later. If you lose your password, you'll have to create a new one.

```
az ad app credential reset --id $appid --append
```

Collect important values

You'll need several important values from the resources set as you continue working with your Azure Digital Twins instance.

> **Note**
>
> Save these values for later use.

Collect instance values

Get the host name of the Azure Digital Twins instance. Copy the output to Notepad for later use.

```
az dt show -n $dtname --query 'hostName'
```

Collect app-registration values

Get the Azure AD tenant ID:

```
az account show --query 'tenantId'
```

POSTMAN

Postman is a REST testing tool that provides key HTTP request functionalities in a desktop and plugin-based GUI. You can use it to craft HTTP requests and submit them to the Azure Digital Twins REST APIs.

In the steps below, curl commands are provided. These commands can be imported into Postman by using the Import functionality.

Alternatively, install curl for windows or use a bash shell.

Retrieve token

To use the Azure Digital Twins APIs, you'll need to get an authorization token from Azure AD.

1. Use the following command to issue a POST request to the /oauth/v2.0/token endpoint for your Azure AD tenant:

   > **Note**
   >
   > Supply the values for *tenant_id*, *client_id*, and *client_secret* as appropriate for your configuration.

```
curl --location --request POST 'https://login.mi-
crosoftonline.com/tenant_id/oauth2/v2.0/token' \
--header 'Content-Type: application/x-www-form-urlen-
coded' \
--data-urlencode 'client_id= <applicationid>' \
--data-urlencode 'scope=https://digitaltwin-
s.azure.net/.default' \
--data-urlencode 'client_secret=<secret/pwd>' \
--data-urlencode 'grant_type=client_credentials'
```

2. Azure AD responds with a bearer token that will be used in the next section:

Use REST APIs

Run the following command to update the REST API property in the GrindingStep digital twin.

> **Note**
>
> Supply the values for *adt_hostname* and *aad_token*. Modify these values before importing into Postman

```
```azurecli
curl --location --request PATCH 'https://adt_host-
name/GrindingStep?api-version=2020-10-31' \
--header 'Authorization: Bearer aad_token' \
--header 'Content-Type: application/json' \
--data-raw '[
```
```

```
  {
    "op": "replace",
    "path": "/ChasisTemperature",
    "value": 62.3
  }
]'
```

Query the twin to see the update

To view the data added by the REST API call, run the following command:

> **Note**
>
> Supply the values for *adt_hostname* and *aad_token*. Modify these values before importing into postman.

```
    curl -i --location --request POST 'https://adt_host-
name/digitaltwins/query?api-version=2020-10-31' \
    --header 'Authorization: Bearer aad_token' \
    --header 'Content-Type: application/json' \
    --header 'Message-Id: 12345' \
    --data-raw '{
        "query": "SELECT * FROM DIGITALTWINS"
    }'
```