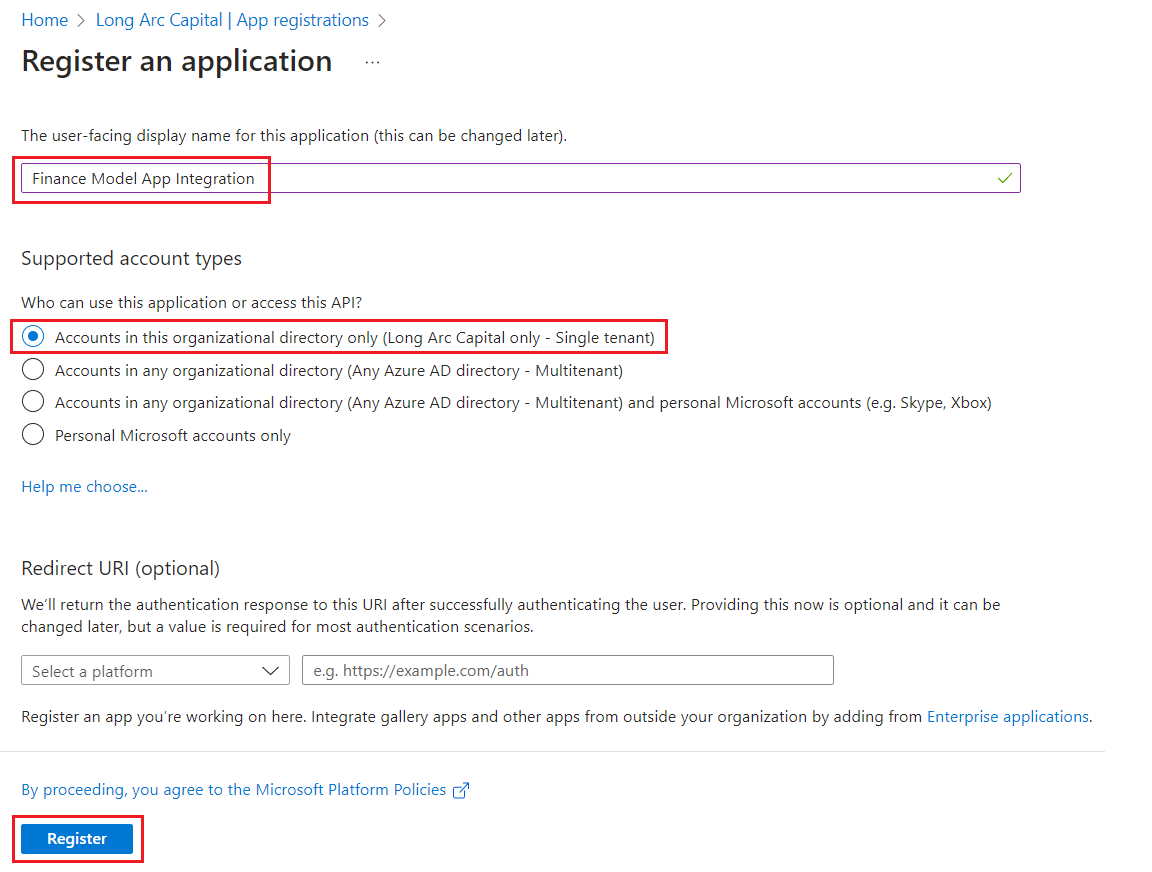
**Power BI Report Integration Using ReactJS**

In this article, you will learn how to integrate Power BI report or dashboard using ReactJS framework.

**Step 1: Register an application with Azure AD and create a service principal**

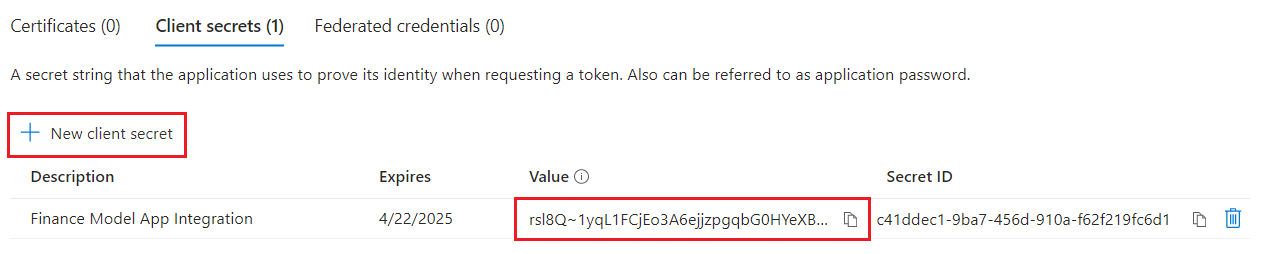
1. Sign-in to the Azure poral (<https://portal.azure.com>)
2. Search for and select **Azure Active Directory**.
3. Select **App registrations**, then select **New registration**.
4. Name the application, for example “Finance Model App Integration”.
5. Select a supported account type, which determines who can use the application. In our case, select “Accounts in this organizational directory only” which is default one.
6. Ignore Redirect URI.
7. Select **Register**.



You have created your Azure AD application and service principal.

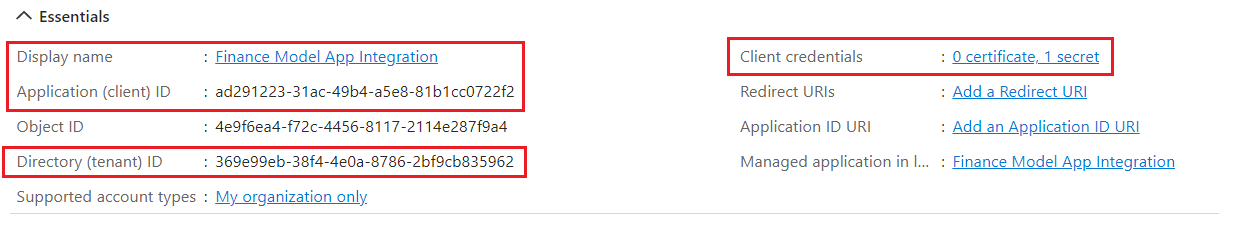
**Step 2: Add a client secret**

1. Search for and select **Azure Active Directory**.
2. Select **App registrations**, then select newly created application (Finance Model App Integration).
3. Search for and select **Certificates & secrets**, then select **New client secret**.
4. Enter a description for this client secret and set a expiry**.**
5. Select **Add**.



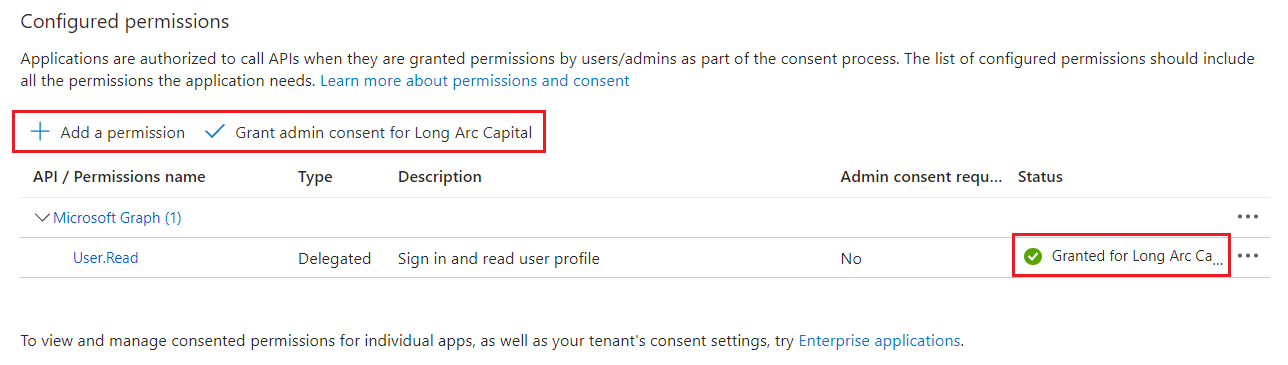
Copy the below information and keep with you for the next steps.

* Application (Client) Id
* Tenant Id
* Application (Display) Name
* Client Secret



**Step 3: Configure API Permissions**

1. Search for and select **Azure Active Directory**.
2. Select **App registrations**, then select newly created application (Finance Model App Integration).
3. Search for and select **API permissions**, then select **Add a permission**.
4. Select **Microsoft APIs**, then select **Microsoft Graph.**
5. Select **Delegated permissions**.
6. Search for and select **User.Read** under User permission section.
7. Select **Add permissions**.
8. Select Grant admin consent for Long Arc Capital, then select **Yes** from Grant admin consent confirmation.



**Step 4: Create Azure AD Group**

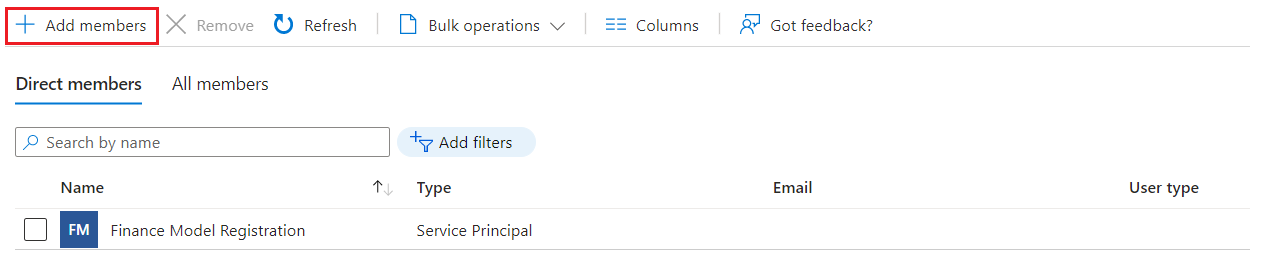
1. Search for and select **Groups**.
2. Select **New group**.
3. Select **Group type** as Security.
4. Enter the name of the group, for example “Finance Model App Group”**.**
5. Enter a description of the group.
6. Select **No** for Azure AD roles can be assigned to the group.
7. Select **Membership type** as Assigned.
8. Select **Create**.

Graphical user interface, text, application, email

Description automatically generated

**Step 5: Add Members to Azure AD Group**

1. Search for and select **Groups**.
2. Search for and select **Finance Model App Group** (newly created group).
3. Select **Members**, then select **Add members**.
4. Search for and select **Finance Model App Integration** (newly created service principal)**.**
5. Make sure selected items are added below.
6. Click **Select**.



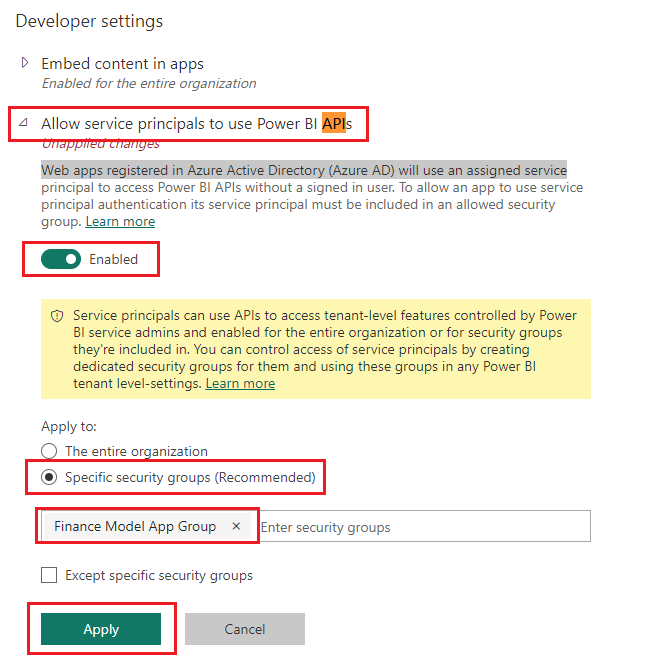
**Step 6: Add Azure AD group to specific security group in Power BI**

1. Sign-in to the Power BI (<https://app.powerbi.com>)
2. Click on three dots (…) and select **Settings**, then select **Admin portal.**
3. Select **Tenant settings** and go to **Developer settings**.
4. Toggle the **Enabled** slider under “Allow service principals to use Power BI APIs”.
5. Next, apply it to a “Specific security groups (Recommended)”, and add the security group you created (currently called **Finance Model App Group**).
6. Select **Apply**.

After clicking apply, the following message should appear. Keep in mind that this can have almost immediate effect or take as much as 15 minutes to complete.

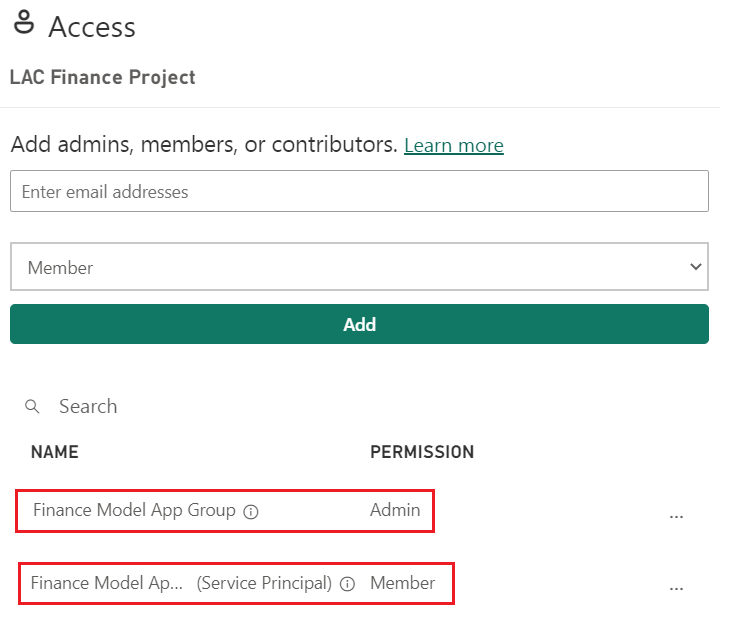
Graphical user interface, text, application

Description automatically generated



**Step 7: Grant access in workspaces in Power BI**

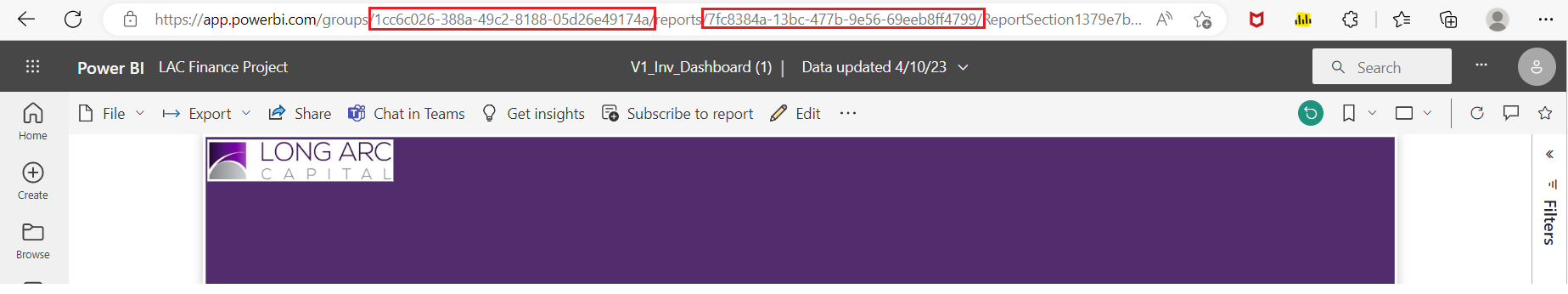
1. Go to Power BI home page, then select **Workspaces**.
2. Search for select your workspace.
3. Select **Access.**
4. Enter security group name (Finance Model App Group) and select **Admin** permission.
5. Select **Add.**
6. Enter service principal name (Finance Model App Integration) and select **Member** permission.
7. Select **Add**, then select Close**.**



**Step 8: Generate Power BI Embedding URL**

1. Go to Power BI rest API link (<https://learn.microsoft.com/en-us/rest/api/power-bi/reports/get-report-in-group>).
2. Select **Try It** Rest API.
3. Sign-in to the account using credentials**.**
4. Pass **groupId** and **reportId** under Parameters section.

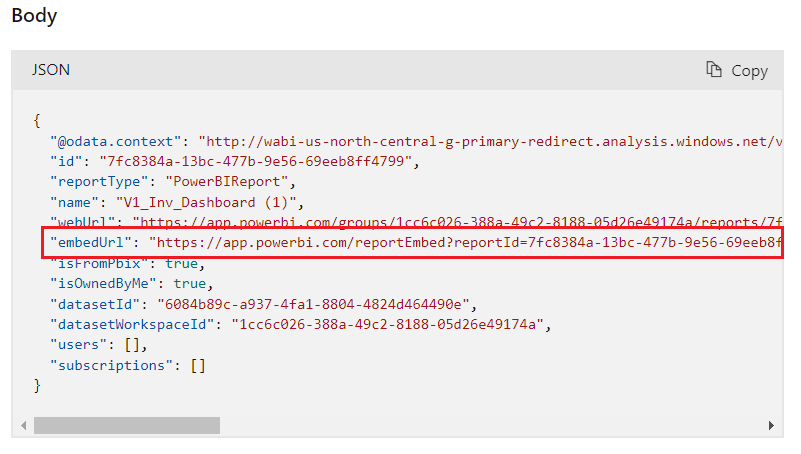
Note: You can copy groupId and reportId from Power BI workspace report url.



1. Select **Run.**
2. Copy **embedUrl** from body section and keep with you for next steps.

Graphical user interface, text, application, email

Description automatically generated



**Step 9: Generate Power BI Access Token using Rest API**

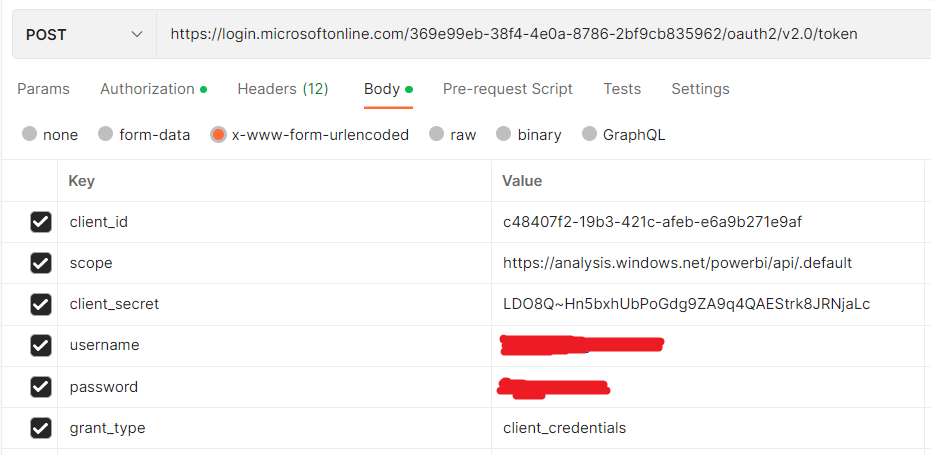
1. To start, make an HTTP Post call to the following URL:

<https://login.microsoftonline.com/tenant_id/oauth2/v2.0/token>

replace **tenant\_id** with generated Directory (tenant) ID in **Step 2.**

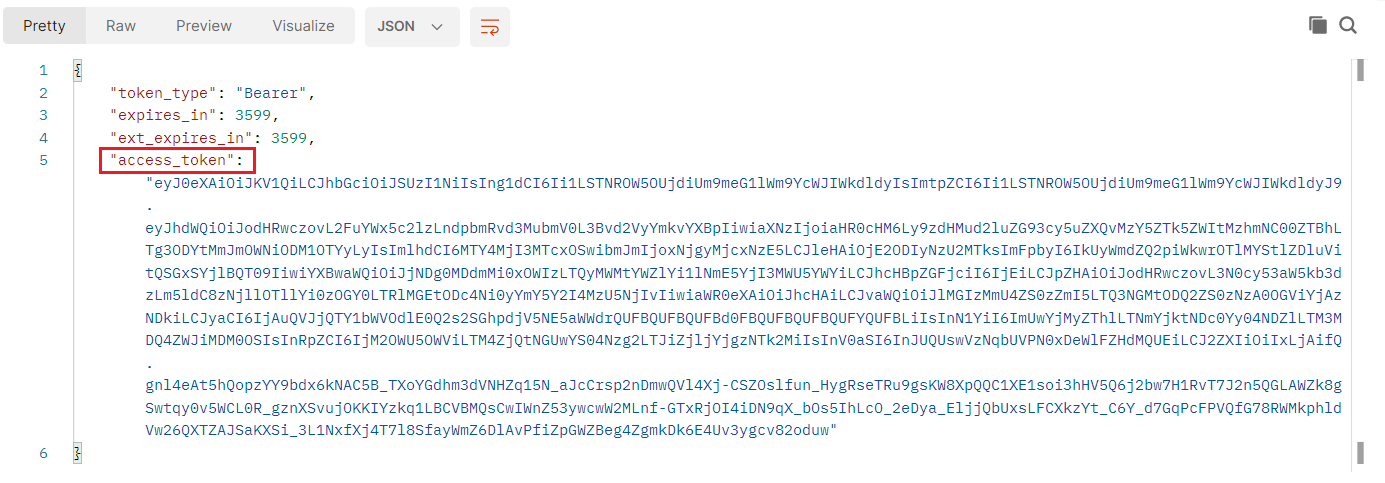
1. Input in the following parameters, using a x-www-form-urlencoded object in the body:

**Sample Request:**



* Use Azure portal credentials for username and password params.
* Replace client\_id, client\_secret values generated in **Step 2**.

**Sample Response (accessToken):**

****

This access\_token is only valid for 1 hour.

**Step 9: Generate Power BI Embed Token using Rest API**

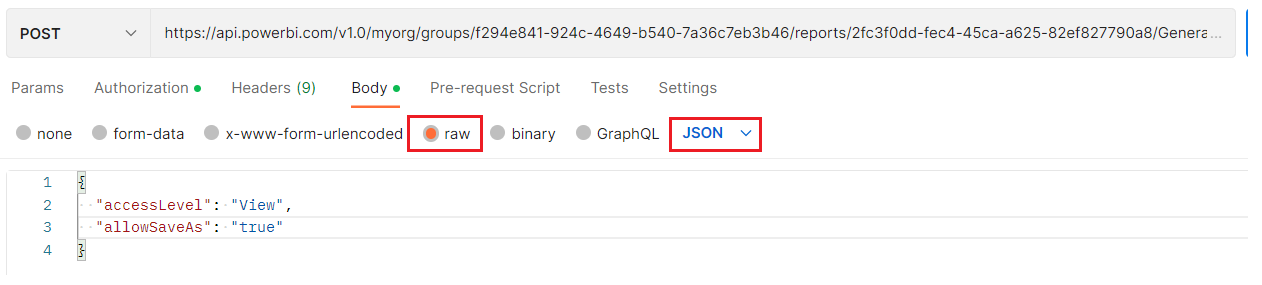
1. To start, make an HTTP Post call to the following URL:

<https://api.powerbi.com/v1.0/myorg/groups/groupId/reports/reportId/GenerateToken>

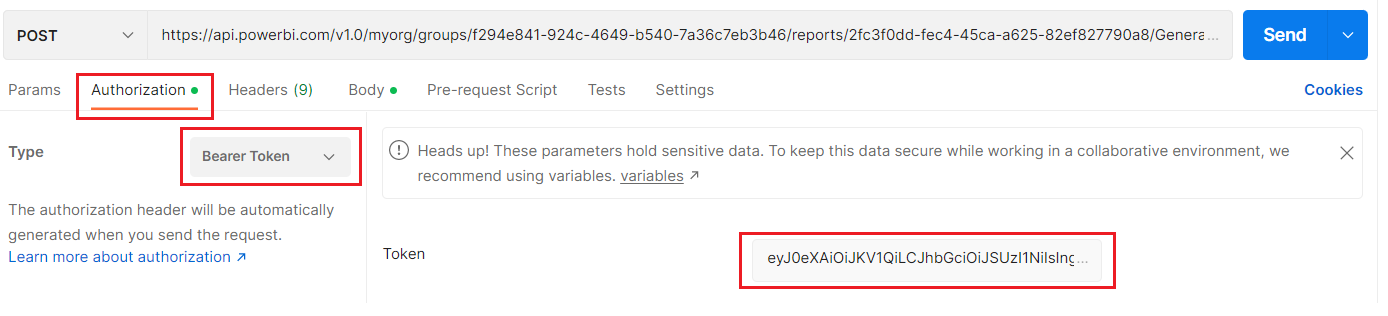
replace **groupId** and **reportId** from Power BI workspace report url (refer Step 8)

1. Input in the following parameters, using a JSON object in the body and Bearer Token (Access Token generated in Step 8) in the authorization.

**Sample Request:**



Use Access Token generated in Step 8 as Bearer Token in authorization.



**Sample Response (embedToken):**

Text

Description automatically generated

This embedding token is needed when loading Power BI dashboard in web application.

**Step 10: Embed Power BI Report in ReactJS**



Text

Description automatically generated

Replace reportId, embedUrl and embedToken and run the application.