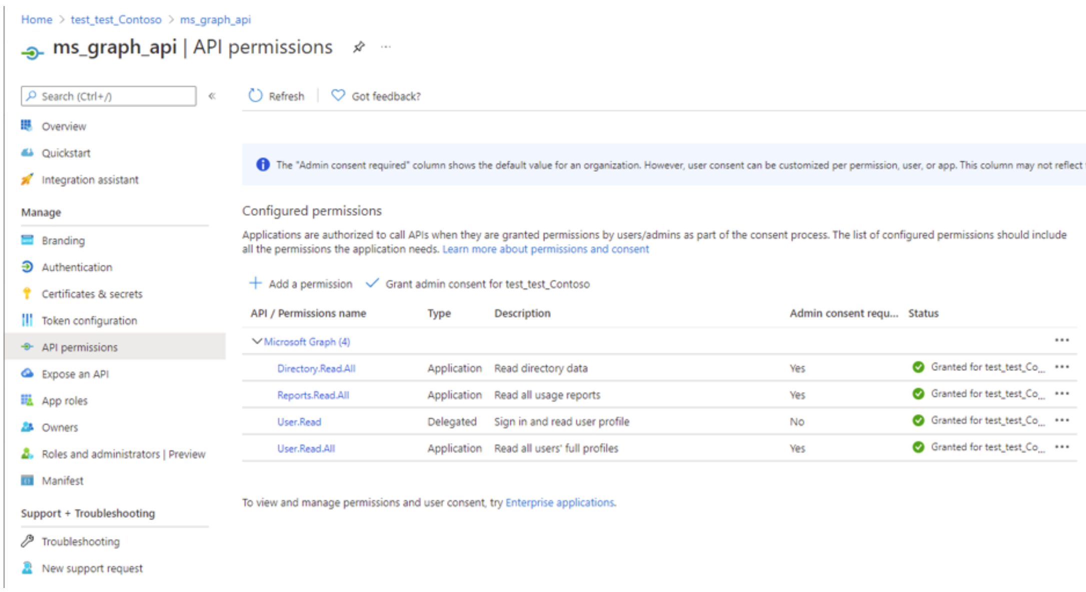
Microsoft Graph Reports API

Module Tutorial & Setup

How to use the Microsoft Graph Reports API module in your own Azure Synapse workspace

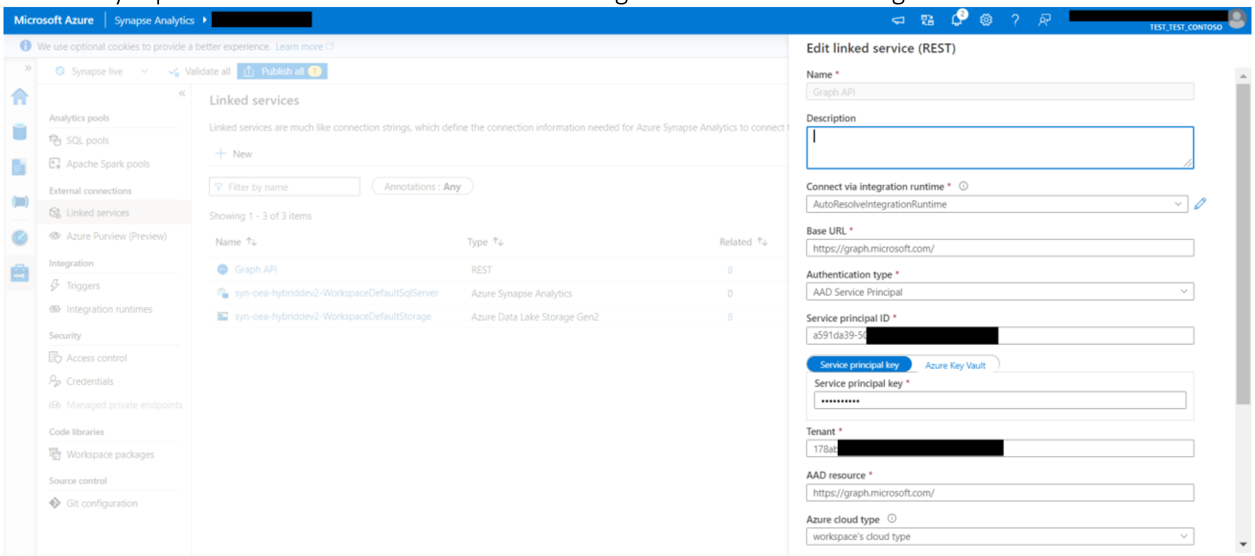
1. Create a linked service to Graph Reports API

- In your Azure, first go to AAD -> App registrations -> New registration and create a new app registration specifically for accessing the Graph Reports API from Synapse.
- Then click “Add Permission” and select Microsoft Graph -> select Application Permissions -> choose the “User.Read.All” permission, and click on the “Add permissions” button.
- Follow the same steps above, except now add the “Directory.Read.All” and “Reports.Read.All” permissions.
- After adding these permissions, you should see something similar to the following screenshot in your own Azure Portal:



  - Click on the “Grant admin consent” link once these permissions have been successfully added, so that you can verify all permissions show a status of “Granted” under the status column.
- Now go to “certificates & secrets” and add a new client secret -> copy that value.
- Now in your Synapse Studio, go to “linked services” under “Manage”, and add a REST service for Graph Reports API -> select AAD Service Principal as the Authentication type.
- Under the base URL and AAD resource fields, type in: https://graph.microsoft.com/
- Under “Service principal key”, paste the value of the secret you copied previously.

- Under “Service principal ID”, enter the “Application (client) ID” of the app registration created in the previous step (go to the Overview section of the app registration). After doing so, your Synapse environment should look something similar to the following screenshot:

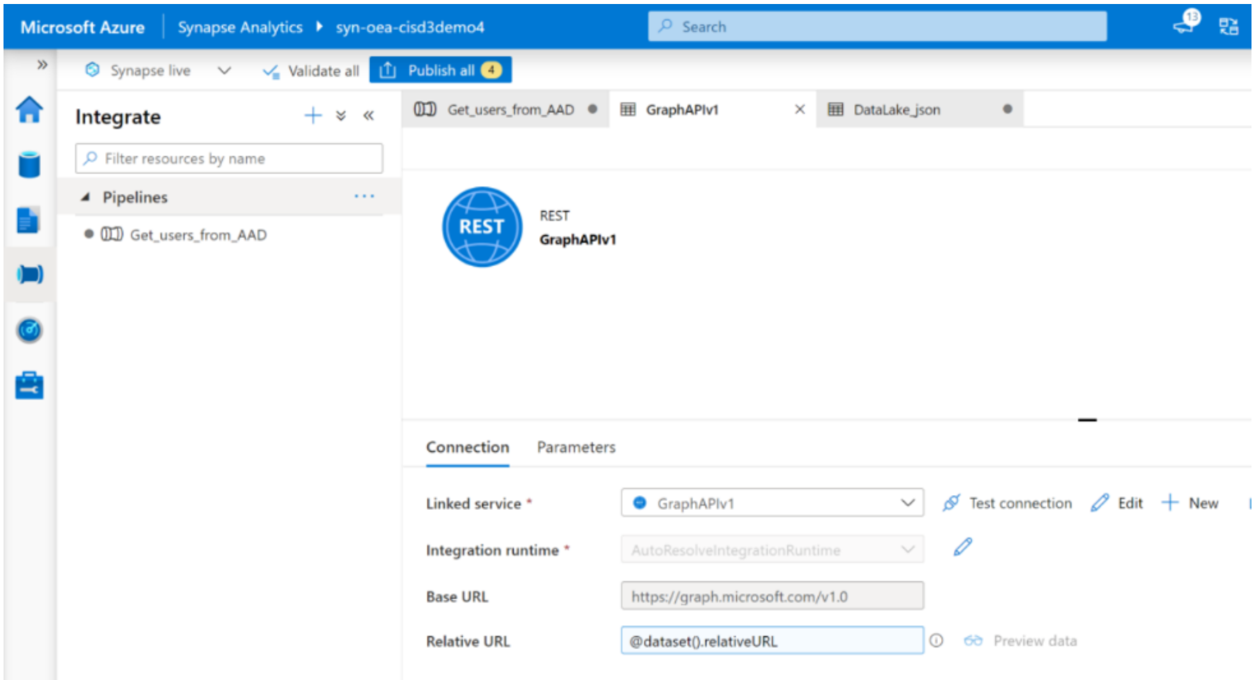


2. Notes

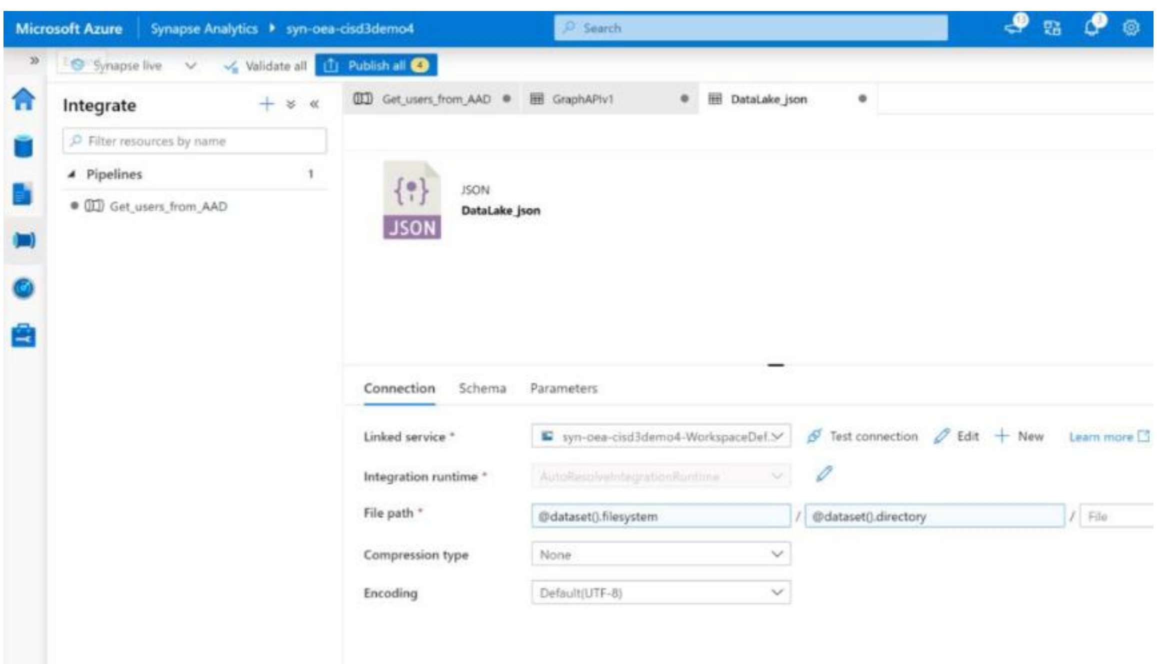
- Make sure that you’re Synapse environment is working within the OEA framework (i.e. the initial setup).
- Before moving forward in this tutorial, you can choose to either proceed with step 3 or step 4 (step 3 explains the process of creating a Graph Reports API pipeline from scratch; step 4 uses the sample datasets and pipeline template provided in this module). You should start with step 4 to familiarize yourself with the pipeline, and easily use the sample notebook.
    - If you want to use your own Graph Reports API data, you can start with step 3. To use the notebook provided, you will have to make sure that the pipeline is landing data in stage1np under a folder “GraphAPI” -> the JSON files being landed in this folder should be “users.json”, “m365_app_user_detail.json”, and “teams_activity_user_details.json” in order to run the notebook seamlessly (the queries used for these three files can be found in the GitHub dataset folder of this module).
    - As of September 1st, 2021, you will also have to sign in to **https://admin.microsoft.com/AdminPortal** to use your own data (due to the default hashing of userPrincipleNames); go to the admin center and login -> go to Settings -> Org Settings -> “Services” page. Select “Reports”. Uncheck the statement “Display concealed user, group, and site names in all reports,” and then save your changes. [Activity Reports in the Microsoft 365 admin center - Microsoft 365 admin | Microsoft Docs](https://docs.microsoft.com).

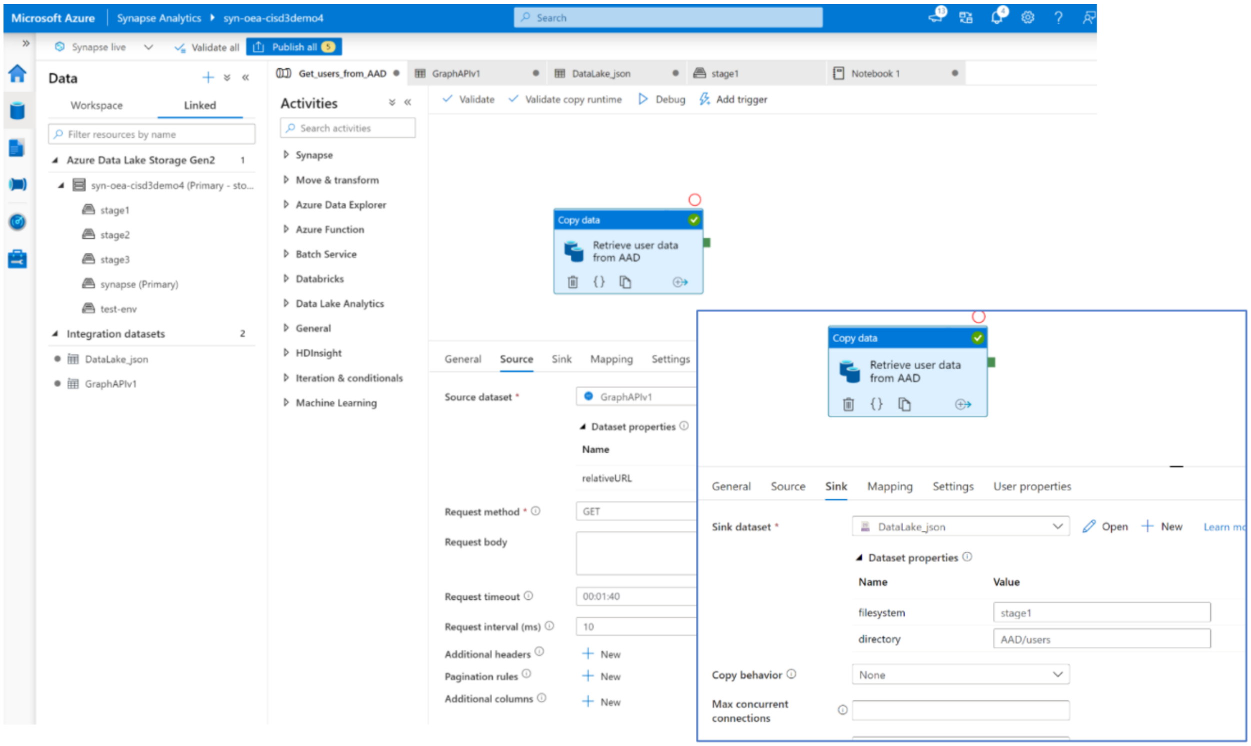
3. Creating the Graph Reports API pipeline from scratch within your Synapse environment

- Create a new pipeline under “Integrate” -> create a new REST dataset as the source, referring to the Graph Reports API linked service you created.



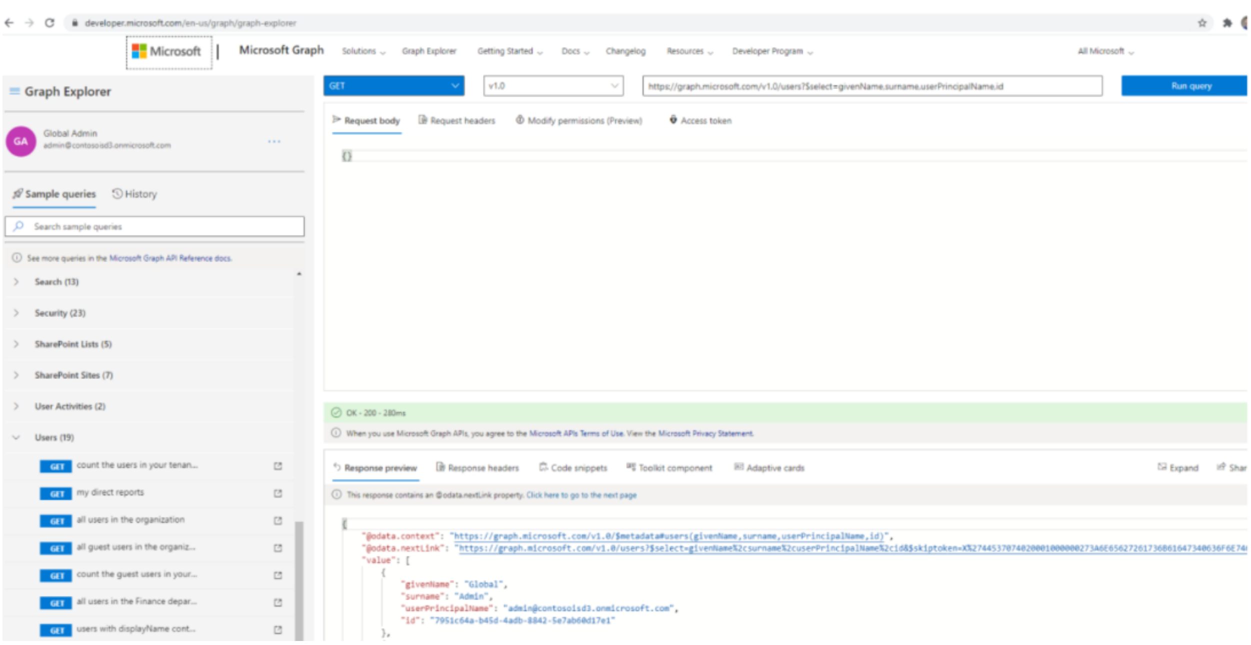
- Create a sink dataset going to the data lake as a JSON.





- You can reduce the data returned by selecting specific attributes in the relativeURL, like this: users?$select=givenName,surname,userPrincipalName,id

  Make sure you put this relativeURL in the “dynamic contents” section in order to retrieve the data from that query.
- Use the Graph Explorer to try it out, and see the data the query is pulling (found at https://developer.microsoft.com/en-us/graph/graph-explorer):



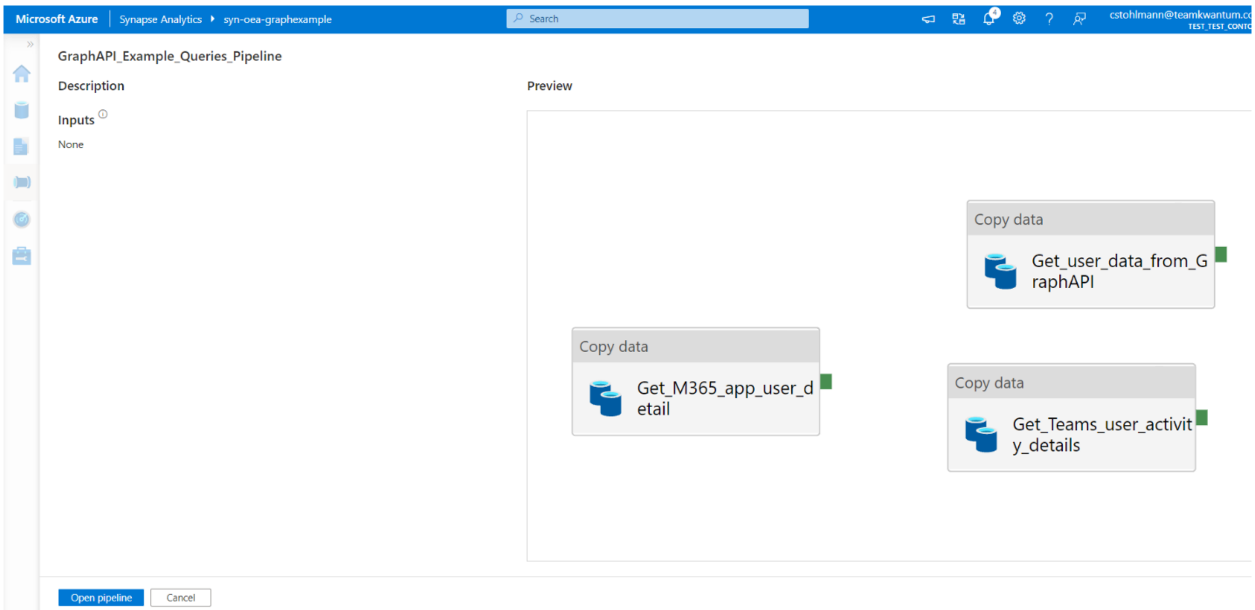
- If you run the pipeline and get a failure because of lack of access like this:

{ "errorCode": "2200", "message": "Failure happened on 'Source' side. ErrorCode=RestSourceCallFailed,'Type=Microsoft.DataTransfer.Common.Shared.HybridDeliveryException,Messa ge=The HttpStatusCode 403 indicates failure.\nRequest URL: https://graph.microsoft.com/v1.0/users?$top=3\nResponse payload:{\"error\":{\"code\":\"Authorization_RequestDenied\",\"message\":\"Insufficient privileges to complete the operation.\",\"innerError\":{\"date\":\"2021-05-28T19:31:00\",\"request-id\":\"abc221e5-a4d1-4845-8a5d-c26353b99af3\",\"client-request-id\":\"abc221e5-a4d1-4845-8a5d-c26353b99af3\"}}},Source=Microsoft.DataTransfer.ClientLibrary,'", "failureType": "UserError", "target": "Copy data1", "details": [] }

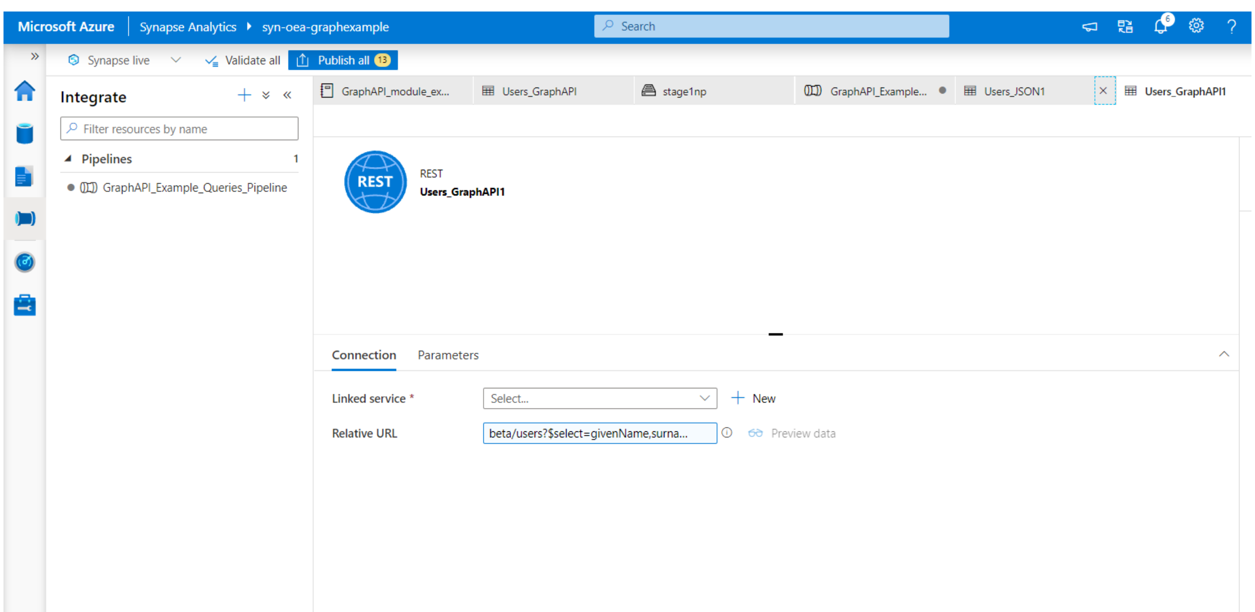
You'll need to double check the app registration you created and the API permissions to it.

4. Using the pipeline template in Synapse for the Graph Reports API

- First download the zip file under "pipelines" in the Graph Reports API module, and the sample JSON files under "datasets".
- Now go to your Synapse environment and navigate to "stage1np" under "data" and "linked" in your Gen2 data lake -> upload the "GraphAPI" datasets under this "stage1np" folder.
- Next, under "Integrate" and select "Add resource" -> then select "Import from pipeline template" -> select the pipeline zip you downloaded.



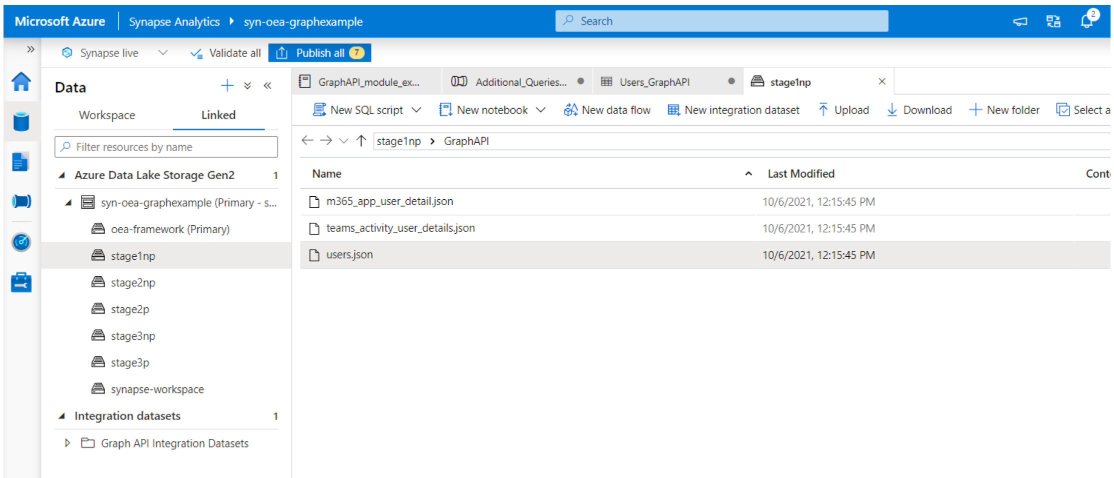
- Now you will just have to connect the REST data source to your linked service: for each of the "copy data" activities, go to "source" and connect your linked service pointing to the Graph Reports API.

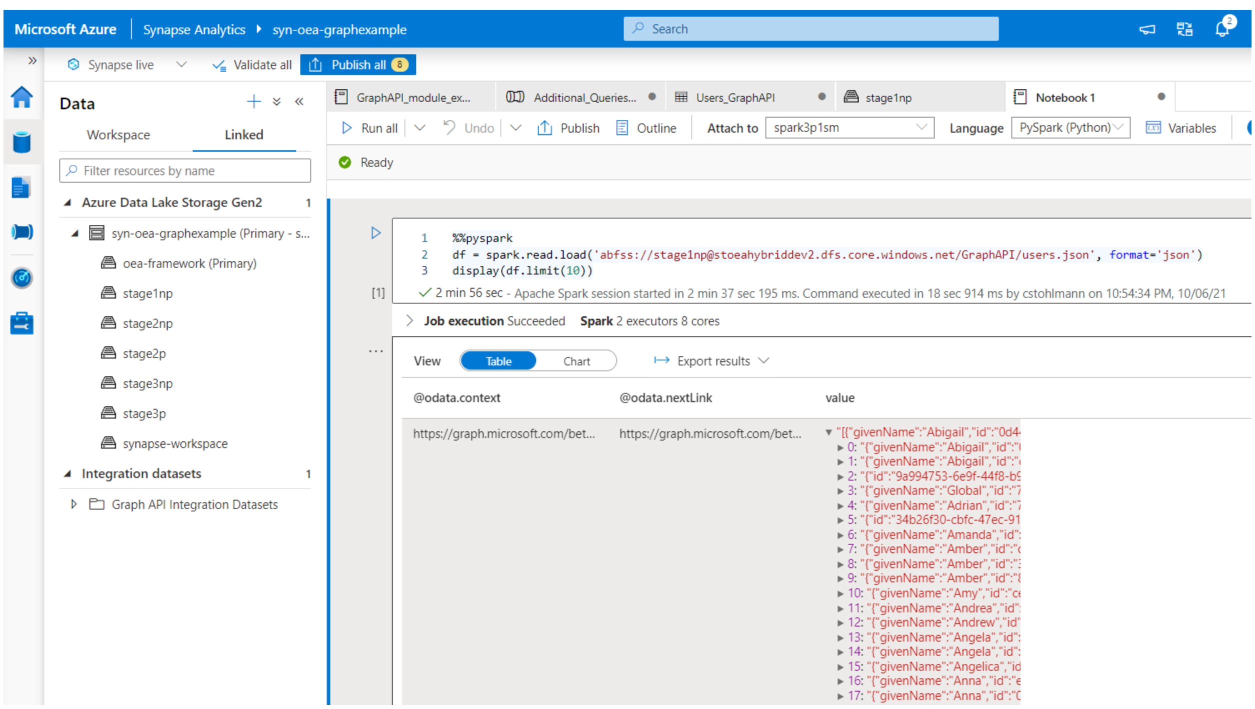


- After you've done this for each activity, you can manually trigger the pipeline to run.

5. View Data in Stage 1np

- You can then go into Synapse and navigate to your data lake and see the JSON file. Right click on it and select "Load to dataframe".

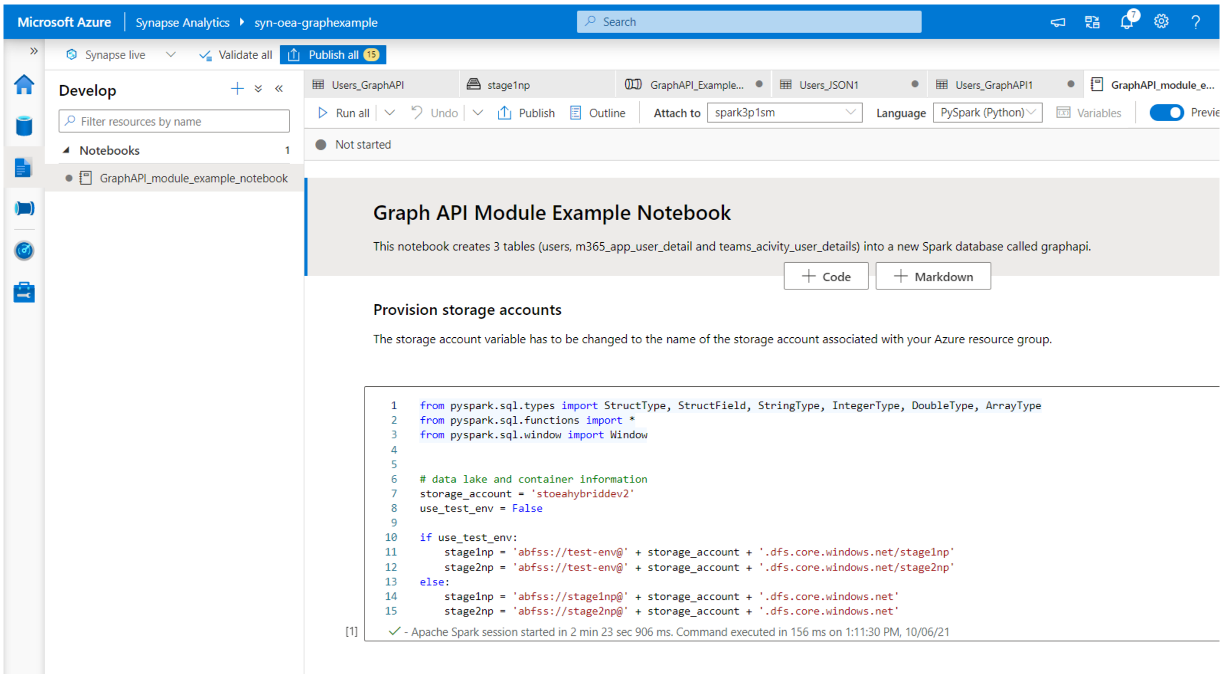




- Now you need to process this user data and write it to stage 2 in a more usable way, and create a spark database that can be easily queried from Power BI.

6. Execute the Notebook

- Download the notebook from this Graph Reports API module -> import the notebook under "Develop"
- Make sure you change the code in the code block "Provision storage accounts", so that you're using your own storage account name.
- Attach to your spark pool and now run this notebook:



- Running this notebook creates two spark databases: stage 2np and stage 2p. The stage 2p data is used within the scope of the PowerBI dashboard provided for this module.

7. View Data in Stage 2np and Stage 2p

- Now you're able to query the data from the newly created spark databases.