



# Graph API Pipeline Tutorial

*Setting up your own pipeline to pull user data from Graph API*

## 1. Create a linked service to Graph API

- In your Azure, first go to AAD -> App registrations -> New registration and create a new app registration specifically for accessing the Graph 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.
- Now go to “certificates & secrets” and add a new client secret -> copy that value.

The screenshot shows the 'API permissions' page in the Microsoft Azure portal. The application 'ar-syn-oea-cisdedemo4' is selected. The left sidebar shows navigation options like Overview, Quickstart, Integration assistant, Manage, Branding, Authentication, Certificates & secrets, Token configuration, API permissions (selected), Expose an API, App roles, Owners, and Roles and administrators. The main area shows 'Configured permissions' with a table listing permissions for Microsoft Graph. A note at the top states: 'The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value for your organization.'

API / Permissions name	Type	Description	Admin consent req...	Status
▼ Microsoft Graph (2)				
User.Read	Delegated	Sign in and read user profile	No	Granted for test_test_Co...
User.Read.All	Application	Read all users' full profiles	Yes	Granted for test_test_Co...

To view and manage permissions and user consent, try Enterprise applications.

- Now in your Synapse Studio, go to “linked services” under “Manage”, and add a REST service for Graph API -> select AAD Service Principal as the Authentication type.

The screenshot shows the 'Edit linked service (REST)' configuration page in the Microsoft Azure Synapse Studio. The left sidebar shows navigation options like Analytics pools, SQL pools, Apache Spark pools, External connections, Linked services (selected), Azure Purview (Preview), Integration, Triggers, Integration runtimes, Security, Access control, Credentials, Managed private endpoints, Code libraries, Workspace packages, Source control, and Git configuration. The main area shows the 'Linked services' list with columns for Name, Type, and Related. The 'Edit linked service (REST)' form is open, showing fields for Name, Description, Connect via integration runtime (set to AutoResolveIntegrationRuntime), Base URL (https://graph.microsoft.com/v1.0), Authentication type (AAD Service Principal), Service principal ID, Service principal key (linked to Azure Key Vault), Tenant, AAD resource (https://graph.microsoft.com/), Azure cloud type (workspace's cloud type), Server Certificate Validation (Enable), Auth headers, and Annotations. The 'Save' button is at the bottom right.

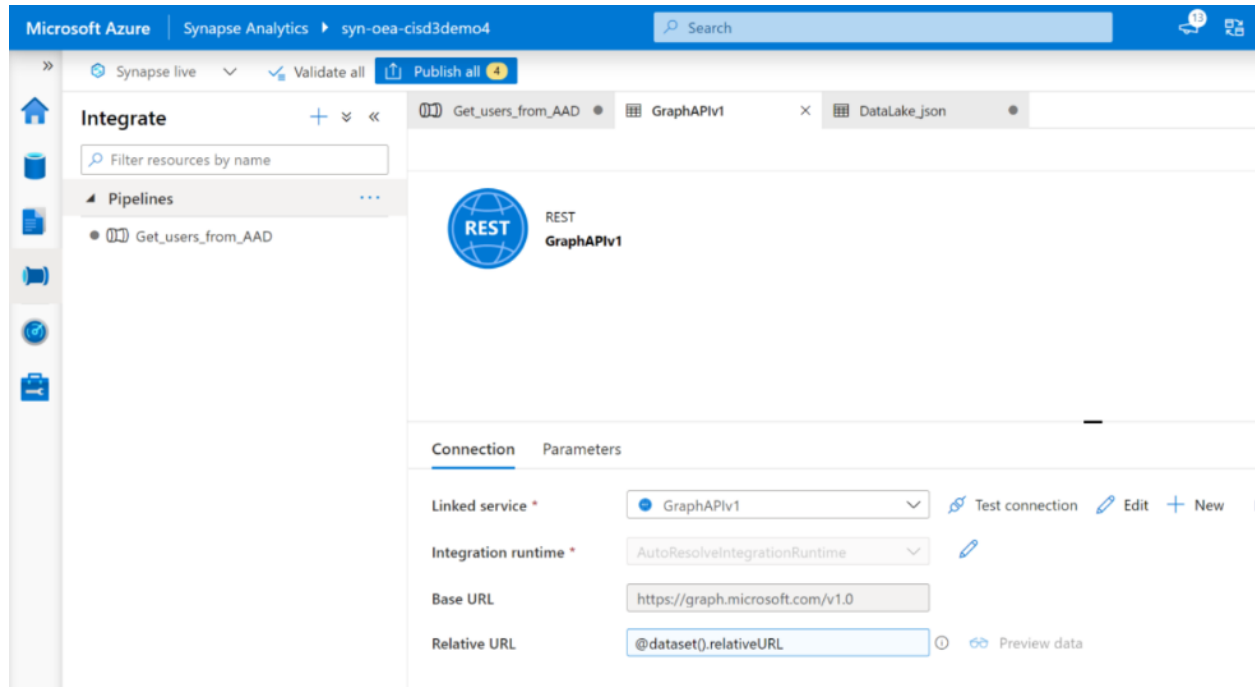
- Under the base URL and AAD resource fields, type in: <https://graph.microsoft.com/>
- Under “Service principal key”, paste the value of the secret you copied previously.
- Under “Service principal ID”, enter the “Application (client) ID” of the app registration created in the previous step (go to the Overview section of the app registration).

## 2. Notes

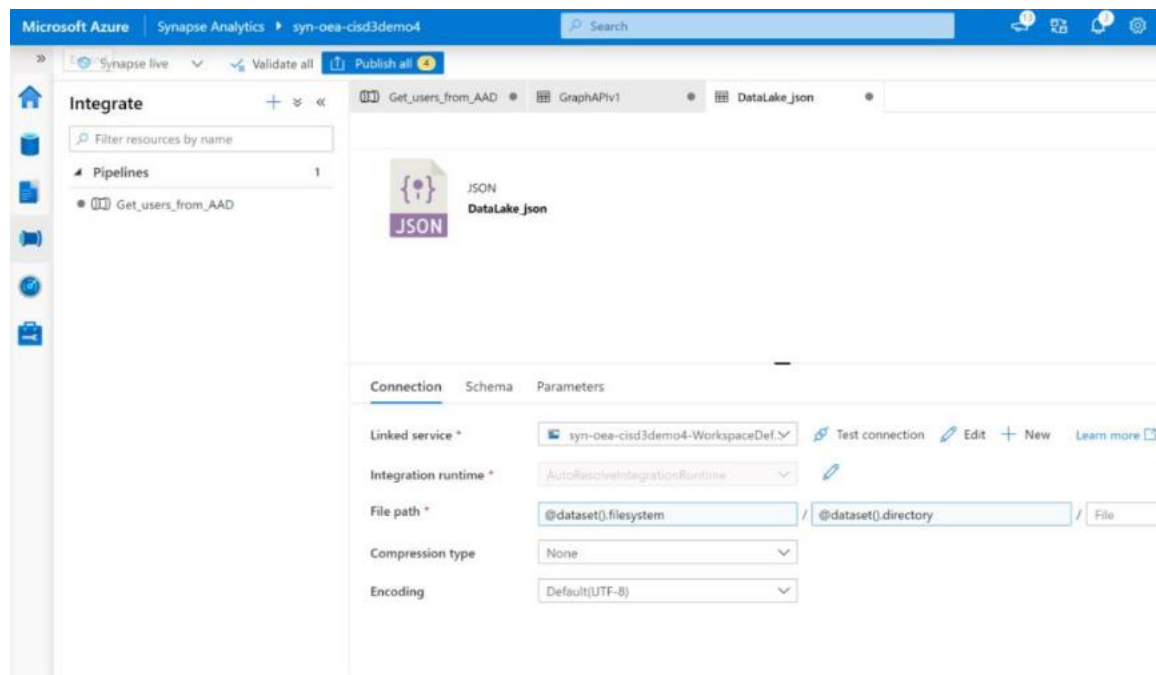
- Make sure that you’re Synapse environment is working within the OEA framework (i.e. the initial setup).
- Before moving forward in this tutorial, you can choose to either proceed with step 3 or step 4 (step 3 explains the process of creating a Graph pipeline from scratch; step 4 uses the sample datasets 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 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).
  - You will also have to sign in to <https://sds.microsoft.com> to use your own data (because of the default hashing of userPrincipalNames); 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.

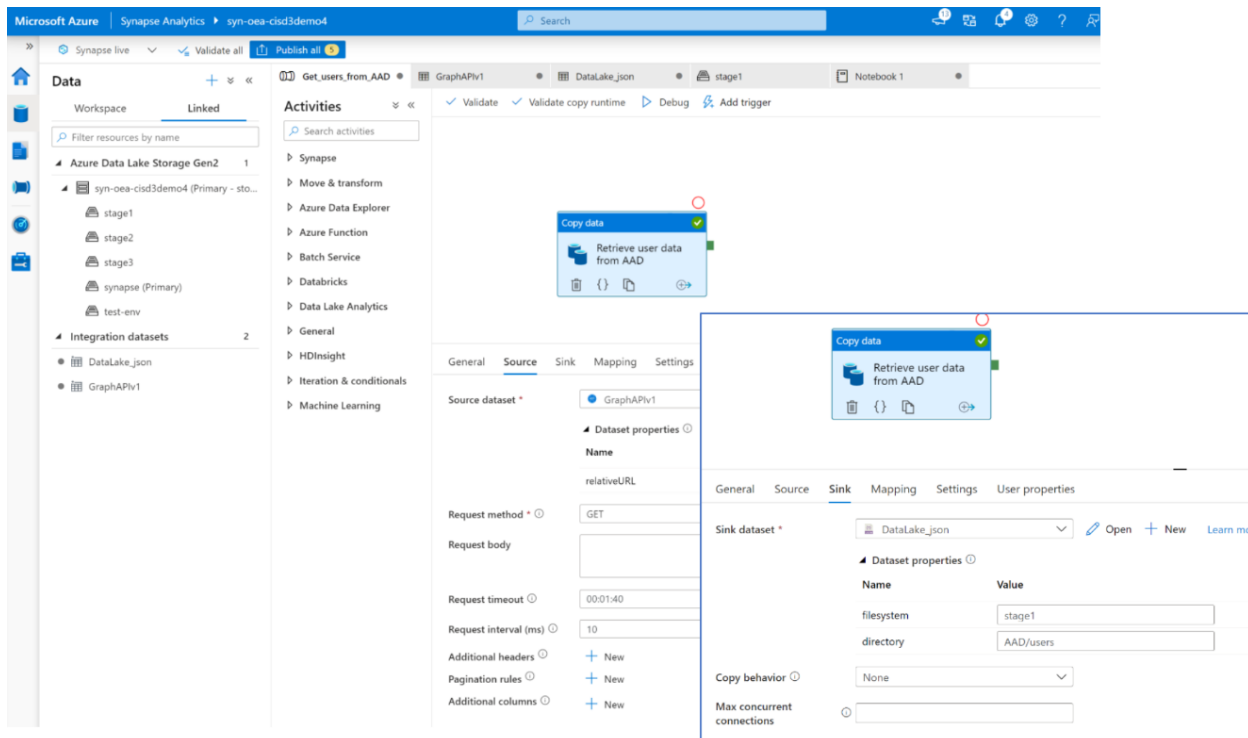
### 3. Creating the Graph pipeline from scratch within your Azure Synapse environment

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



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

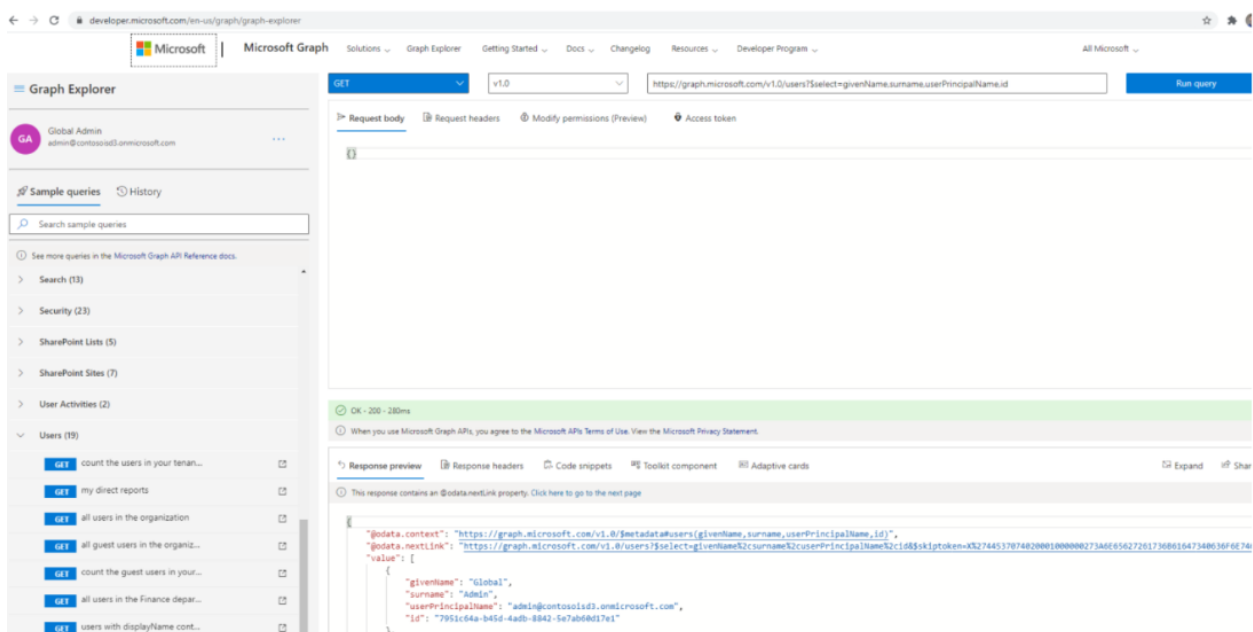




- You can reduce the data returned by selecting specific attributes in the relativeURL, like this: `users?$select=givenName,surname,userPrincipalName,id`

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>):



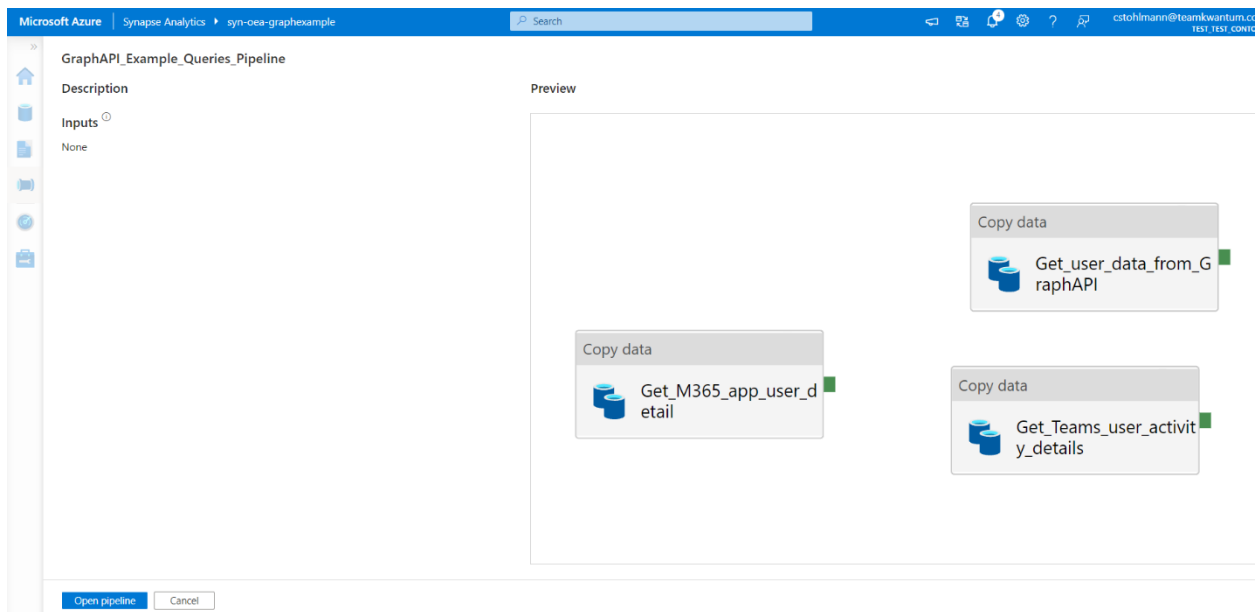
- If you run the pipeline and get a failure because of lack of access like this:

```
{ "errorCode": "2200", "message": "Failure happened on 'Source' side.
ErrorCode=RestSourceCallFailed,'Type=Microsoft.DataTransfer.Common.Shared.HybridDeliveryException,Message=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\": [ ] }
```

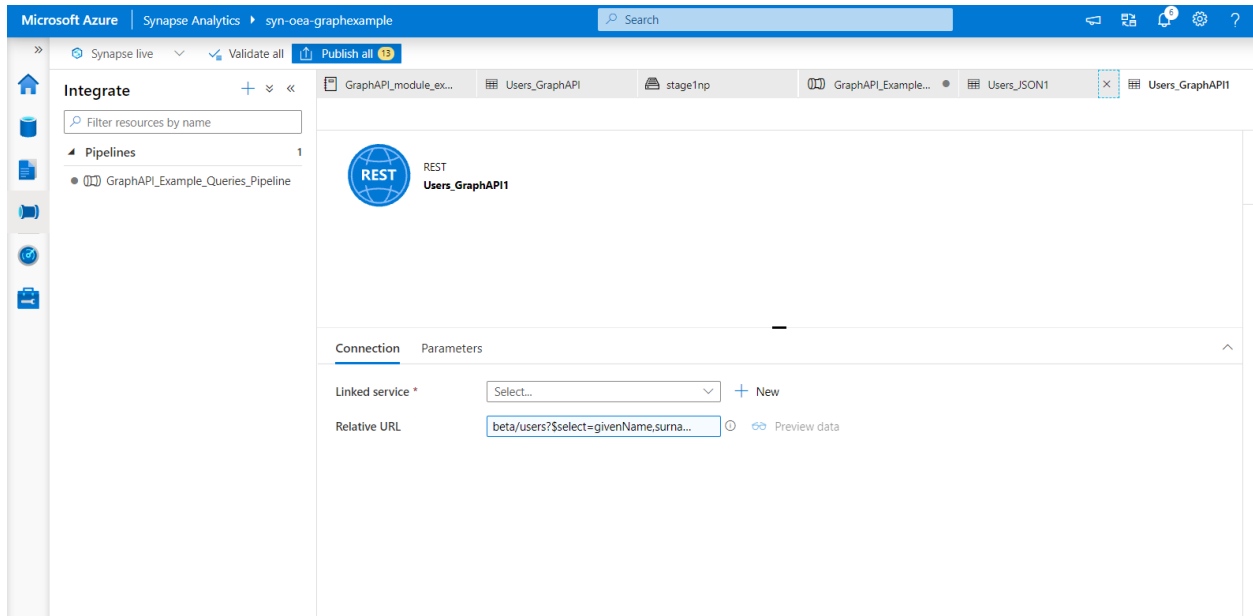
You'll need to double check the app registration you created and the API permissions to it.

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

- First download the zip file under “pipelines” in the Graph 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 -> create a new file named “GraphAPI” -> upload the datasets under this folder.
- Next, under “Integrate” and select “Add resource” -> then select “Import from pipeline template” -> select the pipeline zip you downloaded.



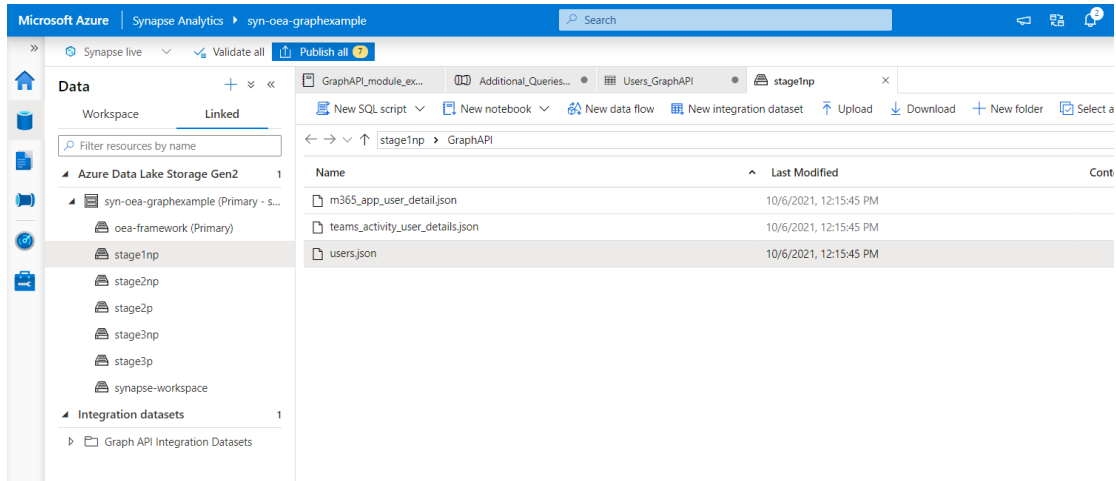
- Now you will just have to connect the REST data source to your linked service: for each of the “copy data” activities, go to “source” -> click “open” -> then select your linked service created in the step 1, pointing to the Graph API.



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

## 5. Working with the data

- 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”, then run the Synapse notebook.



Microsoft Azure | Synapse Analytics | syn-oea-graphexample

Workspace Linked

Filter resources by name

Azure Data Lake Storage Gen2 1

- syn-oea-graphexample (Primary - s...
- oea-framework (Primary)
- stage1np
- stage2np
- stage2p
- stage3np
- stage3p
- synapse-workspace

Integration datasets 1

- Graph API Integration Datasets

GraphAPI\_module\_ex... Additional\_Queries... Users\_GraphAPI stage1np Notebook 1

Run all Undo Publish Outline Attach to spark3p1sm Language PySpark (Python) Variables

Ready

```

1 %pyspark
2 df = spark.read.load('abfss://stage1np@stoeahybriddev2.dfs.core.windows.net/GraphAPI/users.json', format='json')
3 display(df.limit(10))

```

[1] ✓ 2 min 56 sec - Apache Spark session started in 2 min 37 sec 195 ms. Command executed in 18 sec 914 ms by cstohlmann on 10:54:34 PM, 10/06/21

Job execution Succeeded Spark 2 executors 8 cores

View Table Chart Export results

@odata.context	@odata.nextLink	value
https://graph.microsoft.com/bet...	https://graph.microsoft.com/bet...	<ul style="list-style-type: none"> <li>0: {"givenName":"Abigail","id":"0d444980-3123-46f4-85d3-6f49e4619ea5"}</li> <li>1: {"givenName":"Abigail","id":"c19234e7-a816-41af-9524-4be4092fc559"}</li> <li>2: {"id":"9a994753-6e9f-44f8-b5...","givenName":"Global","id":"7951c64a-b45d-4adb-8842-5e7ab60d17e1"}</li> <li>3: {"givenName":"Global","id":"7951c64a-b45d-4adb-8842-5e7ab60d17e1"}</li> <li>4: {"givenName":"Adrian","id":"7a2ca37f-d1f3-4be8-8abe-8711d4eea8d6"}</li> <li>5: {"id":"34b26f30-cbfc-47ec-9131-27fef4433705","givenName":"Adrian","id":"7a2ca37f-d1f3-4be8-8abe-8711d4eea8d6"}</li> <li>6: {"givenName":"Amanda","id":"154f2b2c-9e73-4ecc-917e-06c0e440ea15"}</li> <li>7: {"givenName":"Amber","id":"dbc10b91-8d2c-4cf2-ac08-985ad3b4d6b4"}</li> <li>8: {"givenName":"Amber","id":"3f901837-788d-407b-b6e5-3d09018011fe"}</li> <li>9: {"givenName":"Amber","id":"3f901837-788d-407b-b6e5-3d09018011fe"}</li> <li>10: {"givenName":"Amy","id":"...","givenName":"Amy","id":"..."}</li> <li>11: {"givenName":"Andrea","id":"...","givenName":"Andrea","id":"..."}</li> <li>12: {"givenName":"Andrew","id":"...","givenName":"Andrew","id":"..."}</li> <li>13: {"givenName":"Angela","id":"...","givenName":"Angela","id":"..."}</li> <li>14: {"givenName":"Angela","id":"...","givenName":"Angela","id":"..."}</li> <li>15: {"givenName":"Angelica","id":"...","givenName":"Angelica","id":"..."}</li> <li>16: {"givenName":"Anna","id":"...","givenName":"Anna","id":"..."}</li> <li>17: {"givenName":"Anna","id":"...","givenName":"Anna","id":"..."}</li> </ul>

- 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.
- Run the notebook provided in this Graph module to do this.
- Now you're able to query the user data from the spark database.

Microsoft Azure | Synapse Analytics | syn-oea-graphexample

Workspace Linked

Filter resources by name

Databases 2

- default (Spark)
- graphapi (Spark)
  - Tables
    - m365\_app\_user\_detail
    - teams\_activity\_user\_details
    - users
      - Columns
      - Properties

GraphAPI\_module\_ex... Additional\_Queries... Users\_GraphAPI stage1np SQL script 1

Run Undo Publish Query plan Connect to Built-in Use database graphapi

```

1 SELECT TOP (100) [surname]
2 ,[givenName]
3 ,[userPrincipalName]
4 ,[id]
5 FROM [graphapi].[dbo].[users]

```

Results Messages

View Table Chart Export results

surname	givenName	userPrincipalName	id
Long	Abigail	abigaillong3@contosoisd3.onmi...	0d444980-3123-46f4-85d3-6f49e4619ea5
Smith	Abigail	abigailsmith35@contosoisd3.on...	c19234e7-a816-41af-9524-4be4092fc559
Admin	Global	admin@contosoisd3.onmicrosoft...	7951c64a-b45d-4adb-8842-5e7ab60d17e1
Jordan	Adrian	adrianjordan15@contosoisd3.on...	7a2ca37f-d1f3-4be8-8abe-8711d4eea8d6
(NULL)	(NULL)	agundapaneni_microsoft.com#E...	34b26f30-cbfc-47ec-9131-27fef4433705
Long	Amanda	amandalong47@contosoisd3.on...	154f2b2c-9e73-4ecc-917e-06c0e440ea15
Berger	Amber	amberberger79@contosoisd3.on...	dbc10b91-8d2c-4cf2-ac08-985ad3b4d6b4
Buchanan	Amber	amberbuchanan5@contosoisd3...	3f901837-788d-407b-b6e5-3d09018011fe