**Configure REST API Data Flow from SharePoint to AZ SQL Database**

|  |
| --- |
| **Summary:**  The ADF Data Flow loads data from a SharePoint list (via REST API), flattens it into rows, and compares it against existing rows in an Azure SQL table.   * If a SharePoint item doesn’t exist in SQL (new ID) → **insert it**. * If it does exist in SQL → update it. * It’s essentially a delta load (upsert) pipeline from SharePoint into Azure SQL. |

**Note:** To retrieve the siteId from Graph Explorer perform the following:

* Login with your account to Graph Explorer:
  + [Graph Explorer | Try Microsoft Graph APIs - Microsoft Graph](https://developer.microsoft.com/en-us/graph/graph-explorer)
* Add the following url to perform GET:
  + [https://graph.microsoft.com/v1.0/sites/<tenantName>.sharepoint.com:/sites/<siteName>/](https://graph.microsoft.com/v1.0/sites/%3ctenantName%3e.sharepoint.com:/sites/%3csiteName%3e/)
* Copy the id key value.

**1.1 Configure the REST API Linked Service**

**Note:** The M365 App registration must be configured with the following API permissions:

* Graph API > Application > Sites.ReadAll and Sites.Selected

1. On the Azure Data Factory Studio left pane, click on Manage.
2. Click on Linked Services and then + New.
3. Select REST and click continue.
4. Add the following properties:
   1. Connect via integration runtime: Managed VnetIR
   2. BaseUrl: [https://graph.microsoft.com/v1.0/sites/<tenantName>.sharepoint.com,<siteId](https://graph.microsoft.com/v1.0/sites/%3ctenantName%3e.sharepoint.com,%3csiteId)>
   3. Authentication type: OAuth 2.0
   4. Token Endpoint: [https://login.microsoftonline.com/<M365tenantId>/oauth2/v2.0/token](https://login.microsoftonline.com/%3cM365tenantId%3e/oauth2/v2.0/token)
   5. ClientId: M365 app reg clientId
   6. Client secret: M365 app reg secret value
   7. Scope: <https://graph.microsoft.com/.default>
5. Click Test connection.
6. Once successful connection completes, click Save.

**1.2 Configure the** **AZ SQL Database Linked Services**

1. On the Azure Data Factory Studio left pane, click on Manage.
2. Click on Linked Services and then + New.
3. Select AZ SQL Database and click continue.
4. Enter the following:
5. Integration runtime: managed vnet
6. Subscription: <subscription name>
7. Server name: <Az SQLServer name>
8. Database name:<Az SQL database name>
9. Authentication type: System-assigned managed identity
10. Click on test connection.
11. Once successful connection, click on Create.

**1.3 Configure the AZ SQL Dataset**

1. On the Azure Data Factory Studio left pane, click on Author.
2. Click on Datasets and right click on three dots.
3. Select New dataset.
4. Select Azure SQL Database and click continue.
5. Dataset name: < table name of AZ SQL database>\_delta
6. Linked Servies: select the AZ SQL linked service created.
7. Linked services: Ensure the new AZ SQL Linked service has been selected.

Note: ensure interactive authoring is enabled.

1. Table name: <the target table name>
2. Click Ok.

**1.4 Configure Data Flow from SharePoint data source to Az SQL Database**

1. On the Azure Data Factory Studio left pane, click on Author.
2. Click on Data flows.
3. On the three dots to the right, click New data flow.
4. Click Add source.
5. Add the following settings:
   1. Name: <Rest source name>
   2. Source type: Inline
   3. Inline dataset type: REST
   4. Checked allow schema drift.

Under source options:

* 1. Request method: Get
  2. Relative URL: lists/<listId>/?expand=fields

Under projection:

* 1. Click on Import Schema.

1. Click Add source under REST source activity.
2. Add the following settings:
   1. Name: <SQL source name>
   2. Source type: Dataset
   3. dataset type: <azure sql dataset name>
   4. Checked allow schema drift.

Under source options:

* 1. Input: query
  2. Query: SELECT <PK column> FROM <tablename>

Under Projection:

* 1. Click Import Projection.

1. Click on the + and add a Flatten activity.
2. Add the following settings:
   1. Name: <Flatten name>
   2. Incoming stream: < Rest source name>
   3. Unroll by: body.value
   4. Unroll root: body.value
   5. Click + add mapping
   6. Source column: select body.value.fields
   7. Name as: body.value.fields
3. Click on the + and add a Select activity.
4. Add the following settings:
   1. Name: <Select name>
   2. Incoming stream: < Flatten activity name>
   3. Uncheck auto mapping.
   4. Click + Add mapping.
   5. Select each column to map with the Sink dataset.
5. Click on the + and add a Join activity.
6. Add the following settings:
   1. Name: <Join name>
   2. Incoming stream: < Select activity name>
   3. Left stream: <select activity name>
   4. Right stream: <sql source activity name>
   5. Join type: Left outer
   6. Join conditions: <select activity PK column> == <sql source activity PK column>
7. Click on the + and add a Alter row activity.
8. Add the following settings:
9. Name: <Alter row name>
10. Incoming stream: < Join activity name>
11. Alter row conditions:
    1. Select Insert if and add expression ‘isNull(<sqlSourcename>@<PK column name>)’
    2. Select Update if and add expression ‘!isNull(<sqlSourcename>@<PK column name>)’
12. Click on the + and add a Sink activity.

Add the following settings:

* 1. Name: < Sink name>
  2. Incoming stream: < Alter row activity name>
  3. Sink type: Dataset
  4. Dataset: <azure sql dataset name>
  5. Click allow schema drift.

Under settings:

* 1. Check allow insert.
  2. Check allow upsert.
  3. Check allow update.
  4. Key column: select <PK column name>

Under mapping:

* 1. Check skip duplicate input columns.
  2. Check skip duplicate output columns.
  3. Uncheck auto mapping.
  4. Click on + Add mapping.
  5. Select each input column name and map to output column name

The overall data flow will look like the diagram displayed below:

A screenshot of a computer

AI-generated content may be incorrect.

**1.5 Configure the Pipeline**

1. On the ADF left pane, click Author.
2. Click Pipeline and click three dots on new pipeline.
3. Under General tab, add the pipeline name.
4. Under Settings, add the following:
   1. Dataflow: < the newly created dataflow name>
   2. Run on (AzureIR): <select Azure IR or ManagedVnetIR>
5. Click on Publish at the top to save.
6. To Test the pipeline, ensure Data flow debug is enabled and Click ‘Debug’.

**1.6 References**

* Graph Explorer: [Graph Explorer | Try Microsoft Graph APIs - Microsoft Graph](https://developer.microsoft.com/en-us/graph/graph-explorer)
* [List items - Microsoft Graph v1.0 | Microsoft Learn](https://learn.microsoft.com/en-us/graph/api/listitem-list?view=graph-rest-1.0&tabs=http)