

Azure Data Factory

Cargar un archivo JSON Object a un Dedicated SQL Pool

Se busca procesar el siguiente archivo **JSON Object**. Básicamente es el mismo archivo que el punto 14, pero se le agrego el elemento “details” que vendría a ser del tipo ‘**Object**’. En Spark vendría a ser del tipo ‘**Struct**’.

```
1  [  
2  {  
3    "customerid":1,  
4    "customername":"UserA",  
5    "registered":true,  
6    "courses":["AZ-900", "AZ-500", "AZ-303"],  
7    "details" :  
8      {  
9        "mobile":"111-1112",  
10       "city":"CityA"  
11      }  
12  },  
13  {  
14    "customerid":2,  
15    "customername":"UserB",  
16    "registered":true,  
17    "courses":["AZ-104", "AZ-500", "DP-200"],  
18    "details" :  
19      {  
20        "mobile":"333-1112",  
21        "city":"CityB"  
22      }  
23  }]
```

Comenzamos creando nuestro Dataflow

The screenshot shows the Microsoft Azure Data Factory interface. The top navigation bar includes the Microsoft Azure logo, the workspace name 'appfactory5000', a search bar, and user information 'techsup1000@gmail.com'. The main workspace is titled 'customerdataflow'. On the left, there's a sidebar with icons for home, recent items, and a search bar. The central canvas is empty, with a dashed box labeled 'Add Source' and a button 'Expand resources pane'. The right sidebar shows the 'Properties' panel with tabs for 'General' and 'Related'. The 'General' tab is active, showing the 'Name' field with the value 'customerdataflow' and a 'Description' field.

The screenshot shows the Microsoft Azure Data Factory interface with the 'customerdataflow' Dataflow selected. The central canvas displays a source named 'source1' with 'Columns: 0 total'. Below the canvas, the 'Source settings' tab is active, showing the 'Output stream name' field with the value 'CustomerStream'. The 'Source type' is set to 'Dataset'. The 'Dataset' dropdown menu is open, showing a 'Select...' option and a '+ New' button. The 'Options' section shows 'Allow schema drift' checked.

Cargaremos el archivo desde Azure Data Lake Storage y será un archivo JSON:

The screenshot shows the 'Set properties' dialog in the Microsoft Azure Data Factory portal. The dialog is for a new stream named 'CustomerCourseDt'. The 'Linked service' is set to 'AzureDataLakeStorage'. The 'File path' is set to 'data / raw/customer / customer.json'. The 'Import schema' section has 'From connection/store' selected. The 'Advanced' section is collapsed. The 'OK' button is highlighted with a mouse cursor.

Microsoft Azure | appfactory5000

Search

techsup1000@gmail.com
DEFAULT DIRECTORY

Data Factory | Validate all | Publish all 1

AccessLogpipeline | AccessLogDataFlow | AccessLogDataLateDt | customerdataflow

Validate | Data flow debug

CustomerStream
Columns: 0 total

Add Source

Source settings | Source options | Projection | Optimize | Inspect | Data preview

Output stream name * | CustomerStream

Source type * | Dataset | Inline

Dataset * | Select...

Options | ☒ Allow schema drift ⓘ

Set properties

Name | CustomerCourseDt

Linked service * | AzureDataLakeStorage

File path | data / raw/customer / customer.json

Import schema | ☒ From connection/store | ☐ From sample file | ☐ None

Advanced

OK | Back | Cancel

The screenshot shows the 'Flatten settings' dialog in the Microsoft Azure Data Factory portal. The dialog is for a stream named 'FlattenStream'. The 'Incoming stream' is set to 'CustomerStream'. The 'Unroll by' and 'Unroll root' are both set to 'courses'. The 'Options' section has 'Skip duplicate input columns' and 'Skip duplicate output columns' unchecked. The 'Input columns' section shows '4 mappings: All inputs mapped'.

Microsoft Azure | appfactory5000

Search

techsup1000@gmail.com
DEFAULT DIRECTORY

Data Factory | Validate all | Publish all 2

AccessLogpipeline | AccessLogDataFlow | AccessLogDataLateDt | customerdataflow

Validate | Data flow debug

CustomerStream
Import data from CustomerCourseDt

FlattenStream
Columns: 4 total

Flatten settings | Optimize | Inspect | Data preview

Output stream name * | FlattenStream

Incoming stream * | CustomerStream

Unroll by * ⓘ | courses

Unroll root ⓘ | courses

Options | ☐ Skip duplicate input columns ⓘ | ☐ Skip duplicate output columns ⓘ

Input columns * | Reset | + Add mapping | Delete | 4 mappings: All inputs mapped

CustomerStream's column | Name as

Microsoft Azure | appfactory5000

Search

techsup1000@gmail.com
DEFAULT DIRECTORY

Data Factory > Validate all > Publish all 2

pipeline > AccessLogDataFlow > AccessLogDataLateDt > customerdataflow > CustomerPipeline > CustomerCourseDt

✓ Validate Data flow debug

CustomerStream FlattenStream CustomerSynapse

Flatten settings Optimize Inspect Data preview Description

Options

- ☐ Skip duplicate input columns ⓘ
- ☐ Skip duplicate output columns ⓘ

Input columns *

Reset + Add mapping Delete 6 mappings: All inputs mapped

<input type="checkbox"/> CustomerStream's column	Name as
<input type="checkbox"/> courses	courses
<input type="checkbox"/> abc customerid	customerid
<input type="checkbox"/> abc customername	customername
<input checked="" type="checkbox"/> registered	registered
<input type="checkbox"/> abc details.mobile	mobile
<input type="checkbox"/> abc details.city	city

Microsoft Azure | appfactory5000

Search

techsup1000@gmail.com
DEFAULT DIRECTORY

Data Factory > Validate all > Publish all 2

Accesslogpipeline > AccessLogDataFlow > AccessLogDataLateDt > customerdataflow

✓ Validate Data flow debug

CustomerStream FlattenStream sink1

Import data from CustomerCourseDt

Unrolling arrays from courses to courses with columns 'customerid, customername, registered, courses'

Columns: 4 total

Sink Settings Mapping Optimize Inspect Data preview Description

Output stream name * CustomerSynapse Learn more ⓘ

Incoming stream * FlattenStream

Sink type *

Dataset Inline Cache

Dataset * Select... + New

Options

- ☒ Allow schema drift ⓘ
- ☐ Validate schema ⓘ

El dataset de destino apunta al Azure Synapse

The screenshot shows the 'Set properties' dialog for a sink named 'CustomerSynapseDt'. The 'Linked service' is set to 'AzureSynapseAnalytics'. The 'Table name' is 'dbo.customercourse'. The 'Import schema' is set to 'From connection/store'. The 'Sink type' is 'Dataset'. The 'Output stream name' is 'CustomerSynapse' and the 'Incoming stream' is 'FlattenStream'. The 'Dataset' is set to 'Select...'. The 'Options' section has 'Allow schema drift' checked and 'Validate schema' unchecked. The 'OK' button is highlighted.

Set properties

Name: CustomerSynapseDt

Linked service *: AzureSynapseAnalytics

☒ Select from existing table ☐ Create new table

Table name: dbo.customercourse

☐ Edit

Import schema: ☒ From connection/store ☐ None

Advanced

Output stream name *: CustomerSynapse

Incoming stream *: FlattenStream

Sink type *: Dataset

Dataset *: Select...

Options: ☒ Allow schema drift ☐ Validate schema

OK Back Cancel

Revisamos el mapeo de las columnas en la caja de destino

The screenshot shows the 'Mapping' tab for a sink named 'CustomerSynapseDt'. A warning message states: 'At least one incoming column is mapped to a column in the sink dataset schema with a conflicting type, which can cause NULL values or runtime errors.' The 'Options' section has 'Skip duplicate input columns' and 'Skip duplicate output columns' checked. The 'Auto mapping' toggle is off. The 'Output format' button is highlighted. The '6 mappings: All outputs mapped' status is shown. The mapping table is highlighted with a red box.

At least one incoming column is mapped to a column in the sink dataset schema with a conflicting type, which can cause NULL values or runtime errors.

Options: ☒ Skip duplicate input columns ☒ Skip duplicate output columns

Auto mapping ☐ Reset Add mapping Delete Output format

6 mappings: All outputs mapped

Input columns	Output columns
abc customerid	123 customerid
abc customername	abc customername
% registered	% registered
abc courses	abc courses
abc mobile	abc mobile
abc city	abc city

Si revisamos los tipos de datos que toma el archivo JSON Object desde la fuente

The screenshot shows the 'Schema' tab for a JSON source named 'CustomerCourseDt'. A table lists the column names and their corresponding data types. The table is highlighted with a red border.

Column name	Type
customerid	integer
customername	string
registered	boolean
courses	string[]
details	object
mobile	string
city	string

Y desde el paso fuente indicamos que corresponde a un archivo JSON Array

The screenshot shows the 'Source options' tab for a 'CustomerStream' source. The 'JSON settings' section is expanded, and the 'Array of documents' option is selected, highlighted with a red border.

Source settings | **Source options** | Projection | Optimize | Inspect | Data preview

name

After completion * ☒ No action ☐ Delete source files ☐ Move

Filter by last modified Start time (UTC) End time (UTC)

JSON settings

Document form ☐ Single document ☐ Document per line ☒ Array of documents

Unquoted column names ☐

Has comments ☐

Single quoted ☐

Backslash escaped ☐

Para ejecutar nuestro Data Flow creamos un nuevo Pipeline

The screenshot shows the Microsoft Azure Data Factory interface. The top navigation bar includes the Microsoft Azure logo, the workspace name 'appfactory5000', a search bar, and user information 'techsup1000@gmail.com'. The left sidebar contains a navigation menu with options like 'Data Factory', 'Accesslogpipeline', 'AccessLogDataFlow', 'AccessLogDataLateDt', 'customerdataflow', and 'CustomerPipeline'. The main area displays the 'Activities' list on the left, including 'Move & transform', 'Copy data', and 'Data flow'. A red arrow points to the 'Data flow' activity in the list. In the center, a 'CustomerDataFlow' activity card is shown with a red underline under its name. The right sidebar shows the 'Properties' panel for the 'CustomerPipeline', with the 'Name' field set to 'CustomerPipeline' and a 'Description' field.

This screenshot shows the 'Settings' tab for the 'CustomerDataFlow' activity. The 'Data flow' dropdown is set to 'customerdataflow'. The 'Run on (Azure IR)' is set to 'AutoResolveIntegrationRuntime'. The 'Compute type' is set to 'General purpose'. The 'Core count' is set to '4 (+ 4 Driver cores)'. The 'Logging level' is set to 'Verbose'. The 'Sink properties' section is partially visible at the bottom. The right sidebar shows the 'Properties' panel for the 'CustomerPipeline', with the 'Name' field set to 'CustomerPipeline' and a 'Description' field.

Microsoft Azure | appfactory5000

Search

techsup1000@gmail.com
DEFAULT DIRECTORY

Data Factory > Validate all > Publish all 1

AccessLogpipeline > AccessLogDataFlow > AccessLogDataLateDt > customerdataflow > CustomerPipeline

Activities

- Move & transform
 - Copy data
- Data flow
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

CustomerDataFlow

General Settings Parameters User properties

Run on (Azure IR) * AutoResolveIntegrationRuntime

Compute type * General purpose

Core count * 4 (+ 4 Driver cores)

Logging level * ☒ Verbose ☐ Basic ☐ None

Sink properties

Staging ☒ Staging

Staging linked service ☒ AzureDataLakeStorage Test connection Edit + New

Staging storage folder synapse / Directory Browse

Validamos, Publicamos y Ejecutamos

Microsoft Azure | appfactory5000

Search

techsup1000@gmail.com
DEFAULT DIRECTORY

Data Factory > Validate all > Publish all

AccessLogpipeline > AccessLogDataFlow > AccessLogDataLateDt > customerdataflow

Activities

- Move & transform
 - Copy data
- Data flow
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

CustomerDataFlow

General Settings Parameters User properties

Run on (Azure IR) * AutoResolveIntegrationRuntime

Compute type * General purpose

Core count * 4 (+ 4 Driver cores)

Logging level * ☒ Verbose ☐ Basic ☐ None

Sink properties

Staging ☒ Staging

Staging linked service ☒ AzureDataLakeStorage Test connection Edit + New

Staging storage folder synapse / Directory Browse

Publishing completed
Successfully published

Trigger now
Trigger on-demand run of the last published pipeline

New/Edit

Esta es la estructura de la tabla a la cual serán cargados los datos

```
1 drop table customercourse
2
3 CREATE TABLE [customercourse]
4 (
5 [customerid] int,
6 [customername] varchar(200),
7 [registered] BIT,
8 [courses] varchar(200),
9 [mobile] varchar(200),
10 [city] varchar(200)
11 )
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL script:

```
drop table customercourse

CREATE TABLE [customercourse]
(
[customerid] int,
[customername] varchar(200),
[registered] BIT,
[courses] varchar(200),
[mobile] varchar(200),
[city] varchar(200)
)

SELECT * FROM [customercourse]
```

The query was executed successfully, and the results are displayed in the Results pane. The data is as follows:

	customerid	customername	registered	courses	mobile	city
1	1	UserA	1	AZ-900	111-1112	CityA
2	1	UserA	1	AZ-500	111-1112	CityA
3	1	UserA	1	AZ-303	111-1112	CityA
4	2	UserB	1	AZ-104	333-1112	CityB
5	2	UserB	1	AZ-500	333-1112	CityB
6	2	UserB	1	DP-200	333-1112	CityB

The status bar at the bottom indicates: Query executed successfully. appworkspace9000.sql.azures... sqladminuser (117) newpool 00:00:00 6 rows