Getting started with data tasks using Python in Azure Synapse Analytics

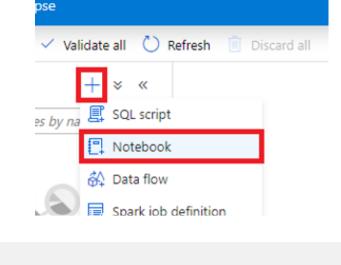
Azure Synapse is a limitless analytics service that brings together enterprise data warehousing and big data analytics. It gives you the freedom to query data on your terms, using either serverless or provisioned resources—at scale. Azure Synapse provides a deep integration between Spark and SQL, enabling you to use any combination of Spark and SQL for your ETL, data exploration, prep, and engineering scenarios.

Tip: Get started with Azure Synapse Analytics in four quick steps.

Azure Synapse From the Azure Synapse home page, select the

Create a Notebook to run PySpark in

Develop hub from the Azure Synapse Studio. Click the **plus sign (+)** and select **Notebook**.

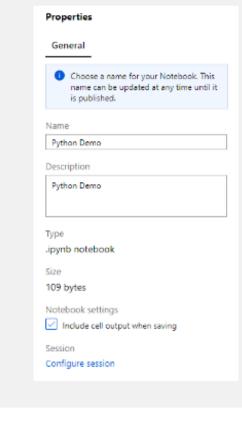


The Notebook supports multiple languages such as PySpark (Python), Scala, .NET Spark

Select PySpark (Python)

(C#), and Spark SQL. For this exercise, select PySpark (Python). In the **Properties** pane, fill out the Notebook name and the (optional) description. The

Notebook name can be up to 140 characters (only letters, numbers, '-', and '_' are allowed). Spaces are only permitted in the middle.



Add text and code cells to your Notebook

Azure Synapse Analytics Python Demo ## Data source: Public Holidays Open Dataset

A text cell can be written using Markdown language. It helps to describe the code in your Notebook.

Simply click + Cell and then Add text cell. Enter the below text in the text cell.

```
Add some Python code in a new code cell by clicking + Cell and Add code cell. Run the code below.
               from azureml.opendatasets import PublicHolidays
```

from datetime import datetime from dateutil import parser

```
from dateutil.relativedelta import relativedelta
Enter code to load the Public Holidays data from the Microsoft Azure Open Dataset. Limit the data to
the past 12 months by running the code below.
               end date = datetime.today()
               start date = end date - relativedelta(months=12)
```

holidays = PublicHolidays(start_date=start_date, end date=end date)

```
Next, convert the source data to a Spark DataFrame. Run the code below.
               holidays_df = holidays.to_spark_dataframe()
```

1 holidays_df.count()

Get a count of this DataFrame to see the total number of rows.

holidays df.count()

[12]

Job execution Succeeded Spark 2 executors 16 cores

display(holidays_df.limit(20))

holidayName

Human Rights Day

Command executed in 4s 434ms by AzureMentor

春分の日

Job execution Succeeded Spark 2 executors 16 cores

countryOrRegion

Japan

South Africa

Argentina

Human Rights Day|

Argentina Día Nacional de l... Día Nacional de l...

|countryOrRegion|

South Africa

```
Job execution Succeeded Spark 2 executors 16 cores
                        545
Use the show() method to output the first 20 rows of this DataFrame to sample the data.
                 holidays df.limit(20).show()
When you execute this code, the first 20 rows of the dataset are displayed.
            holidays_df.limit(20).show()
```

holidayName|normalizeHolidayName|isPaidTimeOff|countryRegionCode|

null

null

JP|2019-03-21 00:00:00|

ZA 2019-03-21 00:00:00

AR 2019-03-24 00:00:00

View in monitoring

2019-03-21T00:00:00.000Z

2019-03-21T00:00:00.000Z

2019-03-24T00:00:00.000Z

1 Publish

Using the display() method, the DataFrame can be output in tabular format. display(holidays df.limit(20)) Execute the code to see the result below.

normalizeHolidayName

Día Nacional de la Memoria por la V...

Human Rights Day

春分の日

countryRegionCode

Human Rights Day|

One advantage of showing the results with the display() method is that you can instantly render the output as a variety of charts, such as line, bar, area, scatter, and pie charts.

display(holidays_df.limit(20))

Día Nacional de la Memoria por la V...

Job execution Succeeded Spark 2 executors 16 cores



Python Demo

> Run all

1. Publish

Saving the Notebook There are three ways to save a copy of your Notebook.

2. Publish all Similar to the **Publish** command, the **Publish all** command enables you to save all notebooks and scripts in your Azure Synapse workspace with one click.

The **Publish** command enables you to save an

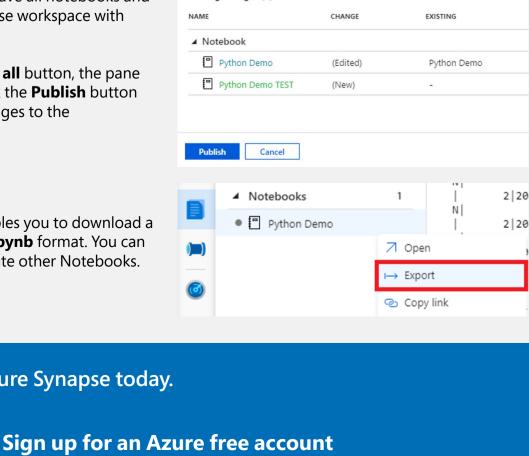
individual Notebook in your Azure Synapse workspace in the cloud. This enables you to go

live environment. 3. Export The Export command enables you to download a

copy of the Notebook in .ipynb format. You can

then import this file to create other Notebooks.

back to your Notebook anytime, anywhere. Publish all You are about to publish all pending changes to the live environment. Learn more 🖸 Pending changes (2) ▲ Notebook Python Demo Once you click the **Publish all** button, the pane Python Demo TEST at right will be shown. Click the **Publish** button to publish all pending changes to the



"> Undo

Get started with Azure Synapse today.





e-book from Packt Speak to a sales specialist for help with pricing,

Get more details in a free technical

other internet website references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any

intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.