Microsoft Azure

Getting started with data tasks using Python in Azure Synapse Analytics

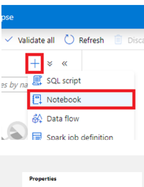
Azure Synapse is a limitless analytics service that brings together enterprise data warehousing and big data analytics. It gives you the freedom to query data on your terms, using either serverless or provisioned resources—at scale. Azure Synapse provides a deep integration between Spark and SQL, enabling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

Create a Notebook to run PySpark in Azure Synapse

From the Azure Synapse home page, select the **Develop** hub from the Azure Synapse Studio.

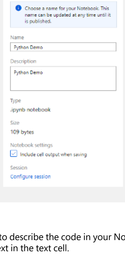
Click the **plus sign (+)** and select **Notebook**.



Select PySpark (Python)

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

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.

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

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.

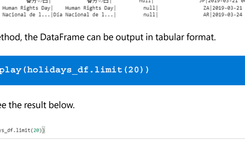
```
end_date = datetime.today()
start_date = end_date - relativedelta(months=12)
holidays = PublicHolidays(start_date=start_date,
end_date=end_date)
```

Next, convert the source data to a Spark DataFrame. Run the code below.

```
holidays_df = holidays.to_spark_dataframe()
```

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

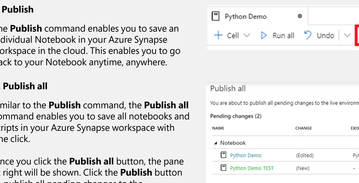
```
holidays_df.count()
```



Use the show() method to output the first 20 rows of this DataFrame to sample the data.

```
holidays_df.limit(20).show()
```

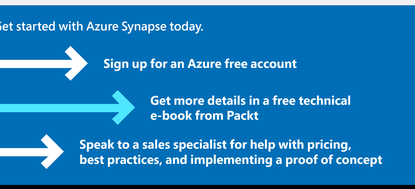
When you execute this code, the first 20 rows of the dataset are displayed.



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.



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.

Saving the Notebook

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

1. Publish

The **Publish** command enables you to save an individual Notebook in your Azure Synapse workspace in the cloud. This enables you to go back to your Notebook anytime, anywhere.

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.

Once you click the **Publish all** button, the pane at right will be shown. Click the **Publish** button to publish all pending changes to the 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.

Get started with Azure Synapse today.

Sign up for an Azure free account

Get more details in a free technical e-book from Packt

Speak to a sales specialist for help with pricing, best practices, and implementing a proof of concept