

TIM, Python & Power BI

Instructions for setting up the demo template of TIM in Power BI through Python.

1. Contents

[**Goal** 3](#_Toc94192496)

[**Overview** 3](#_Toc94192497)

[**Prerequisites** 4](#_Toc94192498)

[**Setup** 5](#_Toc94192499)

[**Demo Flow** 11](#_Toc94192500)

[**Jupyter Notebooks** 12](#_Toc94192501)

[**Remarks** 12](#_Toc94192502)

1. Instructions

# **Goal**

The goal of this demo template is to show you how to set up an automated process with TIM using Python as the processing tool and Power BI for visualization.

# **Overview**

This template is based on the following files within the zip-folder accompanied with this document:

* **Python Scripts**
  + Input\_Data\_Engineering.py
  + TIM\_Forecasting.py
  + Output\_Data\_Engineering.py
* **Jupyter notebooks**
  + Input\_Data\_Engineering.ipynb
  + TIM\_Forecasting.ipynb
  + Output\_Data\_Engineering.ipynb
  + Testing\_Notebook.ipynb
* **Power BI**
  + Template.pbix
* **JSON file**
  + credentials.json
  + TIM\_job\_configuration.json
* **Batch file**
  + Run\_Template.bat
* **Input csv files** in the folder **Input\_Data**
  + publicHolidays.csv
  + target.csv
  + weather.csv
* **Output csv files** in the folder **Output\_Data & subfolders**
* **Result csv files** in the folder **Results**
  + accuracy\_df
  + evolution\_df
  + features\_df
  + forecast\_job\_results\_table\_df
  + job\_logs\_df
  + performance\_df
  + predictions\_df
  + properties\_df
  + tim\_input\_df

# **Prerequisites**

* **Power BI Desktop**

You can download Power BI Desktop from the Microsoft website.

* **Python 3.6 or higher**

You can download the latest version of Python from the Python website.

* **TIM Python Client**

You can download the Python Client for TIM here [tim-client · PyPI](https://pypi.org/project/tim-client/). For more information please visit our documentation here: [Overview - TIM Documentation (tangent.works)](https://docs.tangent.works/TIM-on-Platforms/Python-Client/Overview/).

* **TIM Credentials**

If you don’t have credentials for TIM already, contact [info@tangent.works](mailto:info@tangent.works) for more information.

This setup is designed for Windows but can be replicated for other operating systems. Contact [support@tangent.works](mailto:support@tangent.works) for any questions regarding this demo setup.

# **Setup**

Follow these steps to set up the example template.

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Description | Actions | Example |
| 1 | All files required for this setup can be found in the zip-file. Create a separate folder to store these files and later also all results generated by this process. | Extract the contents of the zip-file to a chosen location. |  |
| 2 | These credentials will give you access to the TIM Engine. The python scripts will fetch this information and connect to the API. | Insert TIM Credentials in the JSON file |  |
| 3 | Browse through the Power BI template where you will find example pages for visualize TIM results. | Open the Power BI template |  |
| 4 | We will need to update the data sources to refer to the results folder in your directory. | Go to “Transform Data” |  |
| 5 | Change the pathname so that it refers to the example files in the results folder.  C:\Users\... \Results\ | Change the path name |  |
| 6 | Close & apply these changes. The screen will revert back to the dashboard and the tables will be updated. Make sure all tables are correctly refreshed. | Click on “Close & Apply” |  |
| 7 | Run the bat-file to test the flow. A command line screen will appear and start executing the python scripts in the folder. Once finished new csv- files will become available in the “Results” & the “Output\_Data” folder. Make sure to verify this. | Run the bat-file |  |
| 8 | Click on refresh to collect the latest csv-files from the Results folder. | Refresh the Power BI Dashboard |  |
| 9 | The dashboard is now ready for automation. Open the task schedule to set up an automatic execution process. | Open the task scheduler |  |
| 10 | Make sure to create a basic task to be able to execute a program. | Create a basic task |  |
| 11 | Give the task a specific name. | Name the basic task |  |
| 12 | Depending on the use case you can choose when the process is executed.  This template runs daily. | Set the recurrence |  |
| 13 | You can choose a specific timing. This is important to keep in mind when thinking about data availability and updates of data sources.  This template runs at 8 AM. | Set the timing |  |
| 14 | Set the task to execute a program which will be the bat-file in our folder. | Set the action to “Start a program” |  |
| 15 | Navigate to the bat-file and select it so that the task will execute it for us automatically at the set timings. | Browse to the bat-file. |  |
| 16 | You can finish up the task or open properties if you wish to specify certain parameters of the task. | Finish up the basic task. |  |
| 17 | The automatic setup is now completed. You can verify if the task is ready to run at the set timing. Also you could test the flow by manually clicking on “Run”. | Click on Run |  |
| 18 | You can delete the contents from the Output\_Data folder (subfolders and their content) to start a new clean dashboard.  Optionally you can delete the contents from the Results folder as well, however these files will be overwritten by the latest experiment anyway. | Clean up the folders |  |

# **Demo Flow**

The diagram below shows the data flows, files and scripts used.

Diagram

Description automatically generated

1. **The task scheduler executes the bat-file**
2. **The bat-file executes 3 python script in this order:**
   1. **Input\_Data\_Engineering.py**

This script transforms data from various sources and generates a table that is suitable for uploading to TIM.

* 1. **TIM\_Forecasting.py**

This script collects credentials for accessing TIM, an input data table and configuration settings for TIM. Then it connects to TIM and creates output-files.

* 1. **Output\_Data\_Engineering.py**

This script transforms the TIM output into more useful information

1. **The scripts store data in the Output & Results folders**
   1. **Output folders**

This folder collects the results from each individual execution. These separate files will be used to showcase results from previous executions.

* 1. **Result Folder**

This folder collects results from the latest execution as well the results from the Output\_Data\_Engineering.py script. These files are overwritten at every execution to contain the latest results.

1. **The files from the results folder are used in Power BI to visualize the results**

# **Jupyter Notebooks**

If you wish to test out the python scripts, we have include jupyter notebooks to assist you in exploring the code without breaking the process. For each of the 3 scripts a notebook is present. Also 1 testing notebook with a standard TIM template is available for general testing.

* Input\_Data\_Engineering.ipynb
* TIM\_Forecasting.ipynb
* Output\_Data\_Engineering.ipynb
* Testing\_Notebook.ipynb

# **Remarks**

This example template shows a possible way to implement TIM in an automated process. This process can be generalized and replicated with different processing & visualization tools. For more information contact [support@tangent.works](mailto:support@tangent.works).