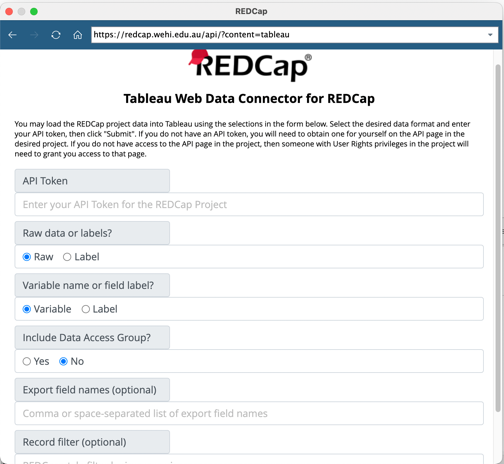
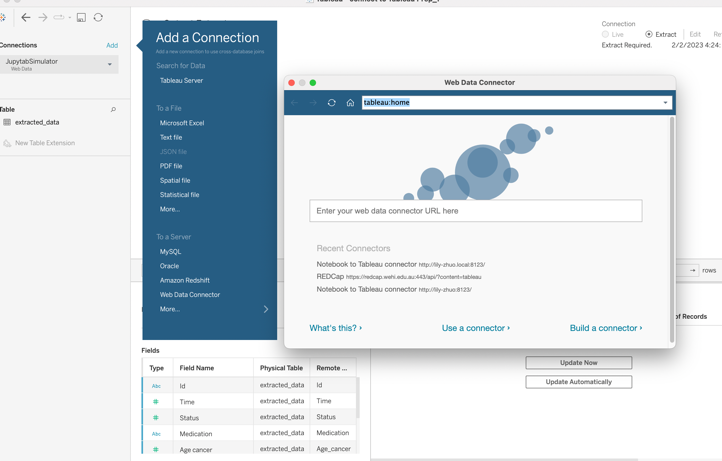
This documentation shows how we use jupyter notebook as an intermediate to connect the redcap database to extract data and used the preprocessed data frame as data source in Tableau Desktop.

Note, Tableau Desktop can directly connect to the data in the data frame through web data connector after allowing the utilization of API in Redcap Tableau external modules setting. But when I tried to connect to the redcap database, it took over 12 hours but still failed to load data. In our case, it may be due to the large amount of data in the database, which is over 540000 rows, while it can successfully connect to a small data frame. Another reason we need an intermediate is that it is hard to use only Tableau Desktop to preprocess the data. In the previous preprocess step, I used Tableau Prep to do the clean and preprocess step, However, there is no web data connector function developed in Tableau Prep until the date(2 Feb 2023)



Here we tried an intermediate: Jupyter notebook and use the package of 

1. Get the exported data from the redcap database
2. Clean and preprocess data in the format used to draw the Kaplan Meier curve
3. The final preprocessed data frame is sent to the web server (local server)
4. Tableau accesses data through the web data connector in the local server

Reference:<https://github.com/CFMTech/Jupytab>

Follow the instruction to set the environment to use jupytab

Two main components:

* Jupytab: API expose dataframe and function from notebook

Command in terminal:

conda create -n jupytab-notebook-env python=3.7

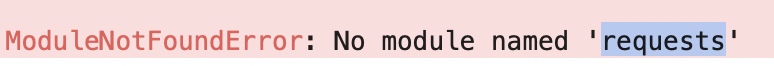
conda activate jupytab-notebook-env

conda install jupytab=0.9.11 -> pip install jupytab if when using conda can not install package ,we can change to use pip to install python package or go to <https://anaconda.org> to find related package in the conda)

conda install ipykernel

python -m ipykernel install --user --name jupytab-simulation-demo

If there is ModuleNotFoundError shown below in the jupyter note book just install module in the environment



-> conda install requests

* Jupytab-server: launch a server that connects to tableau along with a configuration file

Command in terminal:

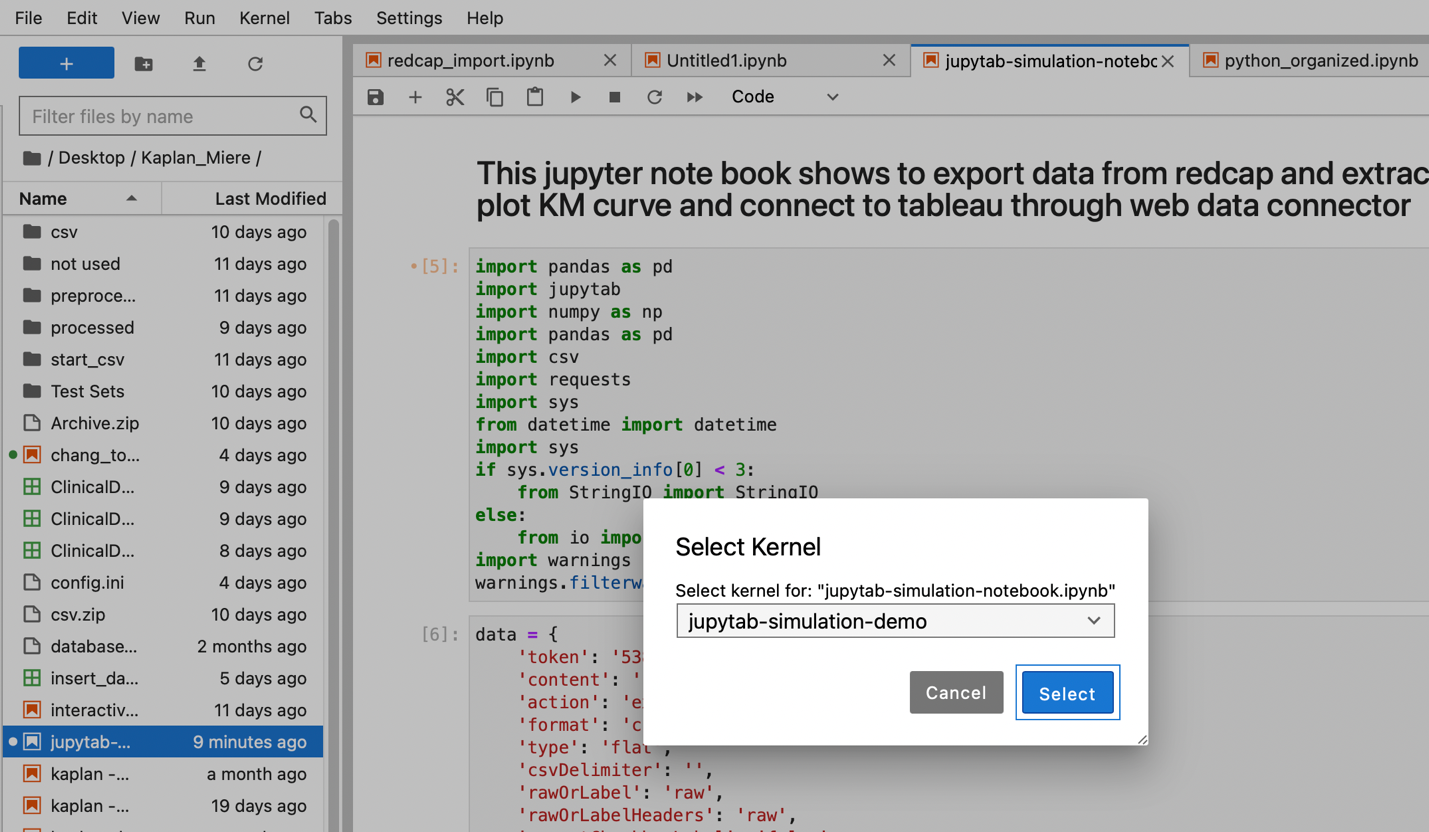
conda create -n jupytab-server-env python=3.7

conda activate jupytab-server-env

conda install jupytab-server=0.9.1 -> pip install jupytab-server

pip install sparkmonitor

Then we write the logic of getting the export data and prepare data to extracted data format inside the jupyter notebook and note to change the kernel we installed with the required dependency

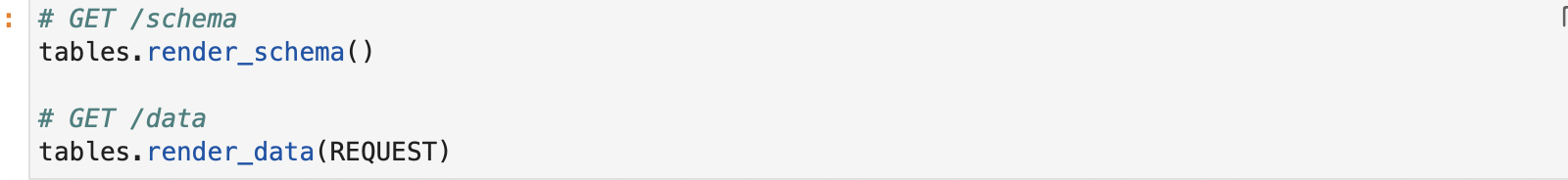


In the notebook, there are two parts of coding are necessary to make the data frame that used as the data source in Tableau:

1. Load data frame with Tables method in Jupytab so that it can be exposed to Tableau



1. Declare our data frame's schema and how the data exported when the notebook executed by jupytab server



Before running the server, the configuration file(config.ini) needs to be created.

Graphical user interface, text, application

Description automatically generated

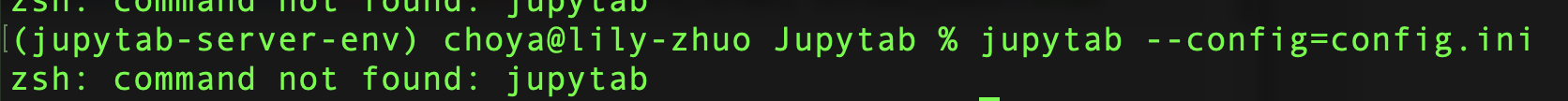
And more settings can refer to <https://github.com/CFMTech/Jupytab#configuration-file>

If security token provided, an encrypted security token will be required for all exchanges with Jupytab. (the address of server will be like http://<usr.name>.local:8123/?security\_token=3f9aa9e0493a8850c147349134b981e92fa136b822dbc6bcc559c7d3)

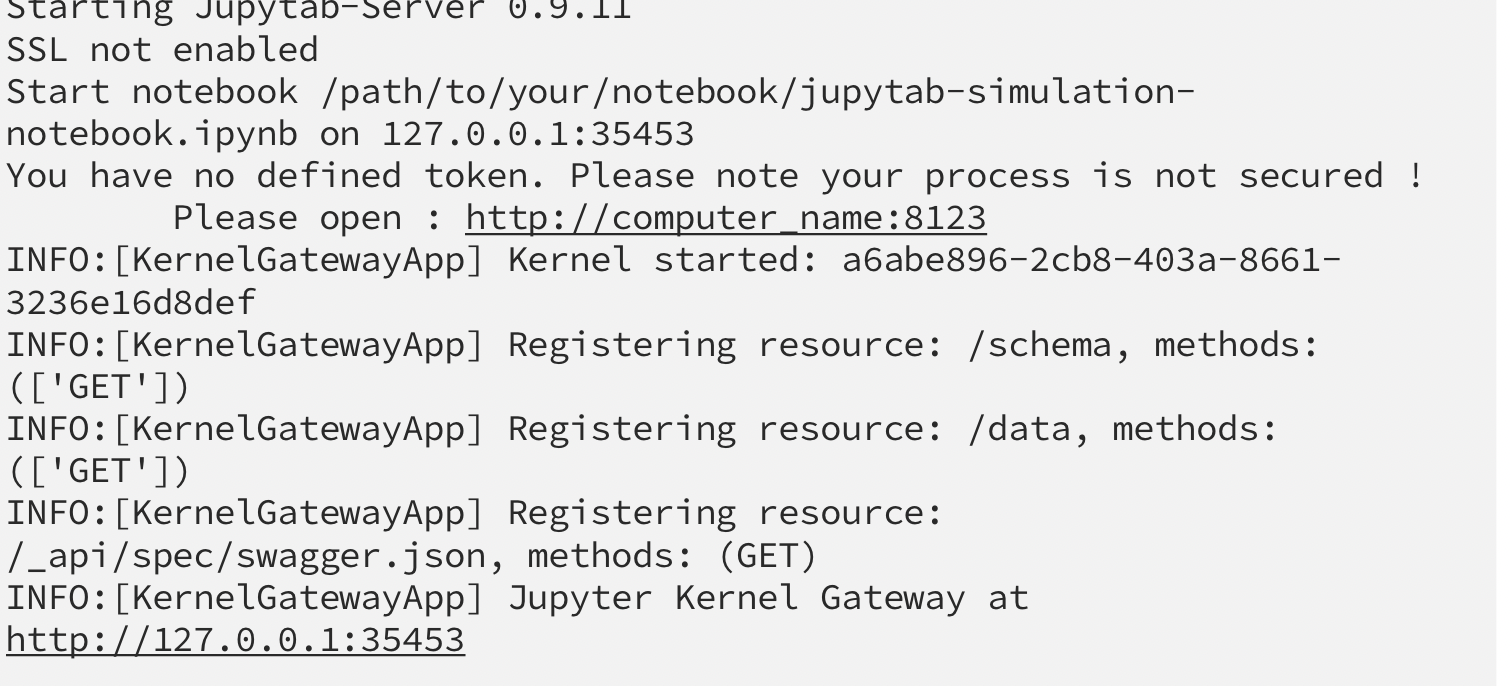
Jupytab contains settings for SSL that could further secure the server

The command for running the server is

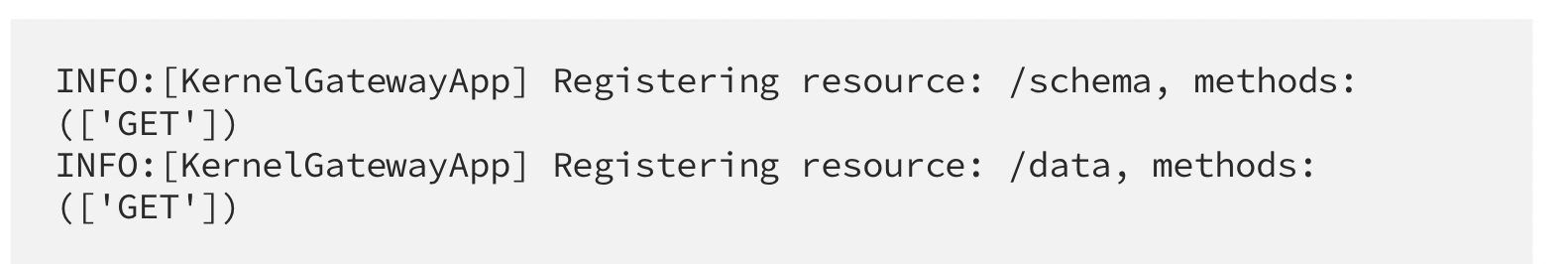
jupytab --config=path\_to\_config.ini

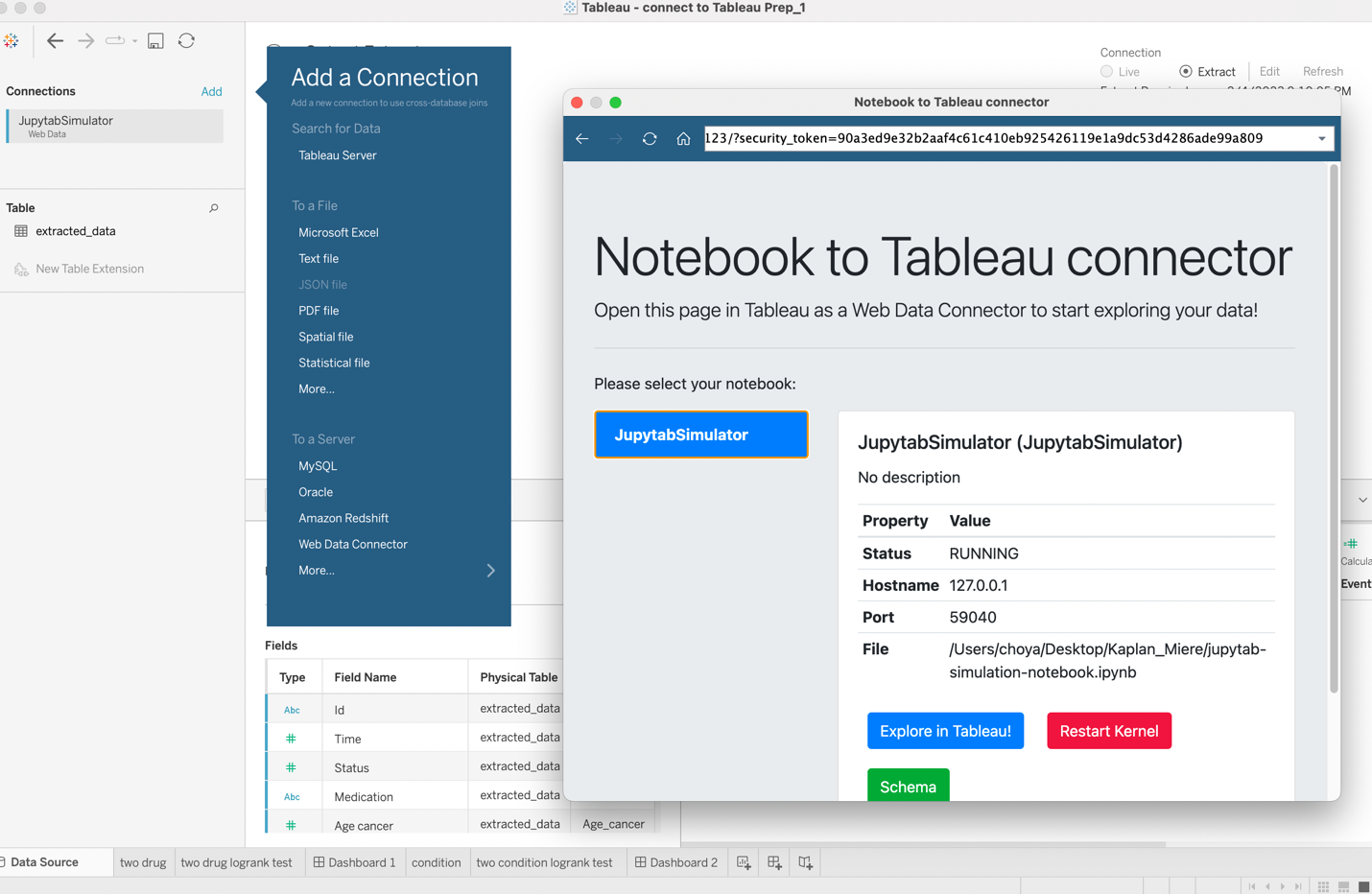


server In the terminal, it will shows content



And may need to wait few minutes to finish the process in jupyter notebook until the terminal shows log below then can go to Tableau to connect data



In Tableau, choose Web Data Connector-> to enter the address shown in the terminal and press enter to select the jupyter notebook-> Explore in Tableau

Reference: <https://towardsdatascience.com/interactive-simulation-with-tableau-and-jupytab-c26adb1be564>