Article

Piyush Adhikari · Dec 11, 2022 3m read

Embedded Python and IRIS on Jupyter Notebook in a virtual environment

I am documenting a demo of InterSystems IRIS featuring Embedded Python and Jupyter Notebook deployed on the same container, and an Embedded Python application developed on that Jupyter Notebook IDE.

I have used the Docker container created by @ Bob Kuszewski as a development environment to demonstrate how Embedded Python app can be developed in such a setting to push and retrieve data to and from InterSystems IRIS. The benefit of using this container as the development environment is that it is a virtual environment with Jupyter IDE and IRIS connected and running side by side. Using this setup over anything can be justified by the speed test of data ingestion and querying I did on multiple platforms and InterSystems IRIS offering the fastest rate of data ingestion and querying.

The Embedded Python application written on Jupyter Notebook fetches CSV data from an external datasets catalogue called data.world using Pandas, and persists the data in a class in IRIS running in the same container.

As the IRIS instance is running in a Docker container, there is no access to Studio, so I have used VS Code to create classes in the IRIS instance. We can connect to IRIS and program in ObjectScript using InterSystems's extensions for Servers Manager and ObjectScript, respectively.

Once data is persisted in IRIS, I have used SQL query to access the data from IRIS and stored it in a data frame.

```
imoprt iris

query = "SELECT Property, Property, Property, Property, Property, FROM
Namespace.Class"
iris.sql.exec(query)
```

Then, I have used Plotly, a library used for data visualization and analytics, to plot bar graph from the data stored in the IRIS class. I referred to the dash-python-iris for using pyplot python library for visualization.

Application's code

```
import pandas as pd

df = pd.read_csv('https://query.data.world/s/tus52dys57qbhqz4qjmla3r34pnuti')

number = df['Number']

name = df['Name']

symbol = df['Symbol']

marketcap = df['Market Cap']
```

```python
price = df['Price']

supply = df['Circulating Supply']

tfhr = df['Volume (24hr)']

import iris

for i in range(1515):

    num = number.loc[i]

    nam = name.loc[i]

    sym = symbol.loc[i]

    mc = marketcap.loc[i]

    pr = price.loc[i]

    sup = supply.loc[i]

    tf = tfhr.loc[i]


    setData = iris.cls("vizdata.vizdata")._New()

    setData.Number = str(num)

    setData.Name = str(nam)

    setData.Symbol = str(sym)

    setData.Marketcap = str(mc)

    setData.Price = str(pr)

    setData.Supply = str(sup)

    setData.TwentyFourHour = str(tf)

    setData._Save()

import iris

import plotly.express as px

import plotly.graph_objs as go

from plotly.offline import init_notebook_mode, iplot

query = "SELECT TOP 20 Name, Number, Marketcap, Price, Symbol,
TwentyFourHour FROM vizdata.vizdata"
df = iris.sql.exec(query).dataframe().sort_values(by='price', ascending = False)

print(df)
```

```
fig = px.bar(df.head(20), x="name", y="price", barmode="group", text_auto='.3s')

fig.update_traces(textfont_size=12, textangle=0, textposition="outside",
cliponaxis=False)

fig.update_layout(height=330)

fig.show()
```

Walkthrough video

https://www.loom.com/share/4c26cd5c719a48789b6a67295db816ed

Resources used

- Docker
- GitHub repo of the
  container: https://github.com/intersystems-community/iris-embedded-python-venv-jupyter
- Data.World: https://data.world/chasewillden/cryptocurrency-price-by-date-2013-february-2018
- Visual Studio Code
- Pyplot

References

1. Dash-Python: https://community.intersystems.com/post/dash-python-iris
2. Speed test documentation: https://usconfluence.iscinternal.com/x/SBwIQ

#Embedded Python #Python #Visualization #InterSystems IRIS #InterSystems IRIS for Health

Source
URL:https://community.intersystems.com/post/embedded-python-and-iris-jupyter-notebook-virtual-environment