Python Script to Import Market Data

How to Import Stock Data into Power BI Using Python: A Step-by-Step Guide

Are you looking to integrate stock data into Power BI? This blog post explores the use of Python scripts to automate the import process, making it easier to track and analyze stock market trends directly in Power BI dashboards.

Step 1: Convert 'YYYY-MM-DD' to UNIX Timestamp

The first step is to convert a date in the format YYYY-MM-DD to a UNIX timestamp. This is necessary because Yahoo Finance uses UNIX timestamps to specify date ranges for historical data.

```
import time

# Convert 'YYYY-MM-DD' to UNIX timestamp

def date_to_unix(date_str):

return int(time.mktime(time.strptime(date_str, "%Y-%m-%d")))

# Example

start_date = '2023-07-01'

unix_start_date = date_to_unix(start_date)

print(f"Start Date: {start_date} -> UNIX Timestamp: {unix_start_date}")
```

Output:

```
Start Date: 2023-07-01 -> UNIX Timestamp: 1688169600
```

Explanation:

- The function date_to_unix takes a date string as input and converts it into a UNIX timestamp.
- time.strptime is used to parse the date string into a struct_time object.
- time.mktime then converts this object into seconds since the epoch (UNIX timestamp).

Step 2: Get the Current Date in 'YYYY-MM-DD' Format

Next, we fetch the current date to use it as the end date for fetching historical stock data.

```
from datetime import datetime

# Get the current date in 'YYYY-MM-DD' format

current_date = datetime.now().strftime('%Y-%m-%d')
```

```
unix_current_date = date_to_unix(current_date)

print(f"Current Date: {current_date} -> UNIX Timestamp: {unix_current_date}")
```

Output:

```
Current Date: 2024-07-28 -> UNIX Timestamp: 1722124800
```

Explanation:

- datetime.now() gets the current date and time.
- strftime('%Y-%m-%d') formats this date into a string like ‘YYYY-MM-DD’.
- This string is then converted to a UNIX timestamp using the date_to_unix function.

Step 3: Read the CSV File to Get the List of Stock Symbols

The stock symbols (trackers) you want to download are stored in a CSV file. We’ll read this file into a DataFrame and extract the symbols.

```
import pandas as pd

# Read the CSV file to get the list of stock symbols

file_path = 'C:/Users/faisa/Downloads/Power BI_Imp Summary/Stocks Dashboard/trackers.csv'

symbols_df = pd.read_csv(file_path)

trackers = symbols_df['Symbol'].tolist()

print("Stock Symbols:")

print(trackers)
```

Symbol	T
AMZN	S
MSFT	S
BTC-USD	C
USDT-USD	C

CSV file Format: **Output (Example):**

```
Stock Symbols:

['AAPL', 'MSFT', 'GOOGL', 'AMZN', 'TSLA']
```

Explanation:

- pd.read_csv(file_path) reads the CSV file into a DataFrame.
- We assume the CSV file has a column named ‘Symbol’ containing the stock symbols.
- symbols_df['Symbol'].tolist() converts the ‘Symbol’ column to a Python list, which we store in the trackers variable.

Step 4: Initialize a List to Hold All Data

We’ll create an empty list to store the DataFrames for each stock’s historical data.

```
# Initialize a list to hold all data

data_frames = []
```

Explanation:

- This list data_frames will be used to accumulate DataFrames, one for each stock symbol.

Step 5: Fetch Data from Yahoo Finance

Now, we’ll loop through each stock symbol, construct the URL for Yahoo Finance, fetch the historical data, and store it in a DataFrame.

```
# Define the start and end periods using UNIX timestamps

period1 = date_to_unix(start_date)

period2 = date_to_unix(current_date)

for tracker in trackers:

url = f'https://query1.finance.yahoo.com/v7/finance/download/{tracker}?period1={period1}&period2={period2}&interval=1d&events=history&includeAdjustedClose=true'

try:

# Fetch data from Yahoo Finance

df = pd.read_csv(url)

df['Symbol'] = tracker # Add a column for the stock symbol

data_frames.append(df) # Append the dataframe to the list

print(f"Data for {tracker} downloaded successfully.")

except Exception as e:

print(f"Failed to download data for {tracker}: {e}")
```

Example Output:

```
Data for AAPL downloaded successfully.
Data for MSFT downloaded successfully.

Data for GOOGL downloaded successfully.

Data for AMZN downloaded successfully.

Data for TSLA downloaded successfully.
```

Explanation:

- We loop over each tracker (stock symbol) in the trackers list.
- A URL is constructed to fetch data from Yahoo Finance using UNIX timestamps (period1 and period2) for the desired date range.
- pd.read_csv(url) attempts to read the data directly from the URL, which Yahoo Finance serves in CSV format.
- A new column 'Symbol' is added to the DataFrame to identify which stock the data belongs to.
- The DataFrame is appended to data_frames for later concatenation.
- If there's an error during the download, it prints an error message indicating the failure.

Step 6: Concatenate All DataFrames into a Single DataFrame

Once all data has been fetched and stored in individual DataFrames, we concatenate them into a single DataFrame for easier analysis.

```
# Concatenate all dataframes into a single dataframe

combined_data = pd.concat(data_frames, ignore_index=True)

# Reorder columns for better readability

combined_data = combined_data[['Symbol', 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close',
'Volume']]

# Display the combined dataframe

print(combined_data.head())
```

Output:

```
Symbol Date Open High Low Close Adj Close Volume

0 AAPL 2023-07-01 192.020004 194.610001 191.889999 192.460007 191.362930 59021500

1 AAPL 2023-07-02 190.850006 193.160004 190.179993 192.800003 191.700000 48655000

2 AAPL 2023-07-03 192.470001 193.490005 192.100006 193.240005 192.150009 37538000

3 AAPL 2023-07-04 193.169998 194.479996 192.479996 193.149994 192.060000 50777000

4 AAPL 2023-07-05 192.800003 195.029999 192.600006 194.169998 193.070000 55548000
```

Explanation:

- pd.concat(data_frames, ignore_index=True) combines all individual DataFrames into a single DataFrame called combined_data.
- The columns are reordered to start with the ‘Symbol’ column for clarity.
- combined_data.head() prints the first five rows of the combined DataFrame to give a preview of the data.

Full Script

Here’s the complete script with all steps explained:

```
import pandas as pd
import time
from datetime import datetime
# Convert 'YYYY-MM-DD' to UNIX timestamp
def date_to_unix(date_str):
return int(time.mktime(time.strptime(date_str, "%Y-%m-%d")))
# Get the current date in 'YYYY-MM-DD' format
current_date = datetime.now().strftime('%Y-%m-%d')
# Define the start date for the data
start_date = '2023-07-01' # Adjust this as needed
period1 = date_to_unix(start_date)
period2 = date_to_unix(current_date)
# Read the CSV file to get the list of stock symbols and their types
file_path = 'C:/Users/faisa/Downloads/Power BI_Imp Summary/Stocks Dashboard/Source/trackers.csv' # Replace with the path to your CSV file
symbols_df = pd.read_csv(file_path)
trackers = symbols_df[['Symbol', 'Type']].to_dict(orient='records')
# Initialize a list to hold all data
data_frames = []
for tracker in trackers:
symbol = tracker['Symbol']
type_of_asset = tracker['Type']
```

```
url = f'https://query1.finance.yahoo.com/v7/finance/download/{symbol}?period1={period1}&period2={period2}&interval=1d&events=history&includeAdjustedClose=true'

try:

# Fetch data from Yahoo Finance

df = pd.read_csv(url)

df['Symbol'] = symbol # Add a column for the stock symbol

df['Type'] = type_of_asset # Add a column for the type (Stock or Crypto)

data_frames.append(df) # Append the dataframe to the list

print(f"Data for {symbol} downloaded successfully.")

except Exception as e:

print(f"Failed to download data for {symbol}: {e}")

# Concatenate all dataframes into a single dataframe

combined_data = pd.concat(data_frames, ignore_index=True)

# Reorder columns for better readability

combined_data = combined_data[['Symbol', 'Type', 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume']]

# Display the combined dataframe

print(combined_data.head())
```