Troubleshooting:

CSV File Creation Issue in Crypto API Automation

During the development of the `Automating Crypto Website API Pull` project, I encountered an issue where the CSV file was not being created or updated as expected. This section documents the steps taken to identify and resolve the problem.

Problem Description

Initially, the script was supposed to create a CSV file in the specified directory (`C:\Users\gusta\Desktop\Dri - Projects\CryptoAPI.csv`). However, despite running the script multiple times, the file was not being created or updated. This issue was puzzling as a test script for file creation worked without any problems.

Initial Steps and Observations

To isolate the issue, I created a basic test script that attempted to write a simple DataFrame to a CSV file in the same directory. The test script successfully created the file, confirming that the issue was not related to file path errors or directory permissions.

Debugging Process

The next step involved adding debugging statements throughout the original script. These statements provided real-time feedback on the script's progress, including successful API requests, data normalization, and the CSV writing process.

I also added a validation step to ensure that the API returned valid data before proceeding with the normalization and CSV writing processes. This step prevented the script from attempting to process empty or malformed data, which could have silently failed.

Solution and Changes

The solution involved a few key changes:

- Added Debugging Statements: Provided clear feedback at each step to monitor the script's behavior.
- Validated API Data: Ensured that the data received from the API was valid before processing.
- Ensured Proper CSV Handling: Explicitly handled the creation and appending of the CSV file to avoid any potential issues.

After implementing these changes, the script successfully created and updated the CSV file in the specified directory.

Conclusion and Lessons Learned

This troubleshooting process highlighted the importance of thorough debugging and validation in script development. By breaking down the problem and methodically testing each component, I was able to identify the root cause and implement a robust solution. This experience reinforces the value of clear, step-by-step debugging and error handling in preventing and resolving similar issues in future projects.

Code Snippets

Below is the updated code with the changes that resolved the CSV creation issue:

```python
import os
from time import sleep
import json
import pandas as pd
from requests import Session, ConnectionError, Timeout, TooManyRedirects

def api_runner():
    global df
    # Initialize an empty DataFrame if df doesn't exist
    if 'df' not in globals():
        df = pd.DataFrame()

    url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
    parameters = {
        'start': '1',
        'limit': '100',
        'convert': 'USD'
    }
    headers = {
        'Accepts': 'application/json',
        'X-CMC_PRO_API_KEY': 'your_api_key_here',
    }

    session = Session()
    session.headers.update(headers)

    try:
        response = session.get(url, params=parameters)
        data = json.loads(response.text)
        print("API request successful.")
    except (ConnectionError, Timeout, TooManyRedirects) as e:
        print(f"API request failed: {e}")
        return

    if not data or 'data' not in data:
        print("No data received from the API.")
        return

    df2 = pd.json_normalize(data['data'])
    df2['timestamp'] = pd.to_datetime('now')
    print(f"Data normalized. Number of records fetched: {len(df2)}")

    df = pd.concat([df, df2], ignore_index=True)
    print(f"Data concatenated. Total records in DataFrame: {len(df)}")

    csv_file_path = r"C:\Users\gusta\Desktop\Dri - Projects\CryptoAPI.csv"

    if not os.path.isfile(csv_file_path):
        df.to_csv(csv_file_path, index=False, mode='w', header=True)
        print(f"CSV file created at {csv_file_path}")
    else:
        df.to_csv(csv_file_path, index=False, mode='a', header=False)
        print(f"Data appended to existing CSV file at {csv_file_path}")

for i in range(333):
    print(f"Starting API Runner iteration {i+1}")
    api_runner()
    print('API Runner completed successfully')
    sleep(60)
```