**AUTOMATION AND CONTINUOUS UPDATING**

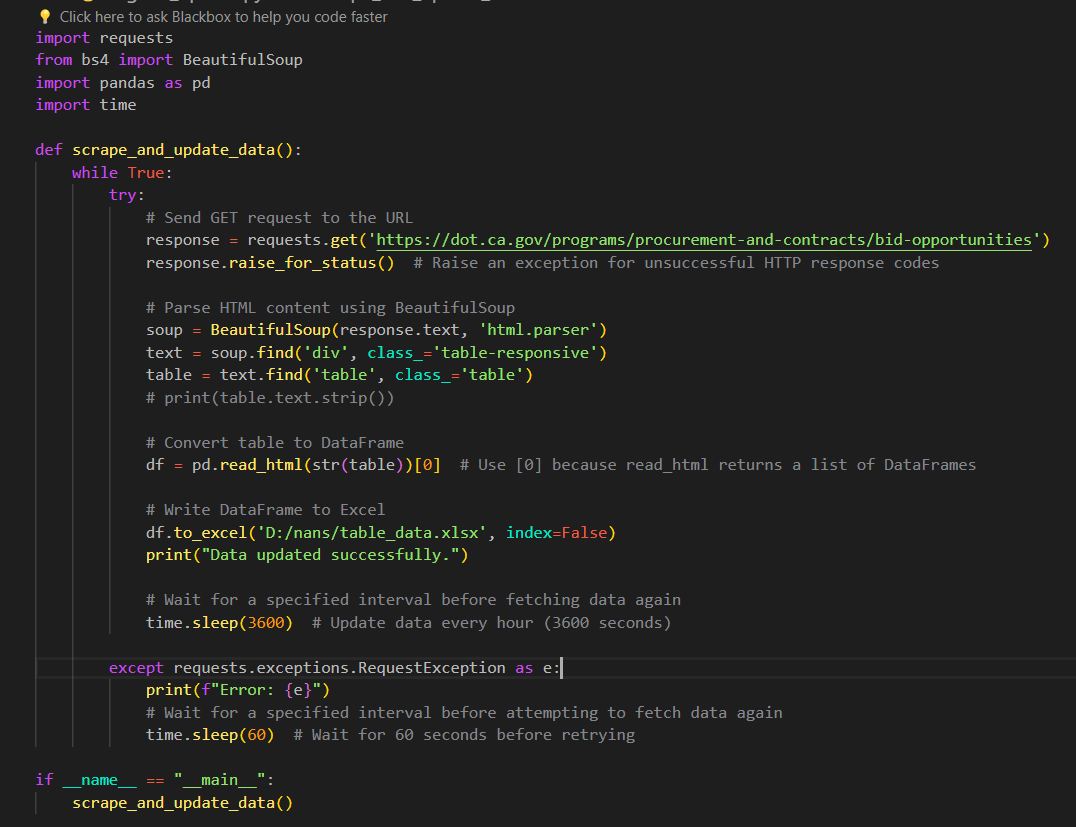
**1. Introduction:** Continuous updates of data sources are essential to ensure the relevance and accuracy of information for stakeholders. This documentation outlines our methodology for automating the process of running and updating data sources automatically.

**2. Data Source Monitoring:** Our system employs continuous monitoring techniques to detect changes or updates in the data sources. This involves periodic checks of the sources to identify new data, modified data, or changes in the source structure. Monitoring mechanisms are implemented to trigger data update processes when changes are detected.

**3. Automated Data Retrieval:** Once changes are detected in the data sources, automated data retrieval processes are initiated. These processes involve sending requests to the data sources using APIs, web scraping techniques, or other data access methods. Data is fetched from the sources in real-time or at scheduled intervals based on the update frequency requirements.

**4. Data Processing and Standardization:** Upon retrieval, the fetched data undergoes processing and standardization to ensure consistency and compatibility with existing datasets. Data processing tasks may include cleaning, filtering, transforming, and structuring the data to meet the desired format and quality standards. Standardization procedures are applied to normalize data attributes and ensure uniformity across different sources.

**5. Update Triggers and Scheduling:** Data update triggers and scheduling mechanisms are configured to automate the timing and frequency of data updates. Triggers are set to initiate the update process when specific conditions are met, such as changes in the data sources or predefined update intervals. Scheduling tools like cron jobs or task schedulers are utilized to execute update tasks at predetermined times or intervals.



**Figure 1 code of automate updation**

USE OF CRON JOBS:

1. **Automation:** Cron jobs automate the execution of data update tasks according to a specified schedule. This eliminates the need for manual intervention, reducing human error and ensuring that data updates occur consistently and reliably.
2. **Scheduled Execution:** With cron jobs, users can schedule the execution of data update scripts or commands at specific times or intervals. For example, a cron job can be set to run a Python script every hour, every day, or at other customized intervals based on the organization's requirements.
3. **Flexibility:** Cron jobs offer flexibility in scheduling, allowing users to define custom intervals for data updates. This flexibility enables organizations to tailor the update frequency to match the pace of data changes in their sources and the needs of their stakeholders.
4. **Resource Optimization:** By automating data update tasks with cron jobs, organizations can optimize resource utilization and streamline workflow processes. Instead of allocating human resources to manually execute data updates, cron jobs handle the task efficiently in the background.
5. **Error Handling and Monitoring:** Cron jobs typically provide mechanisms for error handling and monitoring. Users can configure cron jobs to send notifications or alerts in case of errors or failures during the execution of data update tasks. This allows for timely troubleshooting and resolution of issues to ensure uninterrupted data updates.
6. **Scalability:** Cron jobs are scalable and can be easily configured to accommodate changes in data update requirements or scale up to handle larger datasets. As data volumes grow or new sources are added, cron jobs can be adjusted accordingly to ensure continued smooth operation.