**Bug Data Collection and Update Script Documentation**

**Introduction**

The "Bug Data Collection and Update Script" (AutomatedTracker.py) is designed to automate the process of fetching bug data from the Bugzilla API, analyzing the data, and updating an Excel file with the collected statistics. This script leverages the **schedule** library to run the data collection and update process at regular intervals. It targets two categories of bug severities: "blocker/critical" and "major/normal."

**Functionality**

1. **Function Signature**
   * **collect\_and\_update\_data() -> None**

**Steps**

1. **Calculate Dates**
   * The script calculates the last 14 dates in the format MM/DD/YYYY for the horizontal axis of the chart.
   * The same dates in the format YYYY-MM-DD are calculated as well for querying the Bugzilla API for bug data.
2. **API Data Retrieval**
   * The script constructs URLs to query the Bugzilla API for " blocker/critical" and "major/normal" bugs.
   * The API key is used for authentication.
   * For each date in the last two weeks, the script fetches bug data for both categories.
3. **Data Processing**
   * The script counts the number of open " blocker/critical" and "major/normal" bugs created on and after each date searched.
   * The counters are stored in separate arrays.
4. **Bugs Created Calculation**
   * The script calculates the number of bugs created on a specific day by subtracting the count of bugs created on (and after) that day from the count of bugs created on (and after) the next day.
5. **DataFrame Creation**
   * The collected data is structured into a DataFrame.
   * The DataFrame contains columns: "Day," "Blocker/Critical Bugs," and "Major/Normal Bugs."
6. **Excel Interaction**
   * The script opens an existing Excel file containing macros (xlsm format).
7. Ensure that the Excel file (TestEmaildoc.xlsm) is saved to the file path "D:/OneDrive - TMEIC/Desktop/Bugzilla Tracker/TestEmaildoc.xlsm”
   * The DataFrame data is written to a specified range on a sheet named "Test1."
8. **Save and Close Workbook**
   * The Excel workbook is saved and closed.
9. **Quit Excel Application**
   * The Excel application is quit.
10. **Scheduled Data Collection and Update**
    * The script uses the **schedule** library to schedule the **collect\_and\_update\_data()** function to run at regular intervals (every 1439 minutes, approximately daily).
    * The script enters a loop to run the scheduled tasks indefinitely.
    * The **schedule.run\_pending()** function checks for pending scheduled tasks.
    * The script sleeps for one second between iterations to avoid excessive CPU usage.

**Usage**

1. Import the necessary libraries: **requests**, **json**, **pandas**, **datetime**, **timedelta**, **xlwings**, **schedule**, and **time**.
2. Run the script.
   * The script will automatically schedule the data collection and update process to run at regular intervals (every 1439 minutes + runtime of code 1 day).
3. Observe the Excel file specified in the script being updated with bug statistics for "critical/blocker" and "major/normal" severities over the last two weeks.

**Conclusion**

The "Bug Data Collection and Update Script" provides an automated way to regularly fetch and update bug statistics from the Bugzilla API. By utilizing the **schedule** library, the script ensures that bug data is collected and updated at specified intervals. The ability to fetch data for different severities and automatically update an Excel file enhances the bug tracking and reporting capabilities.