GitHub Activity Tracker

Emanuele Nuzzo

July 30, 2024

1 Objective

The objective of this assignment is to track activities on GitHub using the GitHub Events API. The application monitors up to five configurable repositories and generates statistics based on a rolling window of either 7 days or 500 events, whichever is less. These statistics are made available to end-users via a REST API, showing the average time between consecutive events for each combination of event type and repository name.

2 Features

- Monitors up to five configurable repositories.
- Generates statistics based on a rolling window of 7 days or 500 events.
- Provides a REST API to access the statistics.
- Minimizes requests to the GitHub API.
- Retains data through application restarts.

3 Assumptions

- The GitHub Events API is used to fetch events.
- The application is designed to handle up to five repositories.
- The rolling window is either 7 days or 500 events, whichever is less.
- The application is implemented in Python.

4 Installation and Setup

1. **Clone the repository:**

   ```
   git clone https://github.com/emanuzzo/github_activity_tracker.git
   cd github_activity_tracker
   ```

2. **Create a virtual environment and activate it:**

   ```
   python -m venv venv
   source venv/bin/activate  # On Windows use `venv\Scripts\activate`
   ```

3. **Install the required dependencies:**

   ```
   pip install -r requirements.txt
   ```

4. **Configure the repositories to be tracked:** Edit the `config.json` file to include the repositories you want to monitor.

5. **Run the application:**

   ```
   python app.py
   ```

5 Usage

- **API Endpoint:** `http://127.0.0.1:5000/stats`
- **Response Format:**

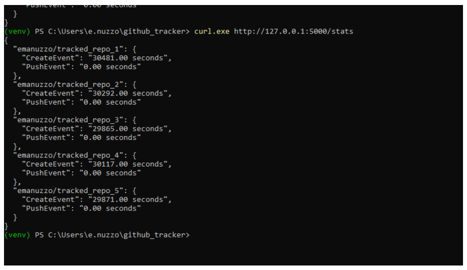
  ```
  {
    "repository_name": {
      "event_type": "average_time_between_events"
    }
  }
  ```

6 Code Overview

- **app.py:** main application file that sets up the Flask server and handles API requests.
- **config.py:** configuration file that allow to consume, with the generated token, the 5 repository api.
- **github_events.db:** SQL lite db created from the application at the first run.

7 Example Response

This is an example of the stats we can get from this application



```
  }
}
(venv) PS C:\Users\e.nuzzo\github_tracker> curl.exe http://127.0.0.1:5000/stats
{
  "emanuzzo/tracked_repo_1": {
    "CreateEvent": "30481.00 seconds",
    "PushEvent": "0.00 seconds"
  },
  "emanuzzo/tracked_repo_2": {
    "CreateEvent": "30292.00 seconds",
    "PushEvent": "0.00 seconds"
  },
  "emanuzzo/tracked_repo_3": {
    "CreateEvent": "29865.00 seconds",
    "PushEvent": "0.00 seconds"
  },
  "emanuzzo/tracked_repo_4": {
    "CreateEvent": "30117.00 seconds",
    "PushEvent": "0.00 seconds"
  },
  "emanuzzo/tracked_repo_5": {
    "CreateEvent": "29871.00 seconds",
    "PushEvent": "0.00 seconds"
  }
}
(venv) PS C:\Users\e.nuzzo\github_tracker>
```

8 API Documentation

8.1 GET /stats

- **Description:** Retrieves the average time between consecutive events for each combination of event type and repository name.
- **Response:** JSON object containing the statistics.

9 Conclusion

This application provides a robust solution for tracking GitHub activities across multiple repositories, offering valuable insights through a simple REST API.