

Notebook Documentation: Features and Functionality

This notebook implements a collection of utilities to interact with the public GitHub API, focusing on querying, filtering, and storing relevant information about repositories, files, and commits. The main implemented features and functionalities are described below:

General Configuration

- Import of key libraries: requests (HTTP requests), pandas (data manipulation), os (file management), logging (execution messages), and time (control of time between requests).
- Logging configuration to display informational and error messages during execution.
- Definition of the base URL for the GitHub API and optional handling of a personal authentication token (GITHUB_TOKEN) to increase usage limits and access more endpoints.

Utility Functions

- Adding authentication headers to requests if a token is provided.
- Saving JSON responses to .json or .csv files, facilitating the analysis and reuse of downloaded data.
- Recursive extraction of nested fields in JSON responses to filter and structure relevant information.

Rate Limit Verification

- Function to check the current state of the API rate limit, displaying information such as total, remaining, and used requests, as well as the time until the counter resets.

API Call Functions

- Four API endpoints are used in the notebook. For each, a function was developed to manage the parameters each endpoint accepts:
 - Public repositories
 - Organization repositories
 - Repository contents

- Commit listings
- The commit query is designed to handle limited responses and iterate until all information is received, with adjustable parameters such as cooldown time and the number of attempts to make.
- Filtering of JSON responses from the API to make the information more readable.
- Examples of filtering and extracting nested fields useful for customized analysis of GitHub information.
- Example of querying and visualizing the obtained data, displaying tables with relevant information about repositories, files, and commits.
- Example of iterative querying to the API until the process is forced to stop due to multiple unsuccessful API calls after reaching rate limits.

In summary, this notebook allows for flexible querying, filtering, and saving of information from the public GitHub API, automating common tasks of extraction and analysis of repository, file, and commit data.