Please write a command-line program that parses an input textual log file from a benchmark run, extracts data from it and saves it in a database.

Your program may take additional parameters, as needed.

Data fields required:

* Benchmark run properties:
* Some description (string)
* Execution environment description (string)
* Date when benchmark has been performed
* Server name where benchmark has been performed (string)
* Application version (string)
* Test suite version
* Testcase properties:
* Method API
* Method name
* Parameters (no need to parse them, just store them as string or json object)
* Testcase execution time in seconds (floating number)

Your program should parse the input log file, ignore all nonmatching lines and collect above data. It should then store the extracted data for the benchmark run, associated testcases and time for each testcase in a SQL database.

It is recommended to create a dictionary of testcases and create relationships between benchmark run, testcases and their time.

We assume that testcases usually are reused for subsequent benchmark runs, and defined relationships should allow to query for testcase execution times across several benchmark runs recorded in the database.

Preferred implementation platform: python 3.x and PostgreSQL (please use a programming language and SQL database of your choosing if you are unfamiliar with the recommended ones).

Submit your program to your github.com repository dedicated to this task.

Please contact us on for questions or if the task description is not clear enough.

An example input log file is provided in the attachment.

Below are the lines from the input file we are interested in. All other lines should be ignored.

Sending {"jsonrpc": "2.0", "id": "1", "method": "API.some\_method", "params": {"param1": "x", "param2": "y", "sort": "blog", "limit": "21"}} for reference time measurement

Got response in 0.6832s

Sending {"jsonrpc": "2.0", "id": "1", "method": "API2.some\_method1", "params": {"param1": "z", "param2": "d", "sort": "blog", "limit": "21"}} for reference time measurement

Got response in 0.4827s