Hi Yaakov,

Your mission (if you choose to accept it…) is to create an app which aggregates data on Python projects in Github.

**Guidelines**

The web application, DB and backend logic should be implemented on the same physical server.

Both web app and the DB should be running under separate containers.

**The stack**

* Development language should be Python.
* You can choose the backend framework to be any of the popular web frameworks (Django, Flask etc.)
* Client side should be implemented as curl/Postman/other simple method to generate http requests.
* Use any DB that serves the logic as long as you can explain why you chose it.
* Each component should be executed in a separate container – using ‘docker run’ commands or docker-compose.

**Server-side logic**

* You are given a simple module that connects to Github's public API.
* Using that module, you need to gain data on Github Python projects requirements files and store them to DB.
* For each of the Python projects you scrap (using the API) you need to save to DB all packages from their requirement file.
* You can ignore all packages that are not in the format of:
  + Packagename
  + Packagename=<version>

**Client-side logic**

* The client will be used only for 2 operations:
  + Trigger the scraping process for a specific number of python projects.
  + Fetch results from the DB on a single project (the client will send the name).

**The Github module**

* For both classes you need to use a valid Github token.
* If you don’t have a Github account create one and use the token it generates.
* Pip modules you need in order to use the module: pygithub, requests

**Github Client**

**get\_popular\_repos –** returns a list of popular repositories in python

**fetch\_repo\_archive\_url –** given a repo name, returns a link to an archive url of the repository

**Github Repo**

**get\_python\_req\_file\_single\_api –** return the contents of the requirements.txt file in the repo

**Other things to consider**

* Consider that some of the Github functions have long execution time and they should not be running inside the web framework thread.
* Note that the client might be triggered multiple times and should be optimized accordingly.
* Logical future development might be getting statistics on the requirement files contents – keep in mind when designing the data representation in the DB.
* Please work according to acceptable coding style and convention.

Good Luck!