Python developer technical task

Technical task is composed of two coding exercises, second one is optional and both include in the title an estimation about how long the task should take you. The task should be completed within the next 3 days after receiving it. You can send your code just answering the email where you received it with a file attached to it, link to repository,...

Github crawler (2-4 hours)

We want you to code a GitHub crawler that implements the GitHub search and returns all the links from the search result, requisites are:

- Python 3
- The crawler should be as efficient as possible (fast, low memory usage, low CPU usage,...)
- Input:
    - Search keywords: a list of keywords to be used as search terms (unicode characters must be supported)
    - List of proxies: one of them should be selected and used randomly to perform all the HTTP requests (you can get a free list of proxies to work with at https://free-proxy-list.net/)
    - Type: the type of object we are searching for (Repositories, Issues and Wikis should be supported)
- Documentation about how to use it should be included
- Output: URLS for each of the results of the search
- The code should also include unit tests with a minimum code coverage of 90%
- For the purpose of this task you only have to process first page results
- For the purpose of this task we want to work with raw HTML and API can't be used
- You can use any library you consider useful for the task (e.g. HTTP libraries, parser libraries,...) but not frameworks (e.g. Scrapy)

Example

Keywords: “openstack”, “nova” and “css”

Proxies: “194.126.37.94:8080” and “13.78.125.167:8080”

Type: “Repositories”

So the input will be a JSON containing:

```
{
  "keywords": [
    "openstack",
    "nova",
    "css"
  ],
  "proxies": [
    "194.126.37.94:8080",
    "13.78.125.167:8080"
  ],
  "type": "Repositories"
}
```

Expected results:

JSON object containing:

```
[
  {
    "url": "https://github.com/atuldjadhav/DropBox-Cloud-Storage"
  }
]
```

Proxies from the example may not be available and results could be different by the time you read this document.

Example 2

Input:

```
{
  "keywords": [
    "python",
    "django-rest-framework",
    "jwt"
  ],
  "proxies": [
    "194.126.37.94:8080",
    "13.78.125.167:8080"
  ],
  "type": "Repositories"
}
```

Output:

```
[
  {
    "url": "https://github.com/GetBlimp/django-rest-framework-jwt"
  },
  {
    "url": "https://github.com/lock8/django-rest-framework-jwt-refresh-token"
  },
  {
    "url": "https://github.com/City-of-Helsinki/tunnistamo"
  },
  {
    "url": "https://github.com/chessbr/rest-jwt-permission"
  },
  {
    "url": "https://github.com/rishabhiitbhu/djangular"
  },
  {
    "url": "https://github.com/vaibhavkollipara/ChatroomApi"
  }
]
```

Extra information (optional task, 45m - 1h 30m)

In the previous task we asked you to implement GitHub search and return just the links, now we want the crawler to be extended so the following information is extracted for each repository link:

- The owner of the repository
- Language stats

You don’t have to return any other extra information for the rest of the link types (“Wikis” and “Issues”)

Example

For the repository in the first example above expected results should be now a JSON like:

```
[
  {
    "url": "https://github.com/atuldjadhav/DropBox-Cloud-Storage",
    "extra": {
      "owner": "atuldjadhav",
      "language_stats": {
        "CSS": 52.0,
        "JavaScript": 47.2,
        "HTML": 0.8
      }
    }
  }
]
```