**Steps to Follow-**

1. check for input queue plug in
2. check on coupling of classes, check on exception handling, clear comments
3. use docker, mongoDB variant

**Links -**

1. https://stackoverflow.com/questions/21907278/github-api-using-repo-scope-but-still-cant-see-private-repos

To access private information, you need to include the access\_token parameter in the URL .

**Example:**

/users/someusername/repos?access\_token=512295a0afb73bdd1c076a00c69f8abcd12345

Access tokens are generated per user. You can generate your personal access token here:

https://github.com/settings/applications

2. https://developer.github.com/v3/repos/commits/

3. **Examples:**

curl -H "Accept: application/vnd.github.mercy-preview+json" https://api.github.com/search/users?q=chunky+garg+in:name+location:gurgaon

curl -H "Accept: application/vnd.github.mercy-preview+json" https://api.github.com/users/chunky-garg/repos

curl -H "Accept: application/vnd.github.mercy-preview+json" https://api.github.com/search/users?q=juhi+ghosh+in:name+location:hyderabad

1. OAuth Access Token
2. header("Authorization", "token c9e1ace0bc151c33a134e2f91848fda236d3a992")

c9e1ace0bc151c33a134e2f91848fda236d3a992

For service account use this to read rate limit at first-

curl -i -u juhi-ghosh:c9e1ace0bc151c33a134e2f91848fda236d3a992 https://api.github.com/user

5. create service account programatically-

**Assumptions -**

1. Accurate results – Higher score user returned per developer profile search is processed
2. Commits developer name checked for exact match with input developer name
3. Username/Authorisation token for a service account will be taken as input to generate the Oauth authorisation token

Authentication not required, following the oauth concept -

Also, because

Deprecation Notice: GitHub will discontinue password authentication to the API.

You must now authenticate to the GitHub API with an API token, such as an OAuth access token,

GitHub App installation access token, or personal access token,

depending on what you need to do with the token.

1. When user not found no record added Ex – *Sharvi Verma, India*
2. Git over HTTPS
3. Did not remove Sysouts to understand code flow
4. https://developer.github.com/v3/#rate-limiting

**Suggest any additional parameters**

1. As the [docs](https://developer.github.com/v3/repos/) indicate, these methods take a type parameter that can filter the repositories returned based on what type of access the user has for the repository. In this way, we can fetch only directly-owned repositories, organization repositories, or repositories the user collaborates on via a team.
2. Fetch items based on score match check against value > 30
3. Assumptions point 1
4. Deliverables pt 6 – scale application by running multiple threads for AppRunner on different/same machines
5. Register callback in parser and use it as a library
6. Multiple Github users identified as response

**Doubts –**

1. Authorized access or not
2. Public profile details – what all fields to read

Choose ideal output format to store and represent the collected results ?

**Asynchronous Processing**

1. Keep on reading the Large data file , not waiting for the requests for the github profile finish to fetch data .
2. Keep on writing/processing data to output file, meanwhile more requests for github are handled.
3. No interdependency between 2 github requests
4. Keep a monitor thread to check on the rate limit periodically - observer
5. Consider if there are 10 input files

**Standard Output 1**

{

"total\_count": 1,

"incomplete\_results": false,

"items": [

{

"login": "chunky-garg",

"id": 7038167,

"node\_id": "MDQ6VXNlcjcwMzgxNjc=",

"avatar\_url": "https://avatars1.githubusercontent.com/u/7038167?v=4",

"gravatar\_id": "",

"url": "https://api.github.com/users/chunky-garg",

"html\_url": "https://github.com/chunky-garg",

"followers\_url": "https://api.github.com/users/chunky-garg/followers",

"following\_url": "https://api.github.com/users/chunky-garg/following{/other\_user}",

"gists\_url": "https://api.github.com/users/chunky-garg/gists{/gist\_id}",

"starred\_url": "https://api.github.com/users/chunky-garg/starred{/owner}{/repo}",

"subscriptions\_url": "https://api.github.com/users/chunky-garg/subscriptions",

"organizations\_url": "https://api.github.com/users/chunky-garg/orgs",

"repos\_url": "https://api.github.com/users/chunky-garg/repos",

"events\_url": "https://api.github.com/users/chunky-garg/events{/privacy}",

"received\_events\_url": "https://api.github.com/users/chunky-garg/received\_events",

"type": "User",

"site\_admin": false,

"score": 33.70004

}

]

}