Freie Universität Berlin

Institute for Computer Science of Freie Universität Berlin

QIFTool - Documentation

ver1.0 - 21.10.2020

Robert Selack

robfu@zedat.fu-berlin.de

QIFTool or Query Issue Finder-Tool is a project created as a bachelor's thesis. It aims to help in the quality research field of technical debt by provoding relevant discussions regarding these debts. The discussion are presented in form of issues from github. The tool uses keywords and additional metrics to find potentially interesting issues. Although it is meant for the field of techincal debt the tool can be also be used to return all different kind of issues' topics.

Inhaltsverzeichnis

1 How to run the program

1. Download the 'qiftool.py' and 'requirements.txt' files from the repository
2. Place both files in the desired location
3. Open the terminal and navigate to the files' location
4. Install all dependencies by running 'pip3 install -r /path/to/requirements.txt' or just 'pip install -r /path/to/requirements.txt' depending on your python version
5. Run the program by using 'python3 qiftool.py'
6. By running it for the very first time, the tool should have created a 'config.ini' file inside the tool's folder. Fill out the necessary parameters following the instructions in 2
7. With the 'config.file' filled out run the programm again just like in step 5
8. The tool should now operate properly and an interactive mode will be seen. Follow 3 for further instructions

2 Configuration File - config.ini

This file is created by running the program for the very first time. It is used to give the user a space to use their own parameters used by the tool. The file contains three sections for the user to fill out.

```
[DEFAULT]
path_of_database = current
path_of_download = current

[credentials]
github_api_key = c2c1d13983cbb8c1d9ce7845c20d7937ba7c25a0
google_api_key = AIzaSyCWoAxVmPxqD9KZmD0_7YWx7dEkX28nwGs
google_cse_id = de940461e6efec5b9

[metrics]
keywords = technical debt   refactor    rewrite
issue_comments = 5
repo_contributors = 50
```

1. `[DEFAULT]`

   - this section contains the path for the database and downloaded repositories to be stored in. The user is able to create their own path with the location of the ’qiftool.py”s as a pivot. These can be changed by providiung a valid path on your machine.

2. `[credentials]`

   - this section contains the corresponding credentials necessary to run the used APIs

   (a) `github_api_key`

      i. register on github

      ii. use this link and click on ’generate new token’ to create a new key

iii. paste the key as a parameter

(b) `google_api_key`

i. register on google

ii. use this link and click on 'Get a Key' to create a new key

iii. either choose a project or create a new one

iv. follow the instructions and paste the key as a parameter

(c) `google_cse_id`

i. login to the google account created in the prior step

ii. use this link and click on the project you used to create the google key with

iii. look for the 'Search engine ID' and paste the ID as a parameter

3 Interactive Mode

Once you successfully configered the configuration file in 2 an interactive mode will be seen on the console after running it. In this mode the program will wait for the user

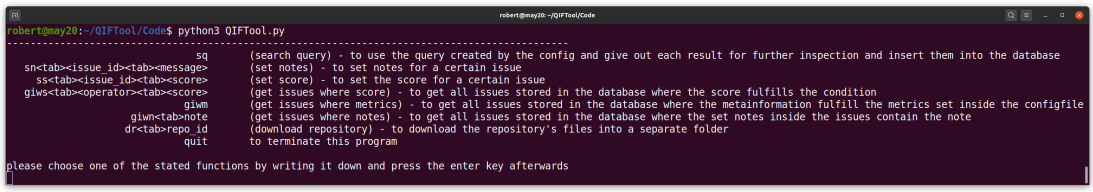

Abbildung 1: default interactive mode

to simply type a desired function into the console and confirming it by pressing enter. After being done with a function the program goes back to displaying the interactive mode as it loops itself around it.

3.1 Functions

function	description
sq	(search query) - start the google search. The metrics set in the configuration file will be used to determine what results will be found and shown.
sn<tab><issue_id><tab><message>	(set notes) - sets a note for a certain issue inside the database **issue_id** - a string of numbers. Found within the *issue_id* field in either the ouput or database of the issue. **message** - a string of characters that will be inserted into the *notes* field inside the database.
ss<tab><issue_id><tab><score>	sets a score for a certain issue inside the database. **issue_id** - a string of numbers. Found within the *issue_id* field in either the ouput or database of the issue.**score** - a number chosen by the user to represent its relevance.
giws<tab><operator><tab><score>	(get issues where score) - displays all issues stored in the database where the score fulfills the condition set by the user. **operator** - all comparison operators allowed by the SQL. <, >, =, <=, >= **score** - a number chosen by the user to represent its relevance and compare the issues inside the database to.

function	**description**
giwm	(get issues where metrics) - displays all issues stored in the database where the pieces of metainformation fulfill the metrics set inside the configfile. This function yields the same functionality as the 'sn'-function but with the database being the source.
giwn<tab>note	(get issues where notes) - displays all issues stored in the database where their notes contain the note set by the user with this very function **note** - a string of characters. This can be used in conjunction with SQL-syntax like providing a ör % around the note.
dr<tab>repo_id	(download repository) - downloads the repository's files into a separate folder. This folder's location is set by the configuration file. The strucutre of the downloaded files also is identical to that of its respository.
quit	terminates this program.