

Table of Contents

- 1. How to obtain the source code2
- 2. The layout of our directory structure.....2
- 3. Self explains of source codes3
 - Client3
 - Server3
- 4. How to build the software.....6
 - 4.1 Server.....6
 - 4.1.1 set up Django.....6
 - 4.1.2. set up nginx, MySQL server.....6
 - 4.1.3. set up middleware_gunicorn6
 - 4.1.4. run server6
- 5. How to test the software6
 - 5.1 Extension test.....6
- 6. How to release a new version of our software7
- 7. How to access the list of outstanding bugs and how to resolve a bug.....7

1. How to obtain the source code

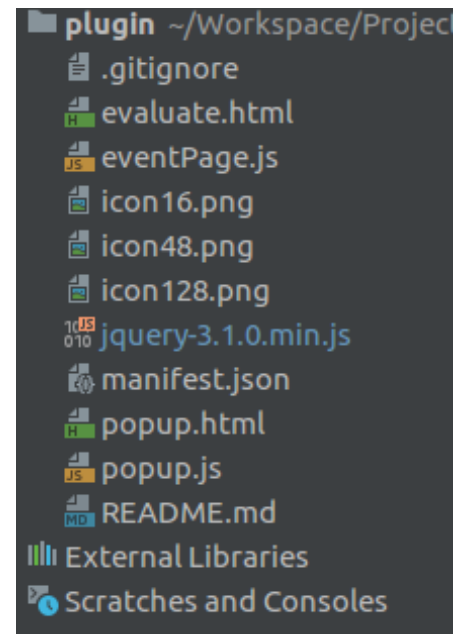
We use GitHub to share code, so the source code could be obtained through GitHub. Because our program is open source program, anyone who want source code can obtain a code through GitHub. You should access the link below.

client: https://github.com/summaryhelpex/SH_ChromeExtension_Client

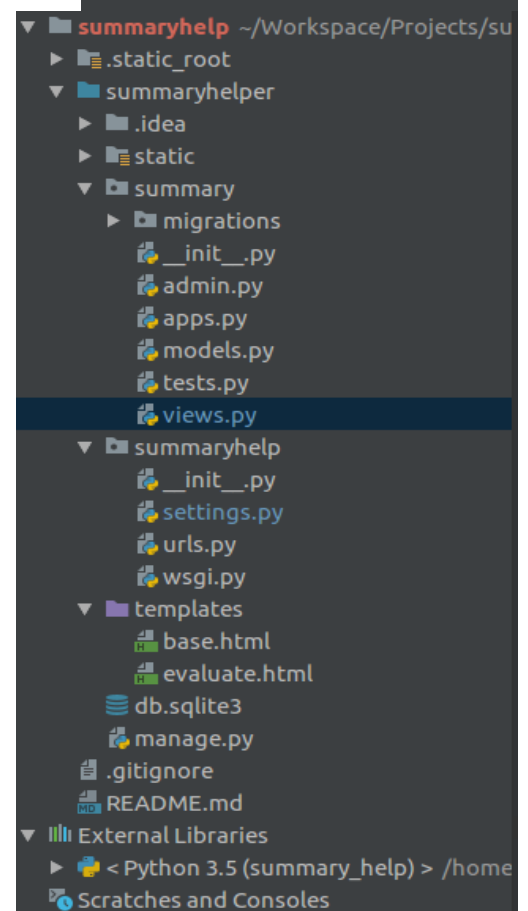
Server: <https://github.com/summaryhelpex/summaryhelp>

2. The layout of our directory structure

1. SummaryHelp client part basically follows the directory structure of the directory structure of the web extension project. And each class is organized in the following directory structure.



2. SummaryHelp api server part is organized in the following directory structure



3. Self explains of source codes

Client

templates directory

manifest.json

The manifest.json file is the main part of the extension. This section describes the version of the extension, extension description, and determines which popup window to display when running and specifying files that run in the background.

popup.html

The html basically covers basic functions when the program is executed. This refers to base.js, where each function is as follows.

◆ chrome.storage.sync.get(['total'],function(selectedText)

When you drag and run the part you want to interpret, a popup window will open in the upper right corner and the contents you dragged will appear in the textarea on the left.

◆ ('#submit').click(function ())

It contains the functions used after the summary button is clicked. First, the text is converted to send to the server, then the text is sent through the contents of the .ajax (), and the evaluation items are displayed.

evaluate.html

evaluate.html is html file to configure evaluate web page. if you click 'summary' button in base.html, after submit function is done, the evaluate.html is added by evaluate-load function in base.js. evaluate page will be located at the bottom of base page

◆ \$('eval').click(function())

After you invoke evaluate.html, you can click on the span stars ('.starRev span') using click (function ()).Then send the score to the server via the internal .ajax, like the original part of the above textarea, and initialize both textareas.

jquery-3.1.o.min.js

jquery.js is one of javascript library. It is easy to implement DOM-related processing, consistent event connection, easy to visualize, and used to interoperate with Ajax and resolve compatibility issues. More detail refer to <http://jquery.com/>

Server

summaryhelp directory

setting.py

This is where the basic configuration files for the Django api server are located. You can set the debug mode

and register the summary application. You can set up a database connection to use Django orm, and a path to manage static files and template files.

📄 urls.py

It contains the URL of adminpage, summary Contains all the URLs of the application and passes the data to the URL file in the summary directory.

📁 summary directory

📄 admin.py

Register the table in the admin page to see the database at a glance.

📄 apps.py

Register the name of the application.

📄 models.py

Django build database with orm and generate table Article table Generate summary table

📄 views.py

◆ getsummary function

The getsummary function is a function that is linked with model. If you put the article you want to summarize into the input, the summary value appears as the result in the model.

◆ storedb function

the storedb function stores the article text requested by the user, the time requested by the user, and the summary text in the database.

◆ view function

The view function is the function associated with the test pagethis function is used to view real-time requests from the client using the web before deploying to the actual server.

◆ Ajax view function

the Ajax view function is a function that receives the requested article from the client, summarizes it, and returns the json type of the ajax communication. and use the storedb function here to store the article and summary.

◆ eval_ajax_view function

eval_ajax_view is a function that stores the evaluation received from the user in the database.

📄 modelmanage.py

◆ starmanage class

controls user's score for Rake scoring module.

◆ extractkeyword class

● update_starscore function

update_starscore function updates starscores

- `get_Keyword` function

`get_Keyword` function returns list of keyword using two keyword extracting module and evaluate the final keyword list. by using Rake and Yake.

in the code, Rake weight used when computing each keyword's weight. We assumed that if result is nice user will give high score. And this is highly related with the contribution of each model. We show total 20 result of the models. 10 per each model. And show only for 5 of these. Sorted with scores.

`star_score` only affect to rake model. And if the rake model was good for the previous result, it's score will be high and will show more than the previous result. But if it was not good, this will get low score and its total percent in result will be gradually decrease.

- `Yake.py`

this code is for Yake (Yet Another Keyword Extractor¹)'s implementation of paper. with `extract_keywords` function, after inputting string of main string, keyword list is returned.

- 📁 `static_root` directory

This directory is a collection static files for the admin pages and test pages, collected in this directory all at once for management. It contains the extension's main icon and logo, css, and javascript files.

- `sh_icon.png`

It is the default icon of the extension program and it is shown in the address bar. This means selecting a key keyword out of a lot of information.



- `sh_logo.png`

This is the logo of our program shown in the header at the top of the main pop-up.



- `style.css`

It is a file for designing UI of main popup window. We focused on convenience from the user's point of view.

- `popup.js`

It is a file defined for drag and button click event.

¹ Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2018). A Text Feature Based Automatic Keyword Extraction Method for Single Documents Proceedings of the 40th European Conference on Information Retrieval (ECIR'18), Grenoble, France. March 26 – 29. https://link.springer.com/chapter/10.1007/978-3-319-76941-7_63

4. How to build the software

4.1 Server

4.1.1 set up Django

SummaryHelp_Server use Django for Python, so you need to set up python and Django for python.

To set up python: <https://www.python.org/>

To set up Django: <https://docs.djangoproject.com/ko/2.2/intro/install/>

4.1.2. set up nginx, MySQL server

summaryhelp is running on a web server basis, I have built a MySQL server.

```
sudo apt-get install nginx
```

```
sudo apt-get install MySQL-server
```

4.1.3. set up middleware_gunicorn

DIt is middleware that connects nginx and Django, so it should be installed to build an efficient web environment.

This middleware also allows us to turn our api server into a daemon program.

```
sudo apt-get install gunicorn
```

4.1.4. run server

1. on the server

Enter in the directory where the manage.py file is located

```
gunicorn summary_help.wsgi:application --bind=0.0.0.0:8000 --daemon--reload
```

2. on the local

Enter in the directory where the manage.py file is located

```
python manage.py runserver(django test server)
```

5. How to test the software

▼ Local Unit Testing.

5.1 Extension test

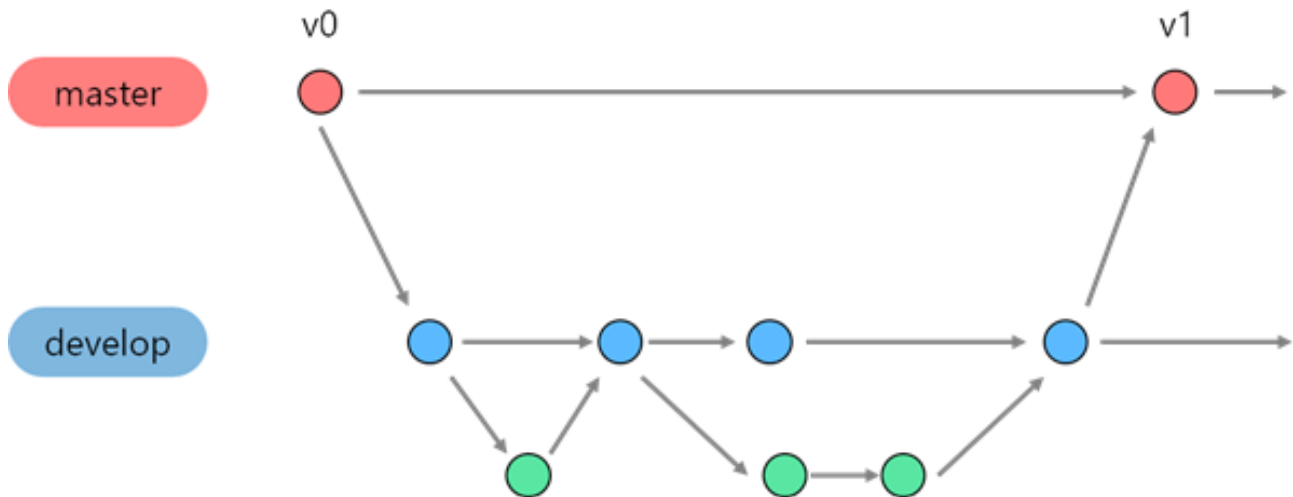
1. Drag function Test

If you drag non-article, copied article can be in the wrong format so if you need to test for dragging, you only drag text.

2. Server Connect Test

if you click summary button or evaluate button and our program is connected to the Server, Success message will be generated. if not, Fail message will be generate.

6. How to release a new version of our software



Our git strategy is 'git-flow'. Please refer to above image.

Master branch manage the version of application. That is, a new version of application is released through Master branch. The process of development will occur in Develop branch. After enough development has been done, updated source code will merge to Master branch. A new version is released to user.

Merge is only possible if other developers leave a review and approve the changes.

7. How to access the list of outstanding bugs and how to resolve a bug

if you find a bug, please let me know by using

<https://forms.gle/dvK8CUGMejPbJrcG8>

You can freely send requests for bugs that you have experienced using our program. You should specify how to reproduce the problem so that we can reproduce the bug. Also, if you want to receive our feedback, you can write your email address and get feedback.