My first open source contribution.

Marietta Lazana - t8160057@aueb.gr

May 2019

Abstract

Contributing to an open source project has proven to be extremely beneficial. It, greatly, reduced inhibitory factors by demystifying the contributory process. Personally, it was a challenge, to examine my abilities and strengths in a quite big and very fast moving project. Facing the real world of Software Engineering was surely a great experience!

Introduction

The "Software Engineering in Practise" course is one of the most important in my undergraduate degree, since it has prompted me to gain my first serious experience in this field. Under the supervision of the professor Diomidis Spinellis and the help of PhD candidate Antonis Gkortzis I gained useful knowledge in software engineering that helped me produce as much professional code as I could at this time. I learned about software design, construction, testing, maintenance, management and more, following the book of IEEE "SWEBOK, Software Engineering Body of Knowledge". Last but not least, I gained a better understanding of git that helped me work on the open source project I chose.

1 Project Understanding

Figure 1: ArchiveBox logo

I had two criteria while searching for a project. The first was high difficulty and the second was to be something that will enthusiasm me. So when I found ArchiveBox I thought that it was a great idea and also when I start reading the

code I didn't understand anything. So I was sure that this was going to be a project that will definately help me improve.

ArchiveBox[1] is a very promising and upcoming project created by Nick Sweeting (GitHub: pirate) and has a vision of archiving everything from the internet. It is motivated by the fact that many people find interesting information on the internet that want to store locally, and most importantly, in multiple redundant common formats(HTML, PDF, PNG, WARC etc.). So, the user gives specific terminal commands including the URL that wants to archive, and then all the formats of the archived website are stored and organized in an HTML page.

While trying to build the project I faced a very basic problem. My operating system (Windows) couldn't support the project. So I came up with the solution of working from a virtual machine that runs on Linux, more precisely I worked with Oracle's VM, virtualbox.

Finally, in order to understand deeper the project I started reading the code of every file and categorize it depending on it's role. Of course, the documentation helped me a lot in this process.

2   My contribution

At first, I was trying to find something that I could contribute. Luckily, the person who ones repository, Nick, mentioned me in a issue with medium difficulty. That was very good for me because I needed to break out of my comfort
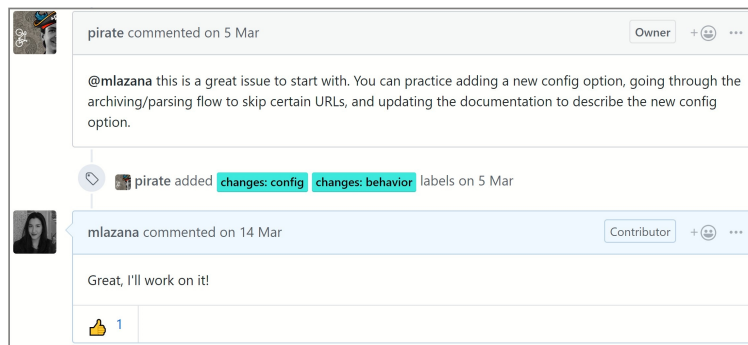

Figure 2: Mentioned in issue 200

zone and deal with a problem I didn't choose.

The issue was about creating a blacklist of URLs that shouldn't be archived. For example, when someone wanted to archive automatically his/hers bookmarks, websites that were useless, in terms of page information, were being archived. So when someone wanted to archive his bookmarks he would have also unimportant information, losing at the same time, disk space.

My first step, while dealing with this issue, was to understand which parts of the code were producing the archiving/parsing flow. So, I found the specific

files that are involved with this process and start experimenting by changing part of them so that I could understand their usage. My thought was to create a method as a filter in order to check which URLs are the proper ones.
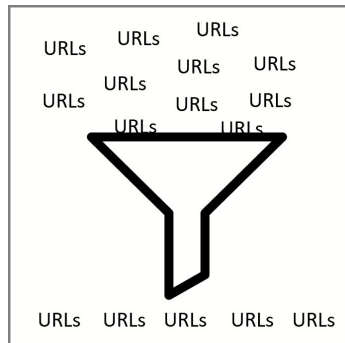
Figure 3: URL Filter

Before that I loaded an environment variable and used the method re.compile() to compile the regex.

```
URL_BLACKLIST = re.compile(
    r'(.*\.youtube\.com)|'
    r'(.*\.amazon\.com)|'
    r'(.*\.reddit\.com)',
    re.IGNORECASE,
)
```

Figure 4: Regex compilation

After that I created a method to exclude any URL that was in blacklist using generator comprehensions.

```
def exclude_blacklisted(links):
    """exclude URLs that match the blacklisted url pattern regex"""
    return (link for link in links if not URL_BLACKLIST.match(link['url']))
```

Figure 5: Method using generator comprehension

Finally, I had one more challenge. I needed to find the perfect place to call this method, as links can only get loaded in a couple of places. Adding links from a new import source, parsing an existing output/links.json index file, or reading an individual json file. So I found a method which is executed before these three methods are called.

2.1 Code quality

First of all, in the begging of my contribution process I tried to read as much code as I could in order to understand the best practises. I detected what kind of names they give to their methods and variables, but also the number of empty lines they use between their methods etc. I did that, as I wanted to contribute code that suits the project.

Further more, I wrote descriptive comments to my methods and variables.

```
links = list(exclude_blacklisted(links))  # exclude URLs that match
                                  the blacklisted url pattern regex
```

Figure 6: Regex compilation

I wrote comments in every method or variable I created, except in the ones that was very obvious what they were doing.

2.2 Continuous Integration & Testing

The project has only Docker Cloud (ci/dockercloud)[2], which allowed me to manage the project in a container without exposing it to the rest of my system. Eventually all my commits passed this check.As far as testing is concerned, code responded positively, so my code was merged to the project!

Exclude blacklisted URLs #200

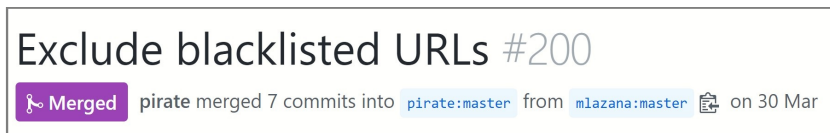**Merged**   **pirate** merged 7 commits into `pirate:master` from `mlazana:master`   on 30 Mar

Figure 7: Merge

3 Project organization

I worked with GitHub in my forked repository. I tried to make small commits at a time in order to be specific in their description. Also GitHub helped me retrieve previous versions when that was needed. Finally when I finished my work I created a pull request describing my changes and also mentiongn the issue I was hoping to close. Bellow you can see my activity in GitHub:

4 Cooperation with the team

From the first moment I had excellent cooperation with the project owner Nick Sweeting. He surely helped me and introduced me to the Web archiving community. Also, I was constantly asking for feedback and code review and he
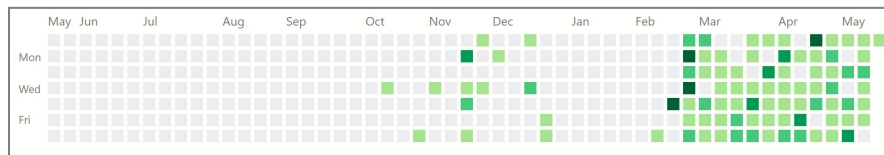
Figure 8: GitHub Activity

was always willing to do so. Further than that, we also had a video call to talk about the project and my next task. I am very satisfied with our cooperation as I learned lot of things according to project organization and python programming.
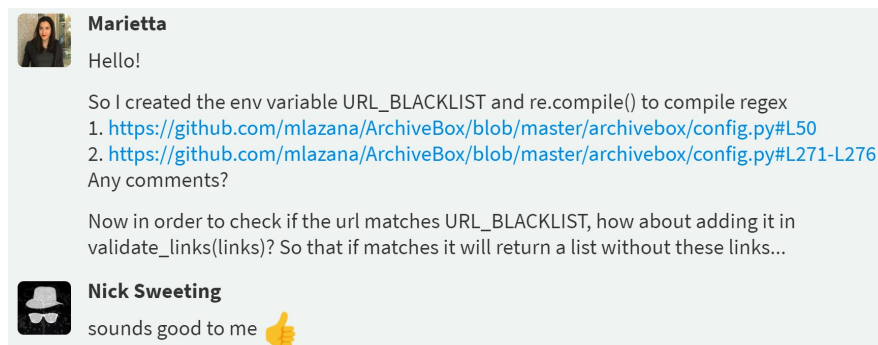

Marietta

Hello!

So I created the env variable URL_BLACKLIST and re.compile() to compile regex
1. https://github.com/mlazana/ArchiveBox/blob/master/archivebox/config.py#L50
2. https://github.com/mlazana/ArchiveBox/blob/master/archivebox/config.py#L271-L276
Any comments?

Now in order to check if the url matches URL_BLACKLIST, how about adding it in validate_links(links)? So that if matches it will return a list without these links...

Nick Sweeting
sounds good to me 👍

Figure 9: Cooperation with project owner

5 Conclusion

My first contribution was an incredible journey as I broke out of my comfort zone and faced the real world of Software engineering. I realised many of my strengths and weaknesses as far as programming is concerned. I saw that I have very analytical thought when I deal with a problem and that helped me a lot. I also realised that in order to be good at software engineering I should be more involved with open source projects because the thing that made improve was Nick's feedback. Lastly, I gained useful knowledge and I am ready to work hard to improve my skills in programming!

References

[1] https://github.com/pirate/ArchiveBox
[2]https://github.com/pirate/ArchiveBox/wiki/Docker