

Assignment 2 of CSC01

Due: 10th July @ 11:59pm

1 Introduction

In this assignment, you will use the test driven development approach and work on a web crawler. I encourage you to use the material from this assignment, over to your project as well. First, let us give credit, where credit is due. We will use the same assignment, as posted here <http://www-scf.usc.edu/~csci572/2018Spring/CrawlerExercise.pdf>. I have also posted the pdf of this assignment on Blackboard for your reference (as assignmentHandout.pdf). Ofcourse, some of the contents, will not apply to us and these items are mentioned in the handout here.

1.1 So what are you expected to do?

1. Make sure to first, run the following set of commands, to ensure that you have the contents of the starter code on your local git repo.

- On your local computer, *cd* into the directory where you first performed *git clone* for Assignment1. This is the directory that contains the *.git* file
- Now, type the following: Comments are indicated by //

```
git remote add instructor https://mcs2@bitbucket.org/mcs2/csc01summer2018master.git
git fetch instructor
```

```
//make sure you are on master branch at this point.
git merge instructor/master
```

```
//after the merge, you should have assignment2 folder on your local
//master branch.
```

- The above git commands create an alias for *instructor*, that you have read permissions (but not write permissions). However, when you commit, make sure to commit on *origin* and not *instructor*.
2. Read the handout, available on Blackboard as (*assignmentHandout.pdf*).
 3. On the starter code, make sure to edit the *pom* file, such that you have the entries for *crawler4j* and *Mockito* and *JUnit*.
 4. In the assignment handout, you are asked to run your code on all the news sites. The first column of 'USD ID ends with' in *table1*, does not apply to us. Make sure to name your three text files for each of the news site appropriately (see the submission instructions).
 5. You can safely ignore the section 6 on the assignment handout. This section does not apply to us. Your complete code must reside on the BitBucket repo by 10th July @ 11:59pm.
 6. You are expected to use the SOLID principles of software engineering <https://stackify.com/solid-design-principles/>. I will talk more about this in next lecture. However, you should have already seen these in your B07 class.
 7. You are asked to use the Test Driven Development approach when putting your code together. Make sure to work on your test cases first, i.e, put your test cases together, and then write stub methods in your *src* so that the test cases compile, but actually fail. Then start writing enough code such that the test cases pass.

8. All your unit test must reside inside the test folder. You must use **Mockito** (where appropriate in your design) and **JUnit** to test your code.
9. You are encourage to use as many branches as you see fit. However, we will take the commit on your 'master' branch at the due date for marking purposes.
10. Do not use any fast forward merge. Only use recursive merge.
11. Your main function must always reside in **App.java**. However, you can create as many new classes that you see fit for this assignment. Besides correctness, and your unit tests, we will also be marking your code for design and how well it uses the SOLID principles.
12. Your main function in **App.java**, already takes in an array of String. This array of String is basically the array of String URL that will be passed in. I will update this item in the next two days and provide more instruction on it.
13. You will first need to spend some time and understand how the **crawler4j** available here <https://github.com/yasserg/crawler4j> actually works. It will help, if you look at the examples and ReadMe. These are provided here at <https://github.com/yasserg/crawler4j/blob/master/README.md>
14. I think this example is also very useful to get started <http://madurangablogs.blogspot.com/2014/06/simple-web-crawler-with-crawler4j.html>.

2 Items to remember

- You are following the test driven development approach. Make sure to work on your test cases first, and have these committed into your master branch (with appropriate commit messages).
- Make sure to use Mockito, JUnit for testing purposes. We want to see how well you use Mockito to test your code and how well you mock your dependencies.
- Make sure to use the SOLID principles of software engineering when writing your code out.
- You are free to use any IDE for the assignment. Make sure to update your .gitignore file.
- You must use the Google Style coding available here for most IDE that you can configure: <https://github.com/google/styleguide>.