Creating a GitHub account

GitHub is the most popular social network for developers. GitHub enables developers from all over the world to share code!

GitHub uses Git, which is also a very popular tool among developers. Git enables developers to version their code in a professional way. If you are using a Linux or a Mac machine, you already have Git in your machine. If you're using Windows, just install it, following the steps on Git's official website: https://git-scm.com/download/win. If you have any trouble, post a question on our forum.

Ok, now that you have Git installed, the very first step to access the code is to create an account on GitHub. Here we will go through this process step by step. We show the screenshots of each step below:

1. Go to the GitHub website (https://www.github.com)
2. Choose a username, email and password on the homepage and click the "Sign up for GitHub" button.
3. When asked to choose your personal plan choose "Unlimited public repositories for free".
4. Leave the "Help me set up an organization next" box **un**checked.
5. Choose whether or not you want update emails from GitHub.
6. Click "Continue".
7. Fill in the small survey and click "Submit" or optionally click on "skip this step".

Congratulations, you just completed the first step towards testing the code!

Forking the code

This code is stored in its own repository. However, it would quickly become chaotic if everyone following this course could just change and add code on a whim. To prevent this from happening, you cannot make changes to a repository you are not a member of. What we can do is to make an exact copy from a repository for our personal use, a so called "fork".

Here we will show how to create this fork:

1. Log in into GitHub (you are already logged in if you just created your account).
2. Go to this repository: https://github.com/SERG-Delft/mooc-software-testing.
3. At the top right, click on **"fork"**.
4. When asked "where should we fork this repository" choose "@your_username". Now, you are telling GitHub to create a copy of our code for you, so that you change it as you want!
5. Wait a few seconds for the process to finish.

You have just forked a repository! This repository will now show up on the home page under your repositories.

Cloning the repository to your machine

Now that you have a copy of our project in your GitHub account, we will make a local copy (clone) of the online repository. This will allow us to change our local version and later push these changes to the online repository. You make a clone of a repository by following these steps:

1. Go to the repository you forked in the previous step
2. Click on the "code" button.
3. Copy the URL that is there. It looks something like *git@github.com:YOUR-USER/mooc-software-testing.git* OR *https://github.com/YOUR-USER/mooc-software-testing.git.*
4. Open your terminal (command prompt, bash, ...). If you are using Windows, you just installed the Git Bash utility.
5. Navigate to the directory where you want to save the project (using the "cd" command).
6. Type "git clone URL", where URL is the link you copied in step 3.
7. Press enter and wait for the process to finish.

You now have a clone of the project on your machine! Open your (Windows) Explorer and check out all the source code there. We are going to use all of them as examples throughout this course.

Import a project on IntelliJ or Eclipse

Now that we have our code and our editor we still need to import the project into our IDE. To do this, follow the next steps, of which some are visualized below the text:

IntelliJ

1. Open IntelliJ.
2. Choose *Import Project.*
3. Navigate to the folder where you saved the source code and click *ok.*
4. Choose "Import project from external model", select "Maven" and click *next.*
5. Click *next* again (or change anything you wish to change).
6. Click *next* again.
7. Choose your latest JDK and click *next.* If there is no SDK, add it first by clicking on the "+" and navigating to your JDK directory (if you do not have a JDK yet you can download it from https://oracle.com.
8. Give your project a name, choose a location and click on *finish.*

You have now imported the source code in IntelliJ, feel free to have a look around at how the project is structured. You should be able to find a *main* and a *test* folder under *src*, these folders contain the code and the tests.

Eclipse

1. File -> Import…
2. Git -> Project from Git
3. Existing local repository
4. Add
5. Browse… Navigate to and select the directory “mooc-software-testing” that you just cloned from GitHub
6. Click the checkbox next to the directory. Click Next.
7. Import as General Project
8. Navigate to mooc-software-testing on the Package Explorer window pane
9. Right click on it, and choose Configure -> Convert to Maven Project

Writing our very first automated test

Go to the GettingStarted class. We see that it contains a single function. This function gets an number (integer), adds 5 to it and then returns the new value. We can create a test class to see if this function behaves like we expect it to. For this example, the test class is already created.

There are already three test methods in this class. However, only the first one is complete.

```
@Test
public void addFiveToTwenty() {
    int result = new GettingStarted().addFive(20);
    Assertions.assertEquals(25,result);
}
```

We see that:

- The very first line @Test, indicates that the following method is a JUnit test. JUnit is the framework we are going to dive into later.
- The method name already explains what we want to test.
- The body of this test method then invokes the function we want to test (addFive()), passing the number 20 to it.
- Given this input, the expected output should then be 25. so we assert that our result is equal to 25 (using the Assertions.assertEquals() method). If the result is not equal to 25, we expect this test to fail.

You can now run this test case and see what happens. Right-click the GettingStarted test or project, and Run Junit test case.

Next, complete the remaining test cases. Uncomment them, and try to fill in the blanks.

Add a few more test cases and make sure that they run properly.

Additional resources on how to use all of the functionality of JUnit: https://junit.org/junit5/docs/current/user-guide/

Pushing your changes to the online repository

The final step is pushing your local changes to the online GitHub repository. For this, we will need the git bash again (the same one we used when cloning the repository). Follow these steps to push your changes to GitHub.

1. First, we navigate to the folder in which the project is saved (using the *cd* command).
2. Then we need to add these changes to the system using the command "git add .". This instruction tells Git that we want to add all the changes to the commit.
3. We need to commit these changes, we can do this by typing *git commit -m "Complete the test class for addFive"*
4. Finally, we need to push our changes to the online repository by typing *"git push origin master"*.
5. The terminal will now ask for your GitHub username and password. Fill them in and press *Enter.*
6. You can now visit your GitHub repository (https://www.github.com/YOUR-USER/mooc-software-testing) and see that your changes are there.