

Git Instructions

2016

Contents

1	Fresh start	3
1.1	Install git and TortoiseGit	3
1.2	Github account	3
1.3	Main repository	3
1.4	Fork repository	4
1.5	Personal repository	5
1.6	Clone personal repository	6
2	Personal repository setup	9
2.1	Correction	9
2.2	Setup remotes	9
3	TortoiseGit > Pull from origin	14
4	TortoiseGit > Push to personal	16
5	Workspace cleanup	22

1 Fresh start

If you didn't do any Github related stuff before this point I would recommend you following the steps described in this section. Also, if you would like to do a fresh start even if you already used Github to send homeworks you can follow this but, of course, there would be some steps that you already did and won't require you to do them again (**you can delete the two folders related to our Github repositories you have on your PC at this point BUT save any work you have done before**).

1.1 Install git and TortoiseGit

Follow this YouTube video in order to install the required tools.

1.2 Github account

Go to Github website and create an account there.

1.3 Main repository

Our main repository from where you will create a **fork** is located at either <https://github.com/OOPCLASS2016/30222>
or
<https://github.com/OOPCLASS2016/30425>
depending in which group you belong to.

1.4 Fork repository

Go to the URL of our main repository as mentioned in subsection 1.3. As show in figure 1 on that URL you should see the name of our organization *OOPCLASS2016* on top left corner of the page. Click on **Fork** button from the top right corner of the page as displayed in figure 1.

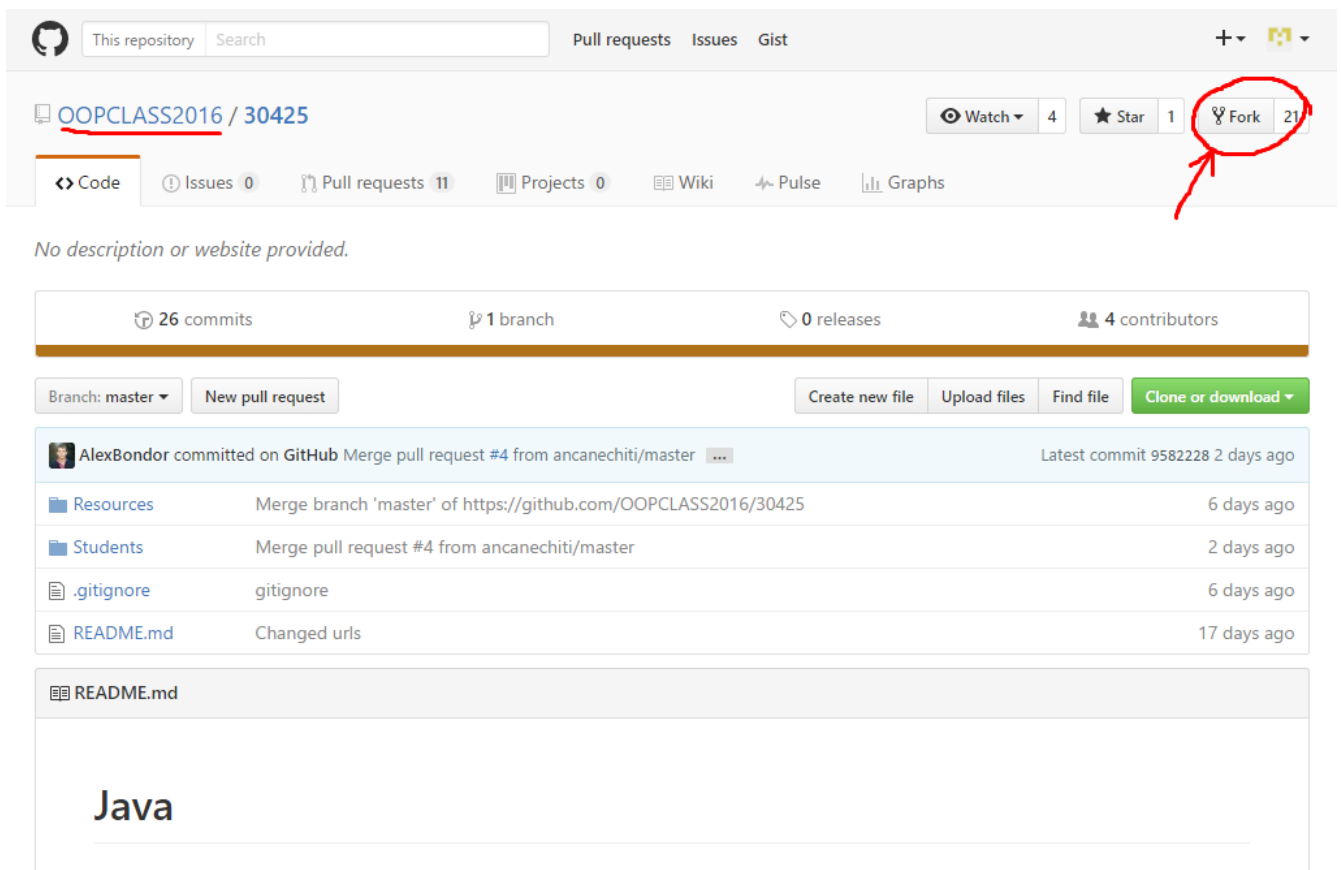


Figure 1: Forking main repository

1.5 Personal repository

Once you have completed the step described in subsection 1.4 you should be redirected to the page of your personal repository. This is a copy of the main repository and you will be working with this from now on. Figure 2 displays exactly the changes on the page that you should look after. Note that instead of the name of our organization, your Github username will be displayed in the top left corner of the page (in my case, *thefunkyjar*). At this point you can be sure you have successfully forked the main repository.

The screenshot shows a GitHub repository page for a user named 'thefunkyjar' with the repository name '30425'. The repository is forked from 'OOPCLASS2016/30425'. The page includes a navigation bar with links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are statistics for 'Watch' (0), 'Star' (0), and 'Fork' (22). The main content area shows the repository's structure, including a list of files and folders: 'Resources', 'Students', '.gitignore', and 'README.md'. The 'README.md' file is selected, showing its content. The content includes a section titled 'Java' and a list of bullet points under the heading 'What to expect'.

thefunkyjar / 30425
forked from OOPCLASS2016/30425

Watch 0 Star 0 Fork 22

Code Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

26 commits 1 branch 0 releases 4 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is even with OOPCLASS2016:master. Pull request Compare

AlexBondor committed on GitHub Merge pull request #4 from ancanechiti/master Latest commit 9582228 2 days ago

File	Commit Message	Time Ago
Resources	Merge branch 'master' of https://github.com/OOPCLASS2016/30425	6 days ago
Students	Merge pull request #4 from ancanechiti/master	2 days ago
.gitignore	gitignore	6 days ago
README.md	Changed urls	17 days ago

README.md

Java

What to expect

- A good progression scheme through the basic concepts of OOP (in Java)
- A good chunk of exercises and assignments which should offer (if properly done) good insight on new OOP concepts and Java's specific innards.
- After completing this course, you should be able to *think* in the (for some) new OOP paradigm with ease and you will be able to model a problem according to these knew principles and concepts

What not to expect

• A bold, new, and exciting framework of interaction. The purpose is not to reproduce an elementary school type class

Figure 2: Personal repository

1.6 Clone personal repository

In this next step you will create a clone of your personal repository on your own PC. From the page shown in figure 2 from subsection 1.5 click the green button which says **Clone or download**. A small box will pop-up as show in figure 3. Please make sure that at this point the text underlined in figure 3 is exactly your Github username! Copy the URL as shown in the mentioned figure.

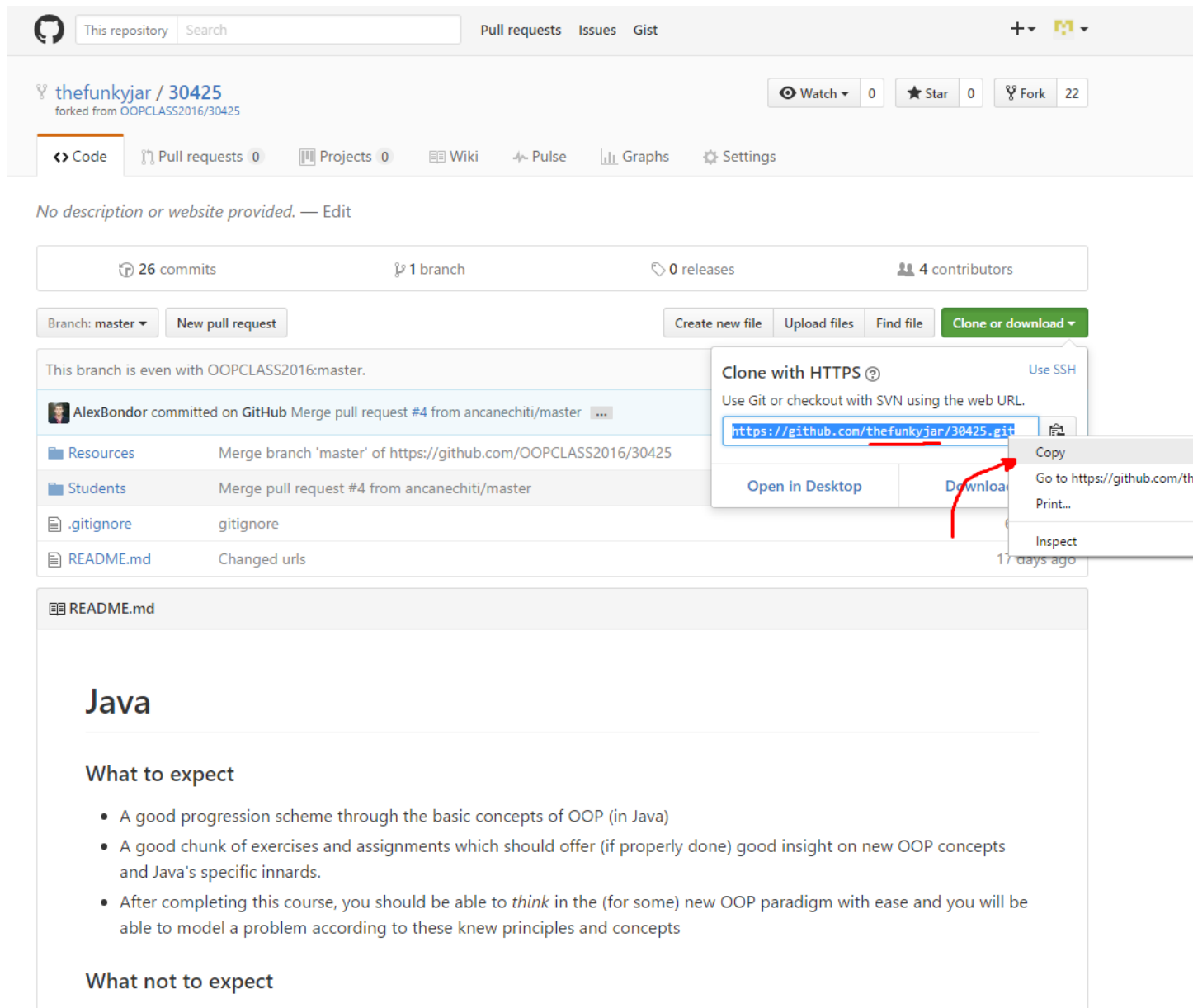


Figure 3: Clone repository

On your PC find a place where you would like to save the personal repository. The folder **30425** or **30222** will be created at this location once you follow the next step. Navigate to the folder you have just created or selected as a destination to save your personal repository. Right-click inside this folder and, as shown in figure 4 click on **Git Clone...**

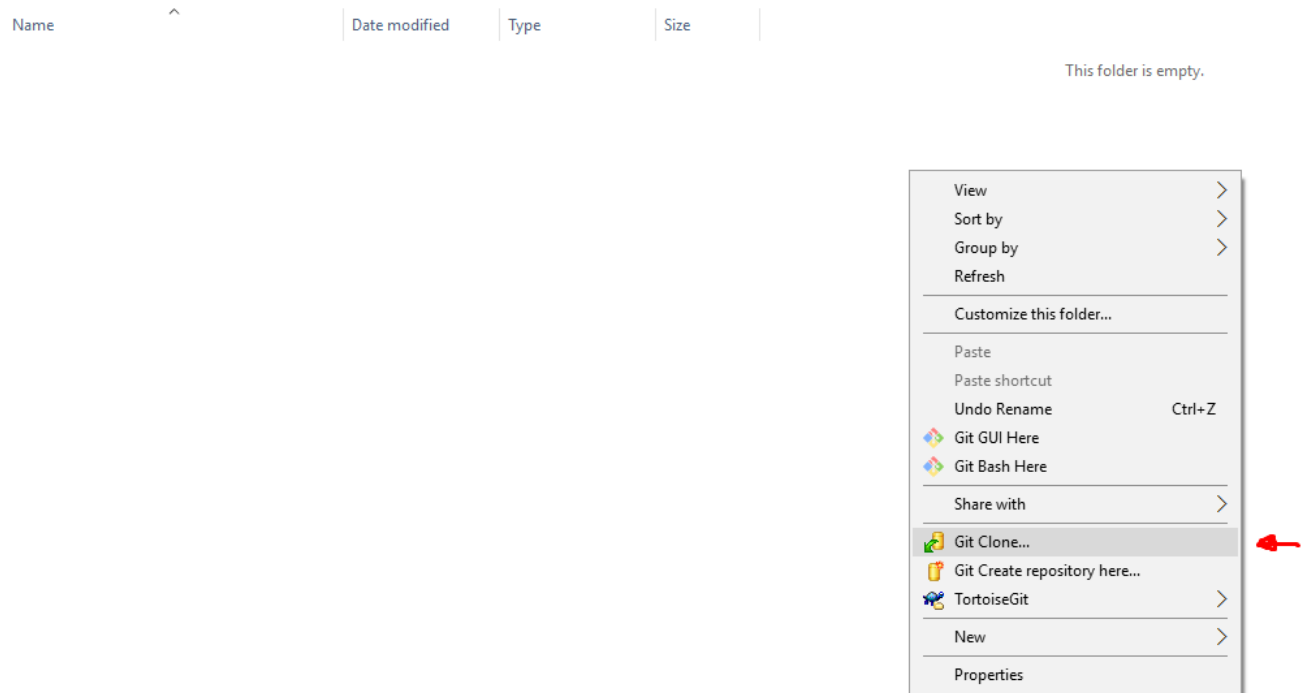


Figure 4: Git Clone..

Once you click **Git Clone..** the message box displayed in figure 5 will pop-up. Paste the URL you have copied at previous steps in the URL section and maybe check the destination folder in Directory section. Hit **OK**.

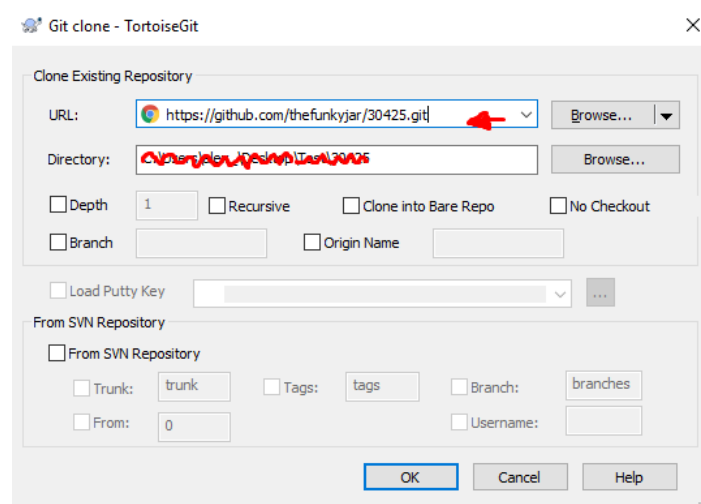


Figure 5: Git Clone.. message box

Once you have completed these steps you should see that a copy of your personal repository has been created. This is going to be the **ONLY** folder in which you will have to work and receive assignments. The folder I am talking about is the one displayed in figure 6. However, the name of the folder could be **30222** if you are in the other group.


Name	Date modified	Type	Size
 <u>30425</u>	10/22/2016 2:05 PM	File folder	

Figure 6: PC version of your personal repository

At this point you have successfully completed the initial steps for being able to work with Github for the assignments you will receive at the OOP labs. Please go to the next sections in order to configure and understand how to work with what you have just created.

2 Personal repository setup

2.1 Correction

If you did a fresh start skip this correction.

If you followed the YouTube videos which showed you how to work with Github then you need to perform some minor correction as we are going to approach things a bit differently. You were asked to clone both repositories: your personal repository and also the main one.

At this point on your PC you have two folders: one containing the changes from the **OOP-CLASS2016** repository and the one in which you work, **<your_github_username>** repository. **I would ask you to DELETE the clone of the MAIN repository from your PC.**

2.2 Setup remotes

In this step we will do the necessary settings so that you will be able to get new changes from the main repository (origin) and put your changes (e.g. assignments) to your own repository (personal). On your PC navigate inside the repository folder (taking into account you have performed the previous step, you must only have the clone of your personal repository on your PC). As shown in figure 7 right-click inside this folder and go to **TortoiseGit > Settings**. (You might want to check your folder's contents by comparing them to the ones you see in figure 7).

Name	Date modified	Type	Size
Resources	10/22/2016 2:05 PM	File folder	
Students	10/22/2016 2:05 PM	File folder	
README.md	10/22/2016 2:05 PM	Text Document	1 KB
	10/22/2016 2:05 PM	MD File	3 KB

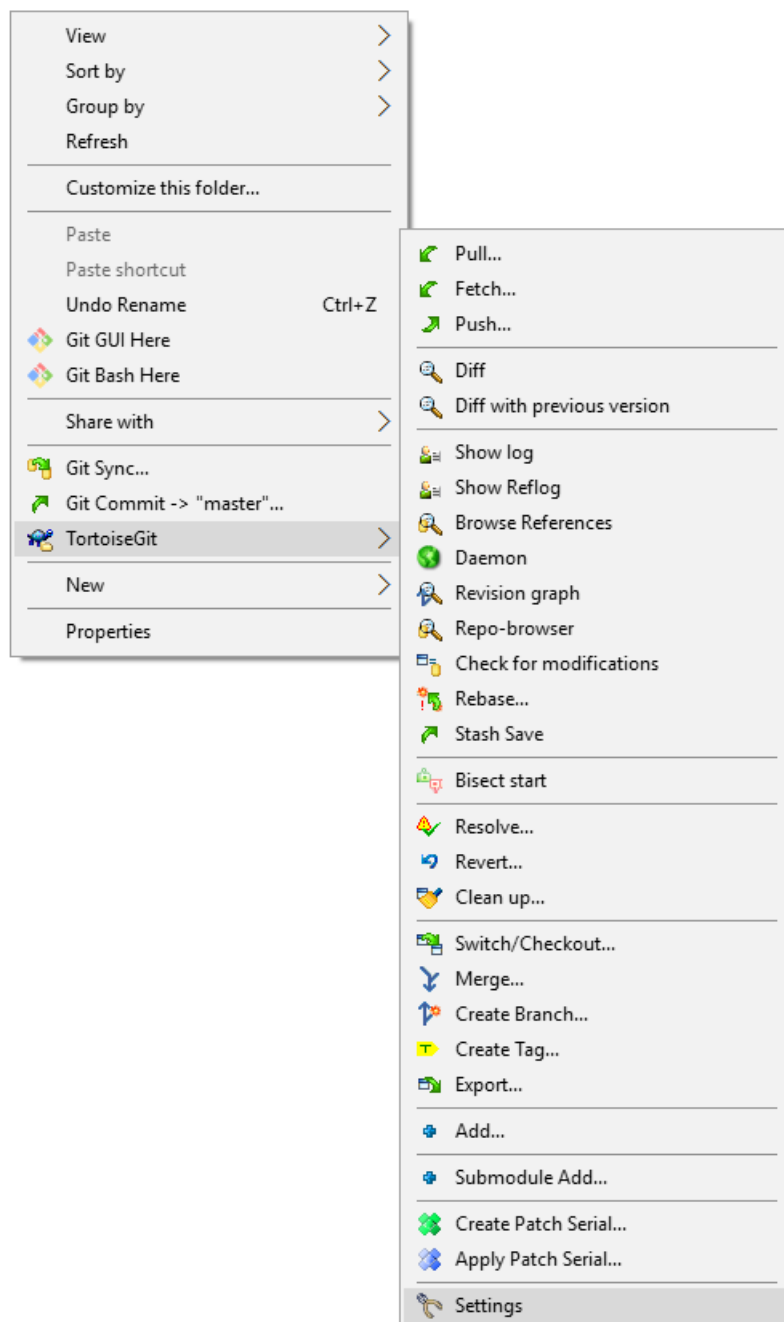


Figure 7: TortoiseGit > Settings

In the windows that pops-up click on **Remote** from **Git** as displayed in figure 8. Under *Remote:* you should only see **origin**. If you check the URL written there, it will contain your Github account name (in my case, *thefunkyjar*).

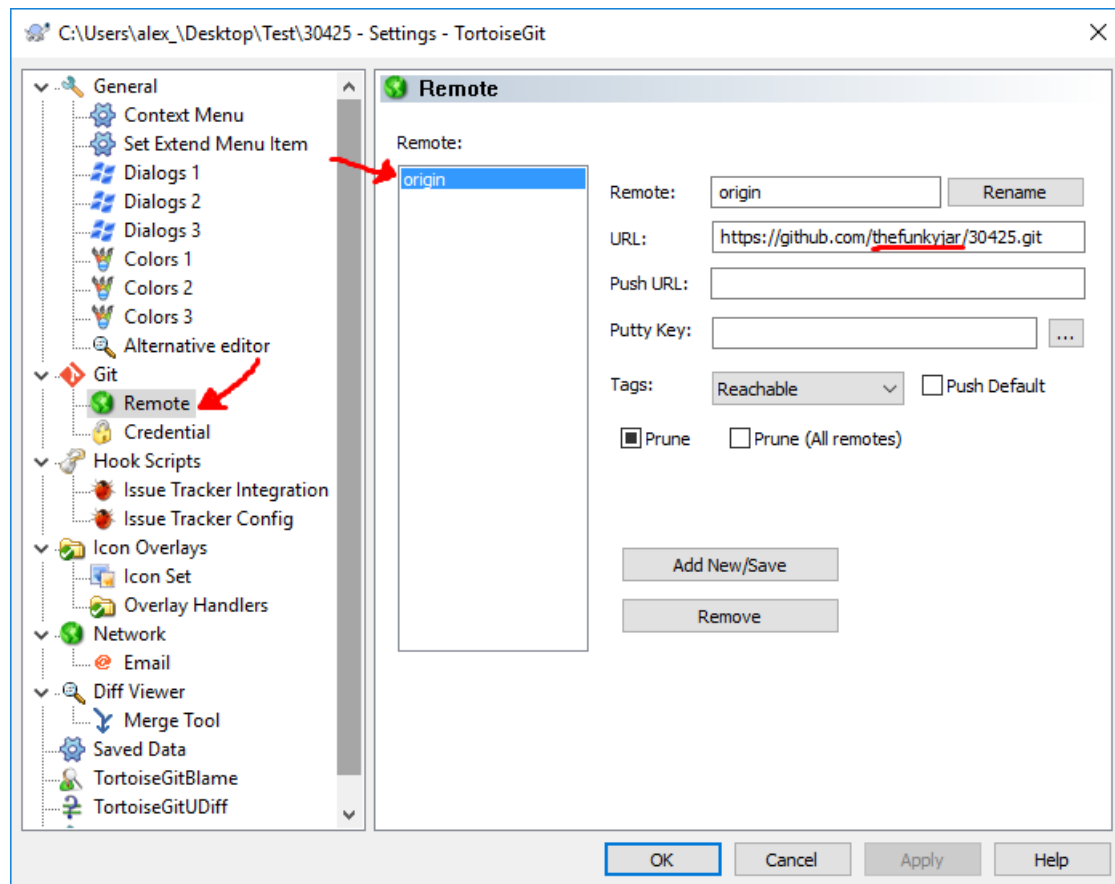


Figure 8: Settings

Rename this setting by changing its name from **origin** to **personal** as shown in figure 9. Then click the button **Add New/Save**, click **Yes**, click **Yes**, click **OK** and **Close**.

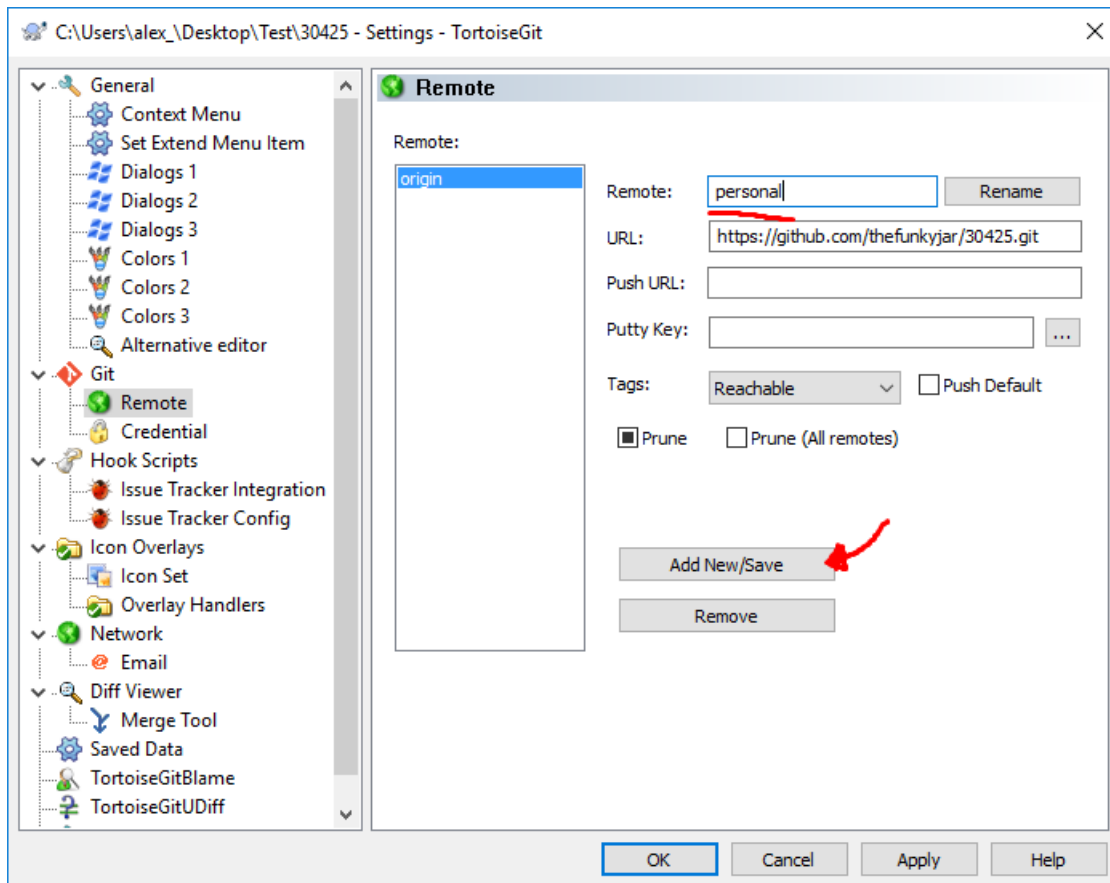


Figure 9: Create new remote

Once the new setting was created you should see both of them under the **Remote:** text. Click on **origin** again but now make sure you replace your Github account name with **OOP-CLASS2016** in the URL field as shown in figure 10. Again, click **Add New/Save** and then **Overwrite**. Close this window by clicking **OK** next.

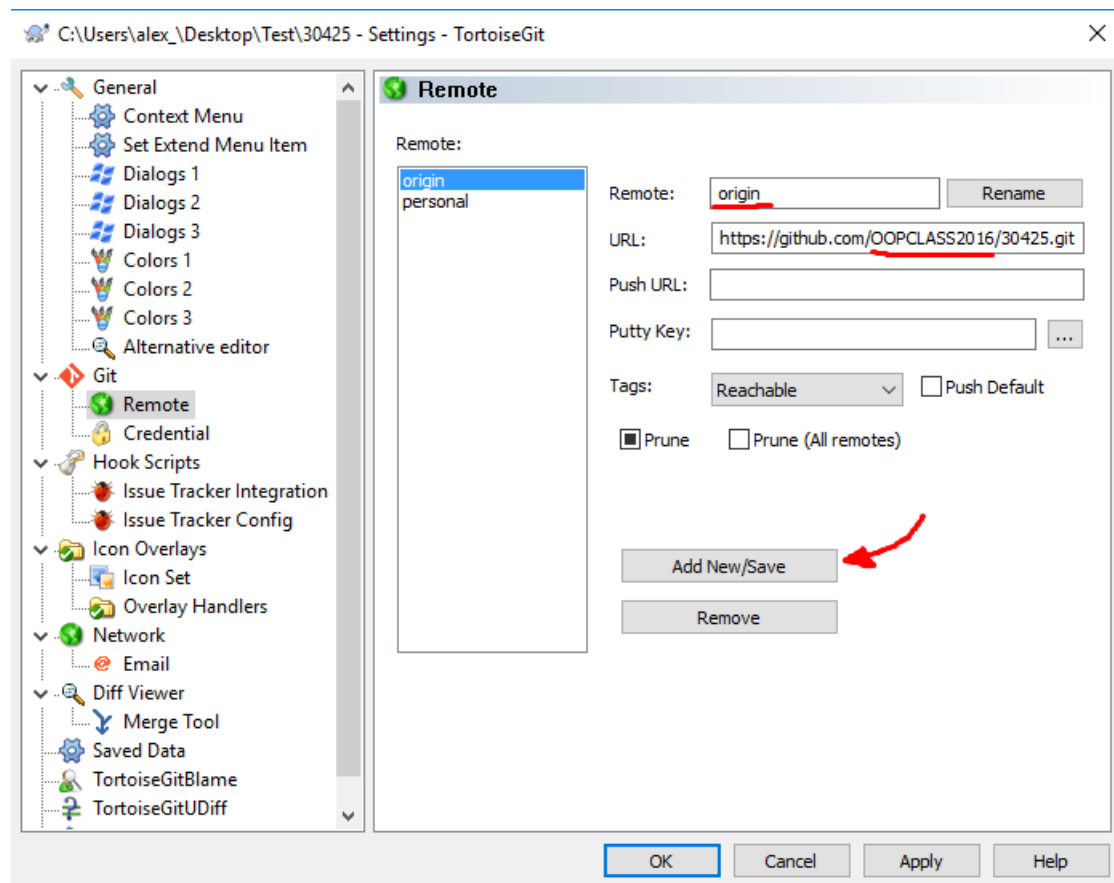


Figure 10: Replace Github account name with organization name

3 TortoiseGit > Pull from **origin**

Every time I will add some assignments as I did for homework 2 you will need to update the folder you have on your PC. In order to do this you will have to **PULL** the changes every time you want it to be up-to-date. In order to do so, you will have to go to inside the repository on your PC. As shown in figure 11 right-click and go to **TortoiseGit > Pull...**

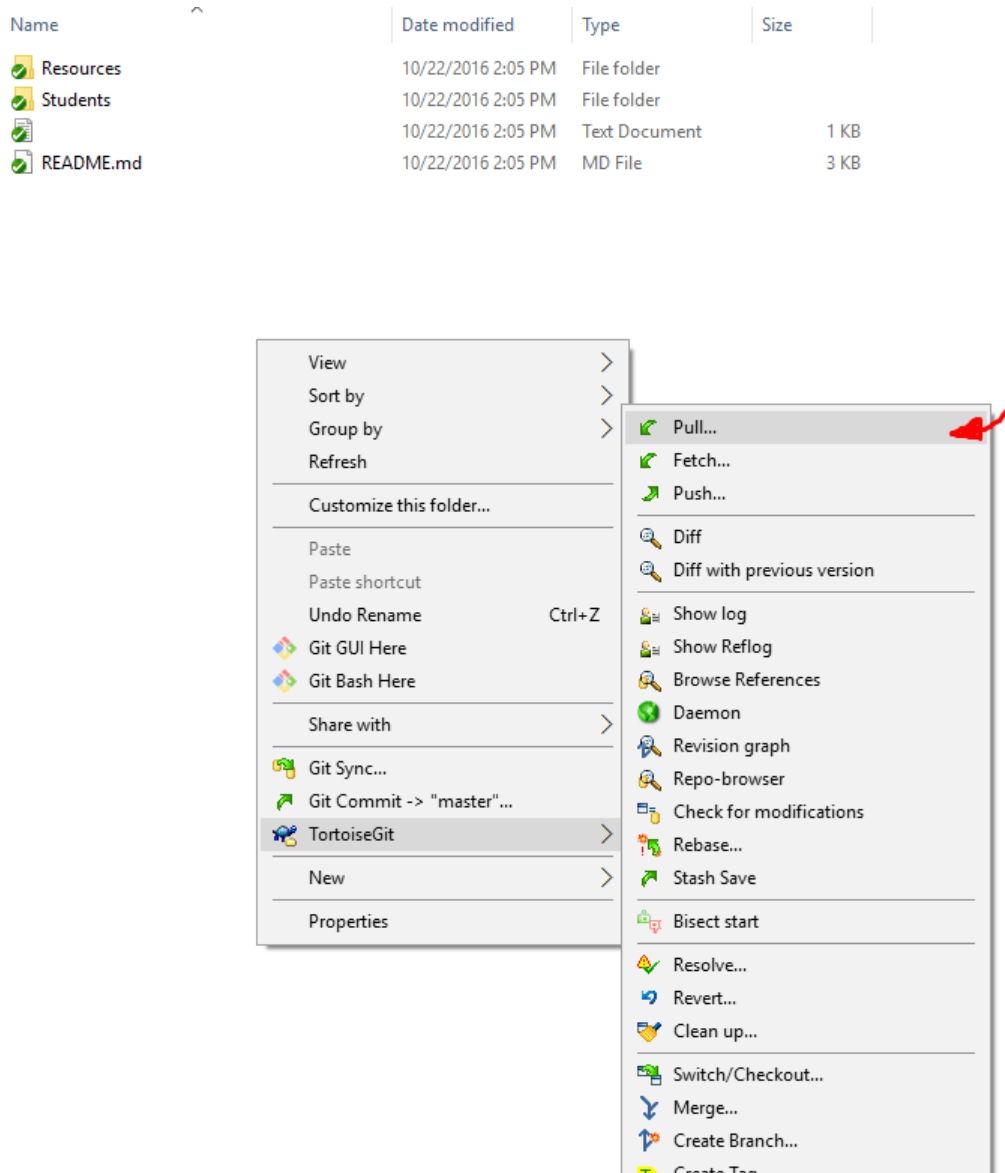


Figure 11: TortoiseGit > Pull..

In order to complete the **PULL** routine you have just started make sure that in the window that just popped-up you select to **PULL** from **origin** as shown in figure 12. Hit **OK** and if everything went alright you should see a blue text which contains *Success ...*

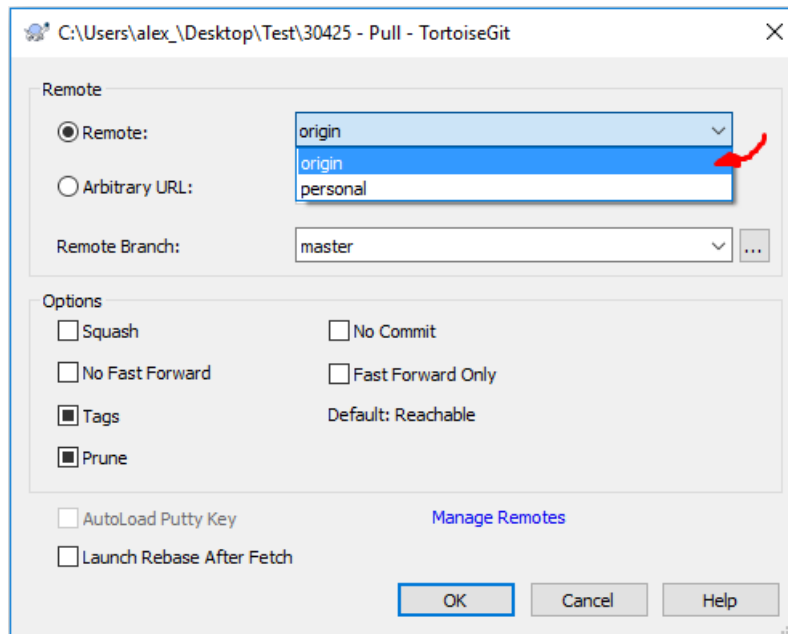


Figure 12: Select **origin** when doing a pull

4 TortoiseGit > Push to **personal**

Every time you will do some changes (e.g. writing the code for an assignment) you will have to **PUSH** those changes to your **personal** online repository. Let's say you have just completed one assignment and you want it to be sent so that I can give you the feedback you need. Assuming that you added the code inside **30425/30222 > Students > Your Name** you will be able to perform the next steps (You can follow section 5 which explains how to setup your workspace from Eclipse in the folder I have just mentioned so that everything you write in Eclipse will be added directly to that folder).

First thing you will have to do is to go to your repository (30425/30222) on your PC right-click and **Git Commit -> "master"...** as shown in figure 13.

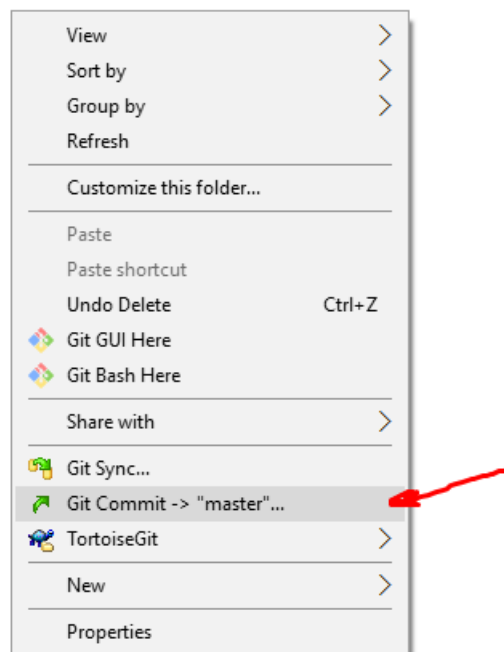


Figure 13: Git Commit -> "master"...

Once you performed the previous step you will be prompted to enter the commit message. The commit message should always be a sentence or more that describe what you did exactly (e.g "Finished my first assignment"). The window where you ahce to enter your commit message is presented in figure 14. Enter your message and click **Commit**.

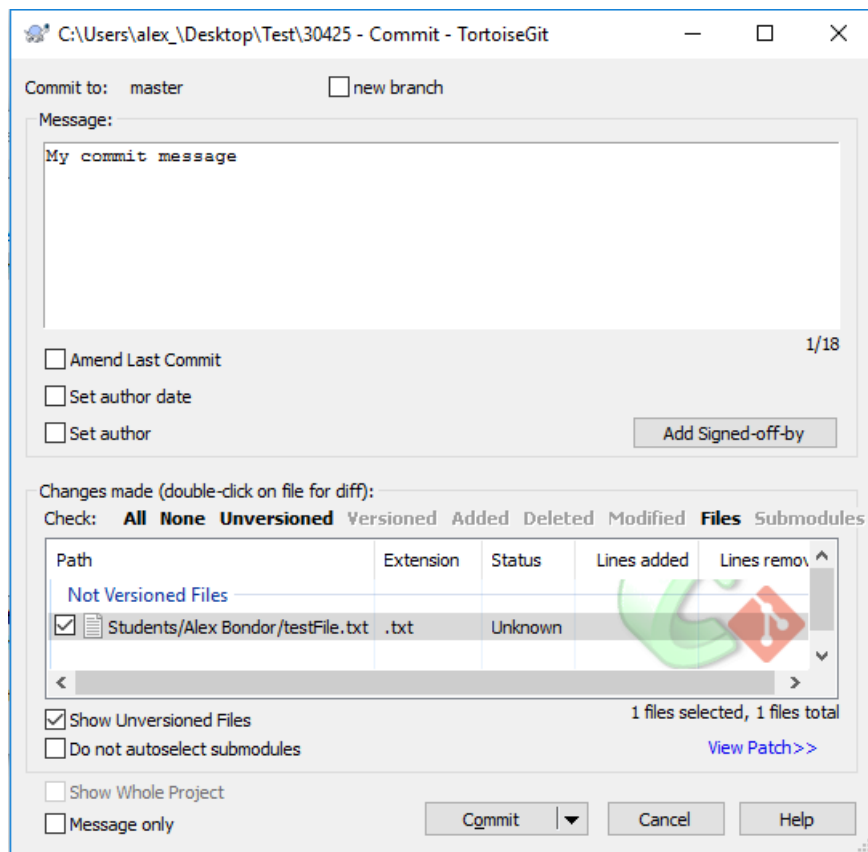





Figure 14: Commit message

In the next step you will have to **PUSH** your changes to your **personal** online repository. Figure 15 shows that you will need to right-click inside your repository and got to **TortoiseGit > Push...**

Name	Date modified	Type	Size
 Resources	10/22/2016 2:05 PM	File folder	
 Students	10/22/2016 2:13 PM	File folder	
 README.md	10/22/2016 2:05 PM	Text Document	1 KB
	10/22/2016 2:05 PM	MD File	3 KB

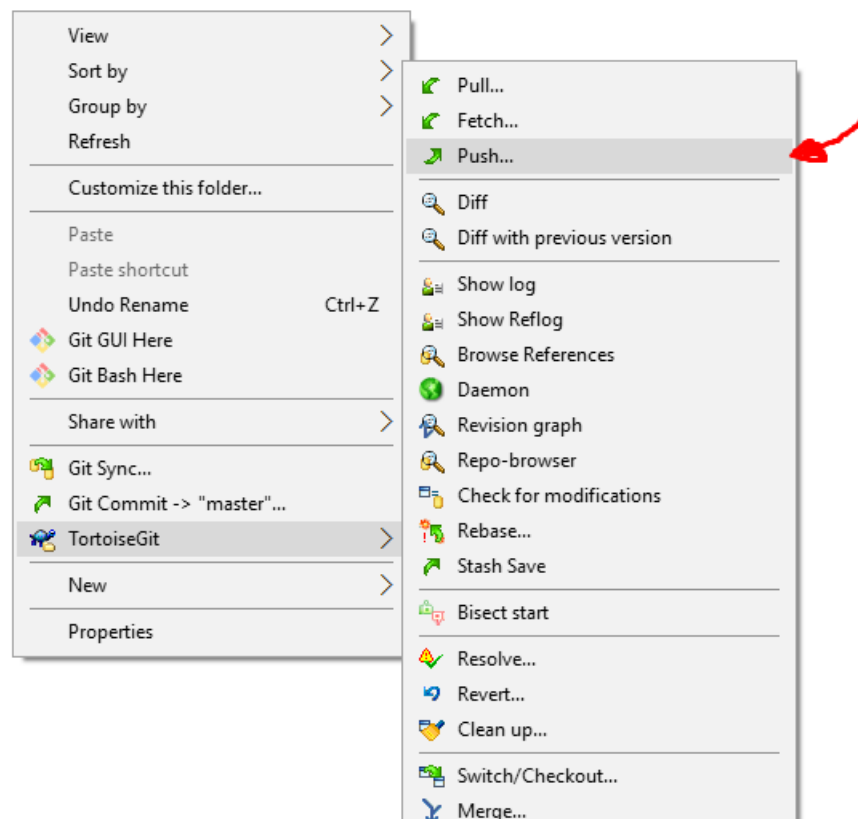


Figure 15: TortoiseGit > Push...

Before concluding the **PUSH** routine you have just started make sure that in the window that just popped-up you select to **PUSH** to **personal** as shown in figure 16. Hit **OK** and if everything went alright you should see a blue text which contains *Success...*

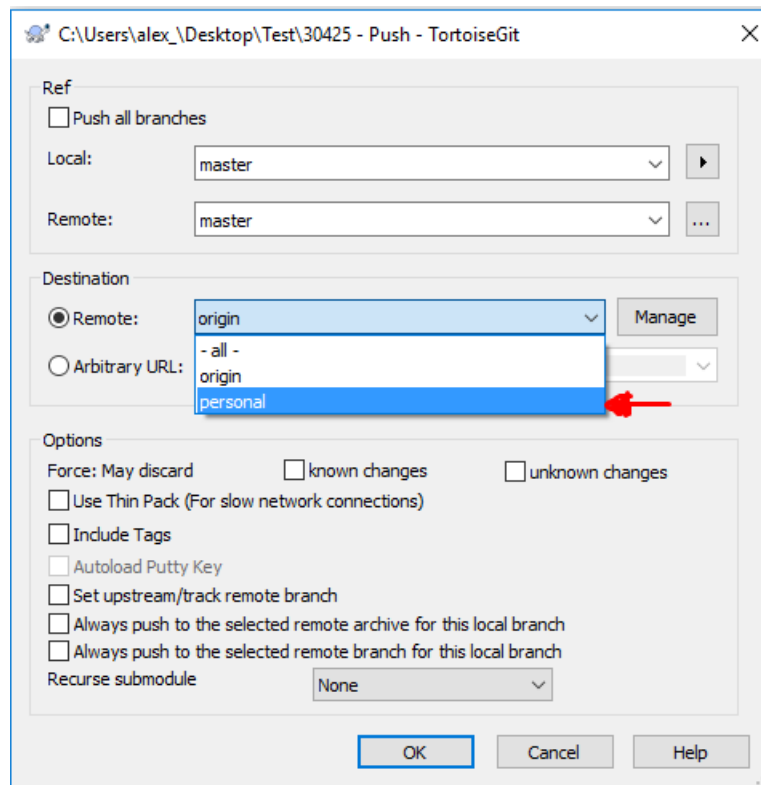


Figure 16: Select **personal** when doing a push

For now, the changes that were just pushed are only available on your personal online repository. In order to make them available for me to review them you will need to **always** conclude with a **pull request**. In order to create a new pull request you will have to navigate from your browser to the URL of your personal repository and go to the **Pull requests** tab as shown in figure 17. Make sure that it is your own repository and not the main one (OOPCLASS2016).

The screenshot shows the GitHub interface for a repository named 'thefunkyjar / 30425', which is a fork of 'OOPCLASS2016/30425'. The repository has 0 watches, 0 stars, and 22 forks. The 'Pull requests' tab is highlighted with a red circle in the navigation bar. Below the navigation bar, there is a section for repository statistics: 26 commits, 1 branch, 0 releases, and 4 contributors. A 'New pull request' button is visible. The main content area shows a commit by AlexBondor that merged pull request #4 from 'ancanechiti/master'. Below the commit list, the 'README.md' file is displayed, containing the following text:

Java

What to expect

- A good progression scheme through the basic concepts of OOP (in Java)
- A good chunk of exercises and assignments which should offer (if properly done) good insight on new OOP concepts and Java's specific innards.
- After completing this course, you should be able to *think* in the (for some) new OOP paradigm with ease and you will be able to model a problem according to these knew principles and concepts

What not to expect

Figure 17: Initiate a pull request

From the **Pull requests** tab it is straight forward what you will have to do. Click the green button **New pull request**, enter a message there and click again **Create pull request**. If you don't see the **New pull request** button don't worry; your new changes have been added to your previous pull request if I didn't accept it already.

If you want to make sure that your pull request was successfully created, you can navigate in your browser to the main repository and in the **Pull requests** tab you can search for the pull request you have just created. If it is there, then I got it and I can check it as well.

5 Workspace cleanup

Please do not attempt to do this section if you didn't read the previous ones.

In a previous email I have told you something went wrong with the **.gitignore** file. Meanwhile I did change it, however, you will need to perform some cleaning as well in order for your files to be committed right.

First step I would like you to perform is to empty your own folder from your personal repository on your PC. Save the folders that were created by you (ignore **.metadata** and any other folder that Eclipse created) in a different location on your PC. With your folder empty **commit** your changes. Follow the steps from section 4 but **DO NOT push and create pull request yet**, just **commit**.

Now make sure your personal repository from your PC is up-to-date by following the steps explained in section 3.

Start Eclipse and, when asked about Workspace, hand in the path to your **OWN** folder inside your personal repository (e.g. 30425/30222 > Students > Your Name) as shown in figure 18. If you are not asked to select the workspace, then you will have to search online how to change the workspace once you start Eclipse.

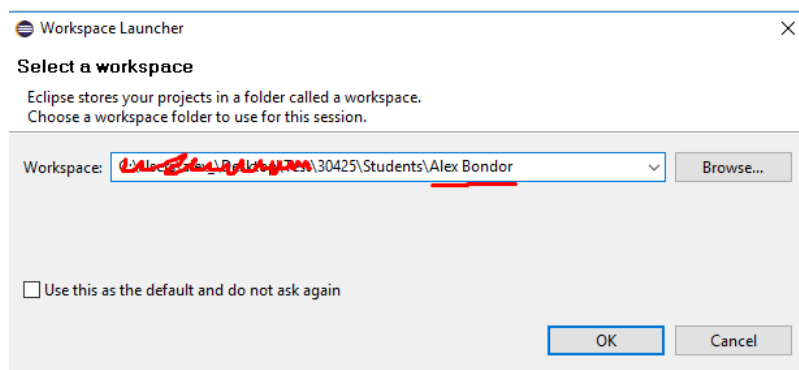


Figure 18: Eclipse workspace

At this point you will have a clean starting point and you can freely create projects or import the ones you saved when I asked you to empty your folder at the beginning of this section. Remark: You will have to create a project first before being able to import others.. (some Eclipse bug).

After you did this you can follow all the steps (commit + push + pull request) from section 4 in order to push your new clean workspace online.