F



|  |  |
| --- | --- |
| **Ezequiel Mora Martinez** - [ezequiel.mora@globant.com](mailto:ezequiel.mora@globant.com) | |
| **Reviewers** | Danny Everlyn Gomez Acevedo - [danny.gomez@globant.com](mailto:danny.gomez@globant.com)  Michele Albarenga - [michele.alvarenga@globant.com](mailto:michele.alvarenga@globant.com)  Thomas Ajjouri - [thomas.ajjouri@globant.com](mailto:thomas.ajjouri@globant.com) |

**Contents**

**Back-end installation** 3

**Docker back-end installation** 6

**Front-end installation** 9

**Docker front-end installation** 11

**Extra module - Testing the environment** 13

**Create Pull Request, Squash and merge** 14

Talent Surfer installation and configuration for local deployment

Summary

The purpose of this document is to explain and describe how to install and configure the Talent Surfer back-end and front-end for local deployment and testing.

To access Github for the first time you will need to type your Globant username in the following format “name-surname” using your corporate password. After the first time you will need to change the username to the following format “name.surname”.

***Git and Docker commands exceed the scope of this documentation, for more information on how to use them we recommend reading the official documentation.***

NOTE: To access the Talent Surfer repositories you will need permissions to enter, write and read the Ernst & Young repository group, these need to be requested and approved by the Project Manager.

# Back-end installation

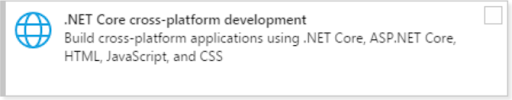
You can run both the front-end and back-end in two different environments. You can choose to use the IIS option or the docker option. We encourage you to try and get used to the docker method since this will be the alternative used for cloud deployment and continuous integration.

First of all, some prerequisites exist before starting the normal installation and configuration. We will try to guide you through all the stages of the process.

1. Git Bash: You need to download and install the Git Bash software from <https://git-scm.com/downloads> - Afterwards you can go on with the default installation, there are no other requirements.
2. Node.js: Use the official Node MSI installer to install Node and NPM.

<https://nodejs.org/dist/v12.14.1/node-v12.14.1-x64.msi>

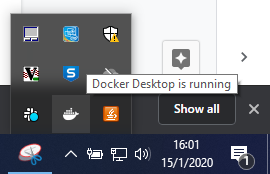
1. Visual Studio Community or Professional for Windows: We recommend that you install the Visual Studio 2019 community version. This version can be downloaded from <https://visualstudio.microsoft.com/es/vs/community/>. During the installation it is very important that you check the .NET Core cross platform development option, since this is necessary to build and run the project solution.



1. NET Core 2.2 SDK: It’s important that you download and install the .NET Core SDK version for Visual Studio 2019 (Similarly if you installed the Visual Studio 2017 version you will need the 2017 SDK Version) - <https://dotnet.microsoft.com/download/dotnet-core/2.2>.

<https://dotnet.microsoft.com/download/dotnet-core/thank-you/sdk-2.2.207-windows-x64-installer>

1. Docker for Windows: It's mandatory to make the Docker installation after the Visual Studio installation, because this software will be integrated with the Visual Studio compilation pipeline. <https://hub.docker.com/?overlay=onboarding> - On this page you need to register a Docker hub account in order to download the software. Run the default installation and after the first Docker run do a secondary click on the Docker system tray icon and in “Settings” share the C:/ Drive.



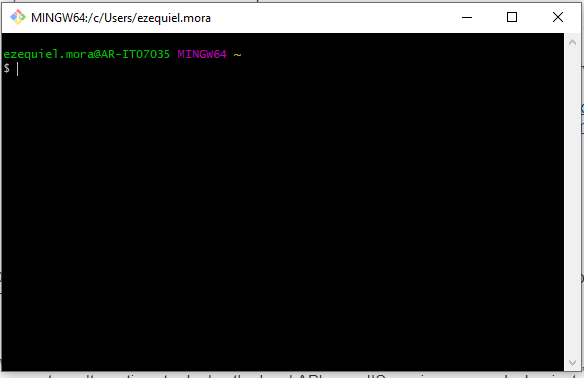
1. Additionally if you run the IIS instance of the project you will need to install SQL Server Express and SQL Server Management Studio:
2. <https://www.microsoft.com/es-es/sql-server/sql-server-editions-express>
3. <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

If you run the Docker instance version of the API this step is optional.

NOTE: Maybe you will require administrative permission to install and run this software hence in some cases it will be necessary to raise a ticket to “Service Desk”.

Now we will enumerate the steps necessary to build and run the local version of the API. As we said there are two alternatives to deploy the local API, as a IIS service or as a Docker instance.

1. Clone the Git repository for the Talent Surfer back-end
   1. Go to the Github Project dashboard <https://github.corp.globant.com/ErnstYoungX/TalentSurferBE> and press the “clone” button to get the git link for clone
   2. Open the Git Bash console, for instance from the Windows Start Button.



* 1. In the Git Bash console now clone the repository “develop” branch; since this is the development branch for deployment

|  |
| --- |
| ***git clone -b develop https://github.corp.globant.com/ErnstYoungX/TalentSurferBE.git*** |

* 1. The Git Bash will ask you to enter your Globant username and password.

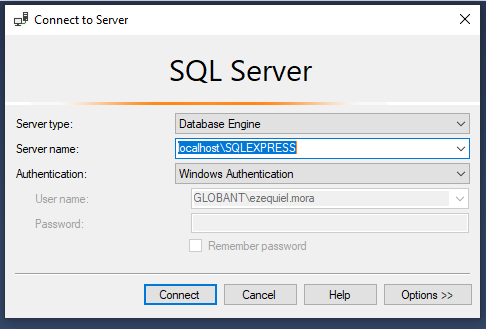
IMPORTANT NOTE: In this tutorial we cloned the HTTPS version of the repository but If you have Github configured with SSH credentials you need to clone the SSH \*.git remote for SSH instead. This link is available from the Github project dashboard “clone” button.

I) IIS Service version

To run the API as an IIS service we need have previously installed SQL Server Express and SQL Server Management Studio

1. Enter the recently cloned repository and open the \*.sln solution file in Visual Studio 2019 and wait for all dependencies to be loaded.
2. Wait until the solution loads all the internal projects, after that go to the menu bar and press the button “Build” and then select “Build Solution” from the drop down menu.
3. In Visual Studio, after all the projects in the solution are loaded and built, press secondary click over the “EY.TalentSurfer.API” and select “Set as Startup project”.
4. From the build drop down list select the IIS Express and press PLAY.
5. This step will automatically create the DB into local SQL Server and populate the tables.

If you wish to access the DB and make queries you need to open SQL Server Management Studio and use the following credentials



# Docker - Back-end Installation

II) Docker container version

For this alternative you need to have Docker hub installed and running previously and it will be the default deployment standard, so we encourage you to try and use this option. If you need to make changes to the Docker compilation pipeline, it's necessary to edit the dockerfile of the project and the docker-compose.yml of the solution.

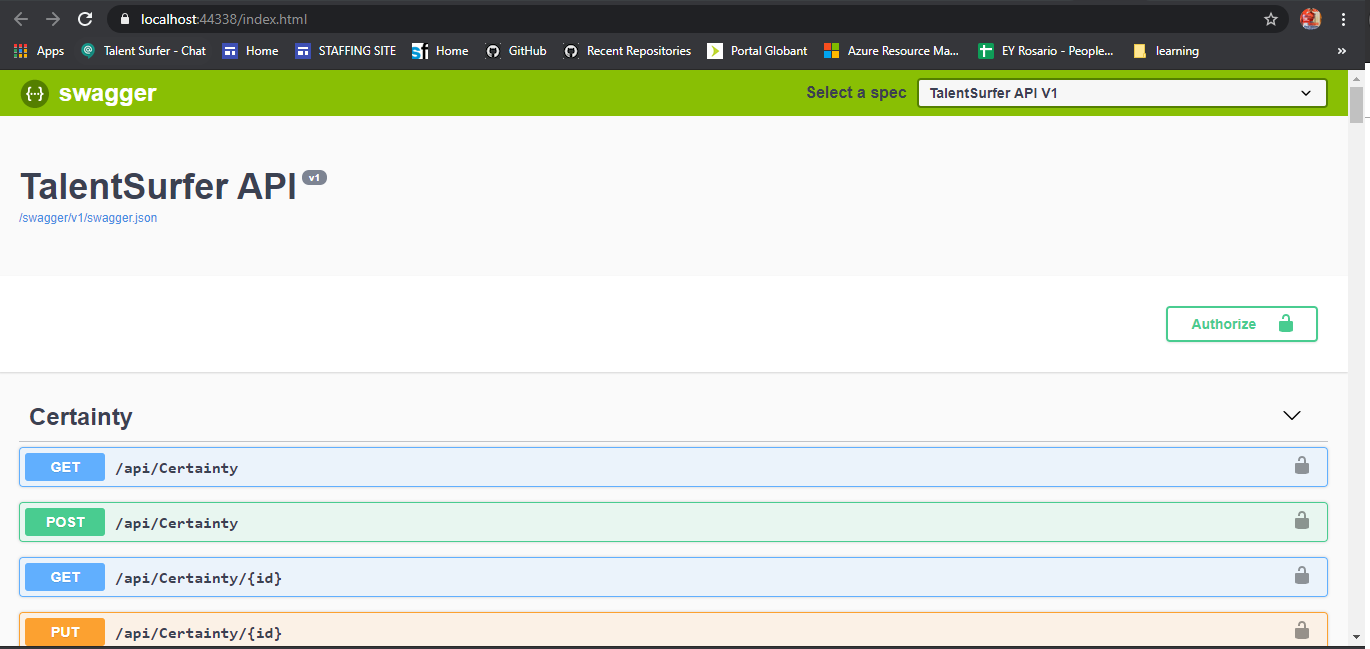
1. Hover the mouse over the Docker compose project and with a secondary click select “Setup as Startup Project”.



1. From the build list select “Docker Compose” and Press “Play”.



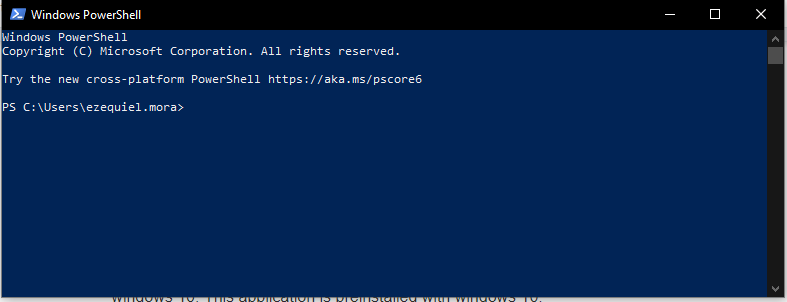
1. This will create two docker containers, one for the web API and another for the Database. Automatically when the project finished the building process, it will open a web browser with the API Endpoints using Swagger. The compilation has completed when the console output says “Ready”.



1. If the web browser doesn’t open automatically you can manually enter the following URL

<https://localhost:44338/index.html>

1. You can check and manage the status of the containers using the CLI PowerShell application for windows 10. This application is preinstalled with windows 10.

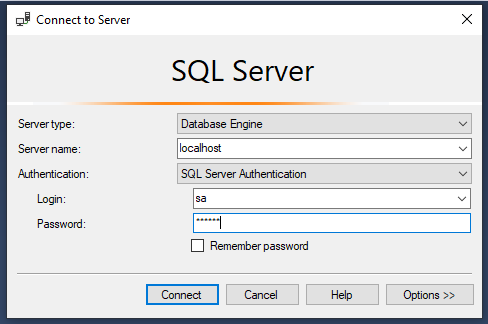


Some of the basic Docker commands include the following

|  |
| --- |
| ***docker image ls #See all created images***  ***docker ps -a #See all running containers***  ***docker stop container\_id #Stop a running container***  ***docker rm container\_id #Remove from hub a stopped container***  ***docker rmi image\_id #Delete from hub a stopped image*** |

For more commands and detailed explanations of each one please review the official Docker documentation.

1. If you wish to view the Docker database container and make queries you need to open SQL Server Management Studio with the following credentials and options.



NOTE: The password of the Docker database is “"admin123!"” and the default username is “sa”.

# 

# Front-end Installation

Similarly to the back-end, in the front-end installation we have two alternatives to run the web app, this can be run as a Node application or as a Docker container; like we said, we recommend the Docker container version of the deployment.

Similarly as we did with the back-end repository we need to enter the front-end repository dashboard <https://github.corp.globant.com/ErnstYoungX/TalentSurferInternalModuleUI> and press the “Clone” button to get the .git link. This way we repeat the previous step with another repository instead.

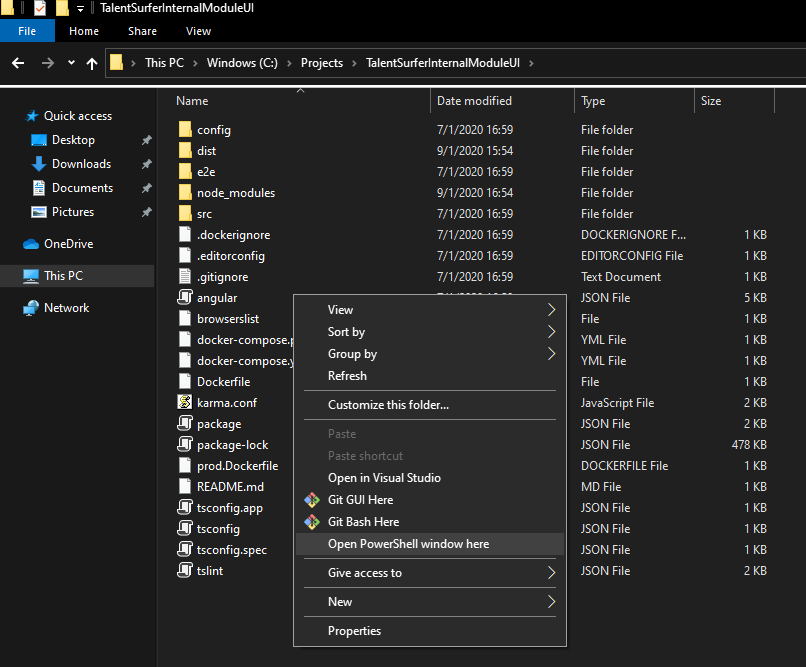
|  |
| --- |
| ***git clone -b develop https://github.corp.globant.com/ErnstYoungX/TalentSurferInternalModuleUI.git*** |

NOTE: Remember that if you are using SSH credentials you will need to clone the SSH version of the \*.git repository.

I) Angular and Node version

This version requires the prior installation of Node and NPM, and as we see in the installation prerequisites we also need to install the Angular CLI to run and build Angular apps.

1. Secondly, we need to open a new Powershell window in the recently downloaded repository folder. To do this we can either navigate to the folder using Powershell commands or do a secondary click over the repository folder while pressing SHIFT and select “Open Powershell in this folder”.



## 

1. In this new Powershell window we need to type the following commands.

|  |
| --- |
| ***npm install -g @angular/cli***  ***# To install the Angular CLI - this doesn’t don’t require to enter the repository folder since it is a global configuration***  ***cd repository-folder***  ***npm install # To install all npm dependencies into the repository folder***  ***this process will take a while***  ***ng serve # To Run the Angular application as a webapp***  ***Optional: in some case you will get an error trying to run the app in that case type in the powershell console this command***  ***Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser*** |

By default the Angular web app runs in the following URL <http://localhost:4200>.

IMPORTANT NOTE: You close the web app by returning to Powershell and pressing CTRL + C.

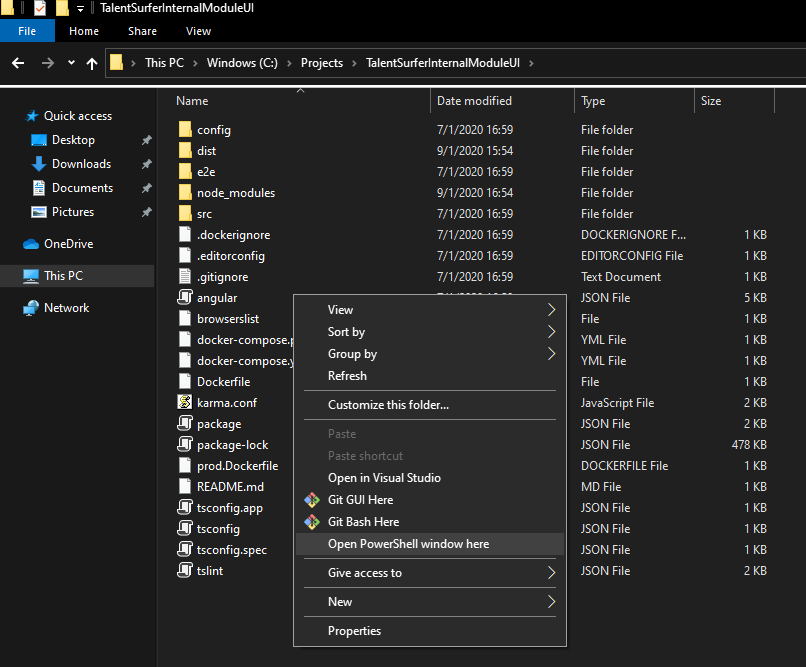
# Docker - Front-end Installation

The process to build and run the local version of the front-end in Docker is much simpler than its Node counterpart and is recommended since this will be the method to use in the cloud and continuous deployment to the server. During this alternative method installing the Angular CLI is optional since it is downloaded and included during the Docker image building process defined in the Docker file.

IMPORTANT NOTE: If you run the Docker version of the back-end it is recommended that you run the Docker version of the front-end too, and always remember to stop and remove all the containers if you make a new build. Also, it is recommended that when you run a new build of the front-end web app when you enter this URL in the browser to press CTRL + F5, this will ignore the cache and load the newest version of the compiled web app.

NOTE: You can also erase the cache from the Internet browser settings too.

## Go to the project folder and inside this folder, do a secondary click while pressing SHIFT and select from the drop down menu “Open Powershell in this folder”.



1. Now you will be able to build the application using Docker with the following commands.

|  |
| --- |
| ***docker build -t my-angular-app:v1 . #Build the webapp in docker inside the folder***  ***docker run -p 4200:80 my-angular-app:v1 #After the build finished need to run the webapp*** |

When you run the Angular app as a Docker container you need to bind the port 80 to the port 4200 in the “Docker run” command.

1. By default the Angular web app runs in the following URL <http://localhost:4200>. To close the running application you need to go to the Powershell window and press the combination CTRL + C.

NOTE: Maybe when you enter this URL you will see a security warning since the URL is not HTTPS securitized so in your browser press the button “Advance” and later “Continue anyway”.

1. After stopping the Angular Docker app you need to stop and remove the container as well as the image since this process is not automatic.

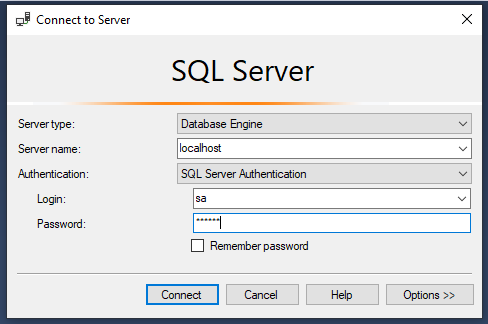
|  |
| --- |
| ***docker image ls #See all created images***  ***docker ps -a #See all running containers***  ***docker stop container\_id #Stop a running container***  ***docker rm container\_id #Remove from hub a stopped container***  ***docker rmi image\_id #Delete from hub a stopped image*** |

NOTE: You don’t need to type the complete image id, only the first 3 characters.

# Extra Module - Testing the local environment

As mentioned we recommend deploying the Docker versions of both the back-end and front-end since the URL are mapping to those addresses and ports. When you have both services up, back-end and front-end you will be able to run and interact with the Talent Surfer application.

1. Enter the Angular web app URL.
2. Sign in using the SSO (single sign on) of your Google corporate account.
3. The page will work but you will require permissions.
4. The permissions are granted by an SQL query opening the SQL Server Management Studio.



1. Add administrative permissions to a user in the Talent Surfer database by making the following Queries.

|  |
| --- |
| ***INSERT INTO [AspNetUserRoles] VALUES (1, 1)***  ***INSERT INTO AspNetUserClaims VALUES (1, 'http://schemas.microsoft.com/ws/2008/06/identity/claims/role', 'Admin')*** |

**IMPORTANT NOTE: It is possible that, in some cases, when you run the Talent Surfer front-end it won’t start properly until you close the connection with SQL Server Management studio.**

**The complete and recommended process is the following:**

1. **Log in to Talent Surfer to create the initial database.**
2. **Secondly, create the connection and then execute the queries.**
3. **Always close the connection before entering the Talent Surfer URL again.**
4. **Open the Web Browser settings and clear the cache and cookies**

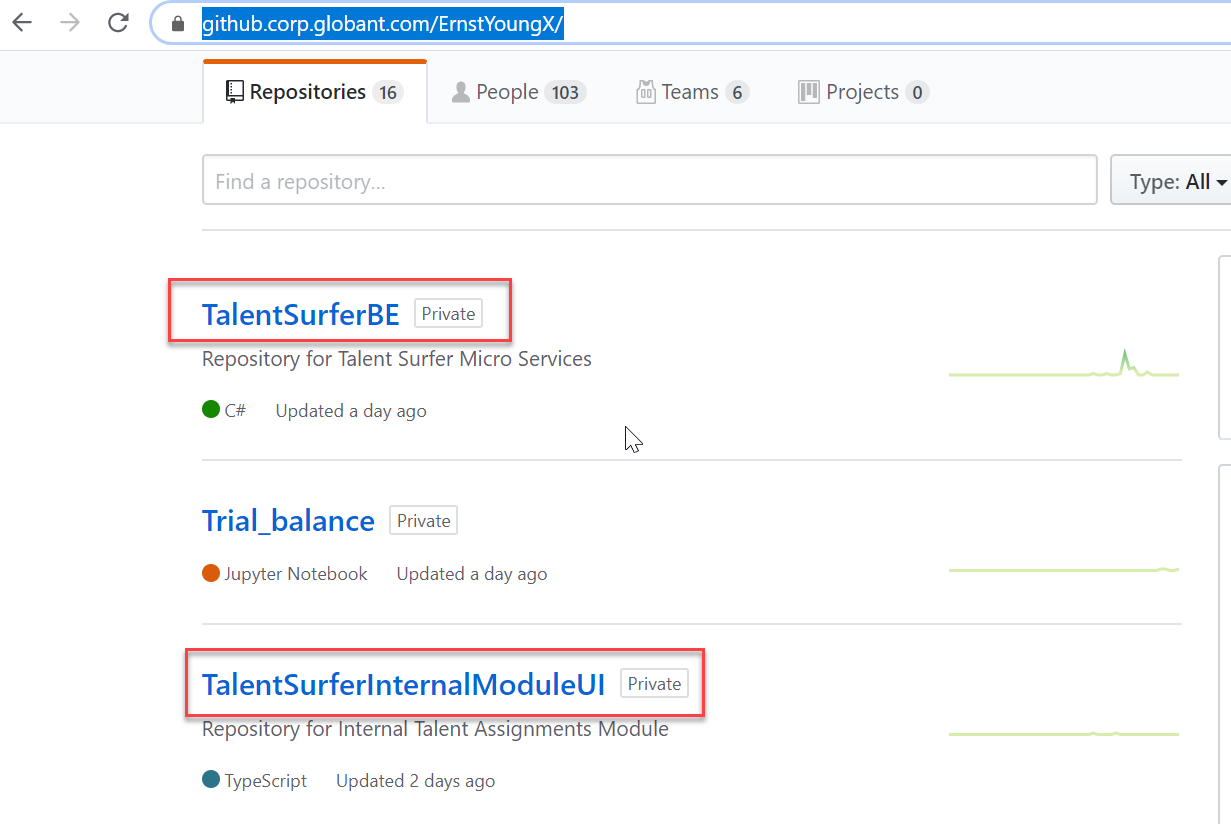
# Create Pull request, squash and merge

We must create a **PR** after finishing a **US**, **CR** or the solution for a bug, with the final purpose to integrate the branches with the develop branch. For this we must do the following:

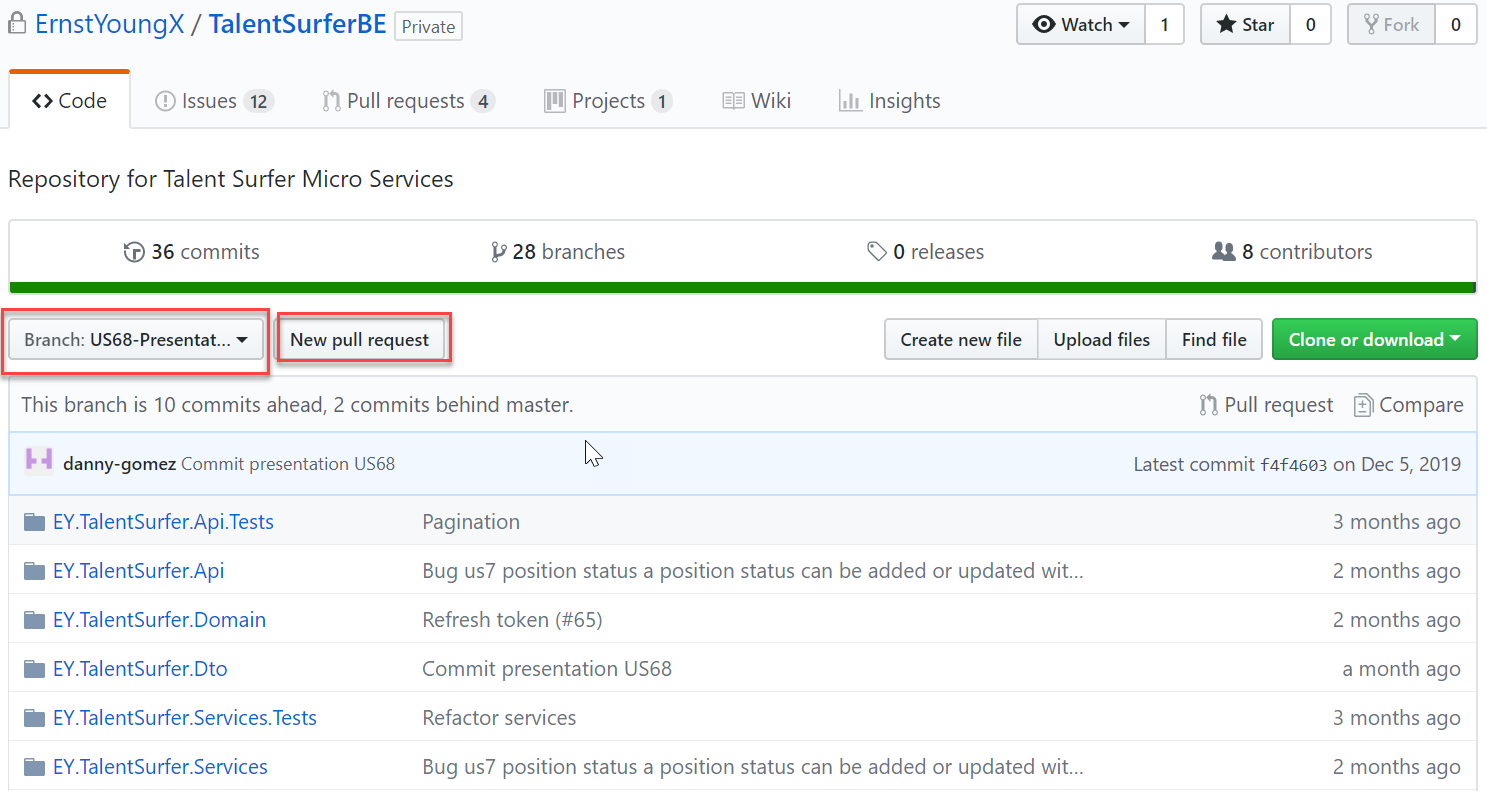
* Enter Github

<https://github.corp.globant.com/ErnstYoungX/>

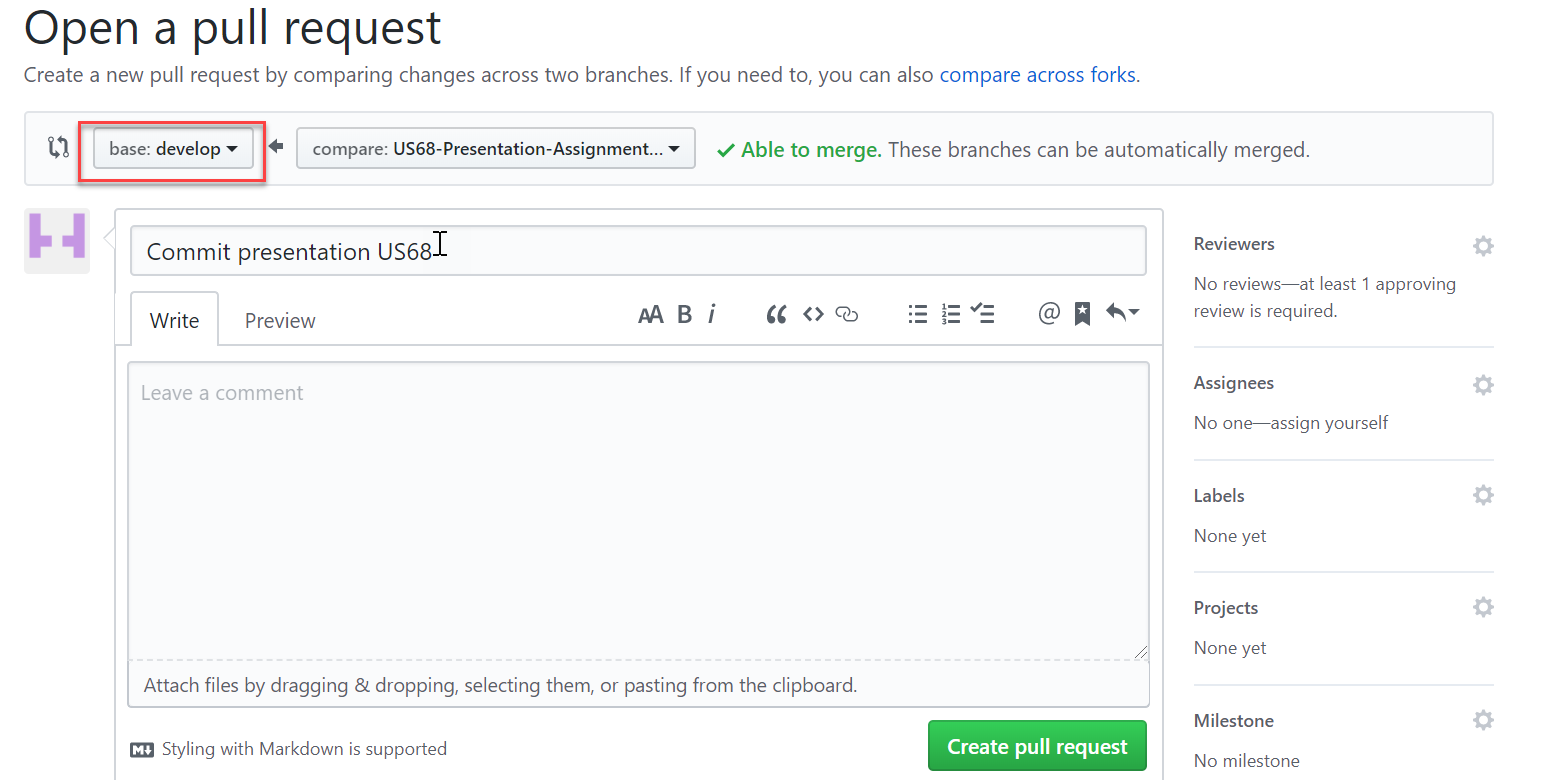
* Choose the repository, back-end or front-end



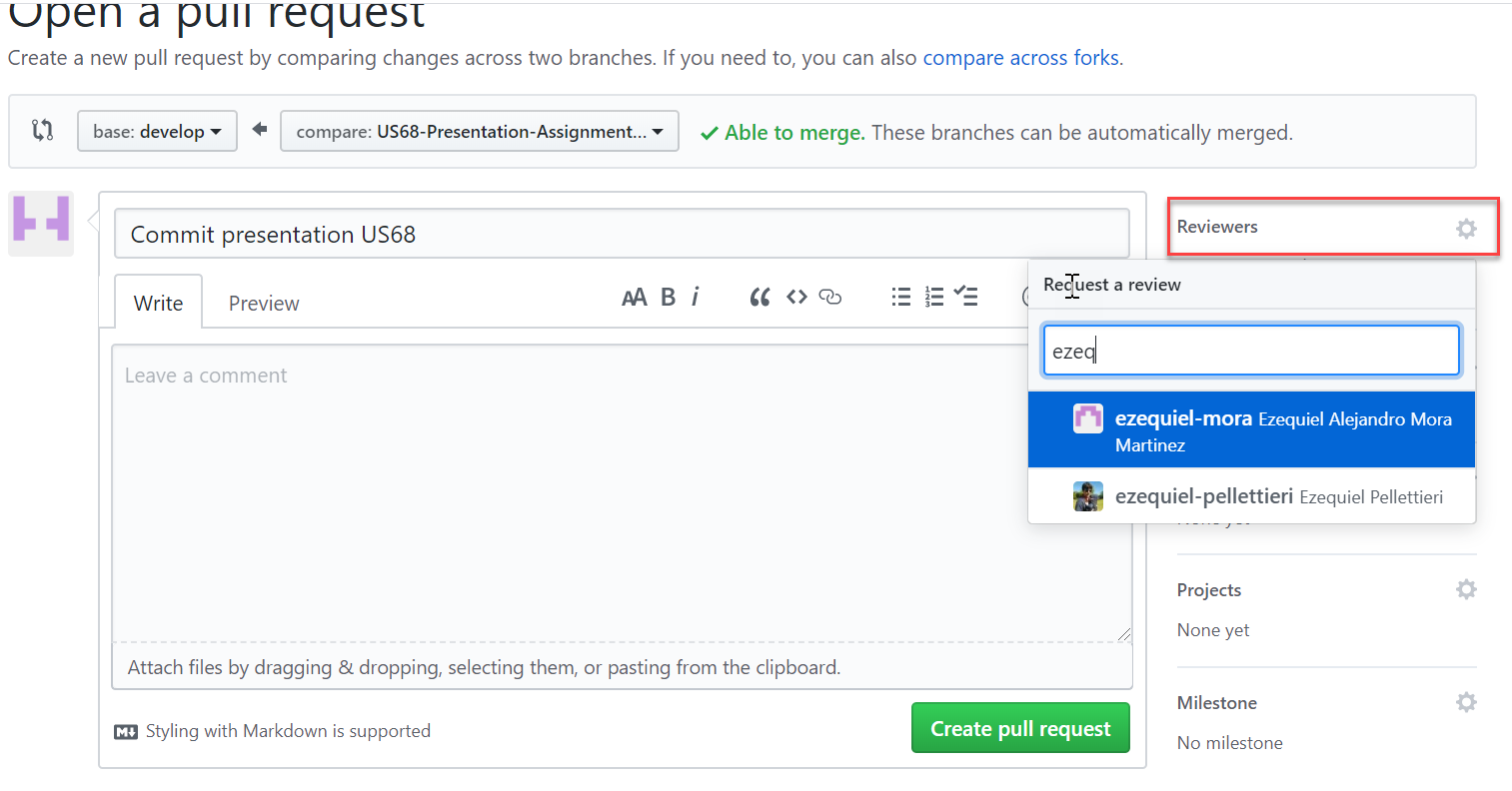
* Select the working branch



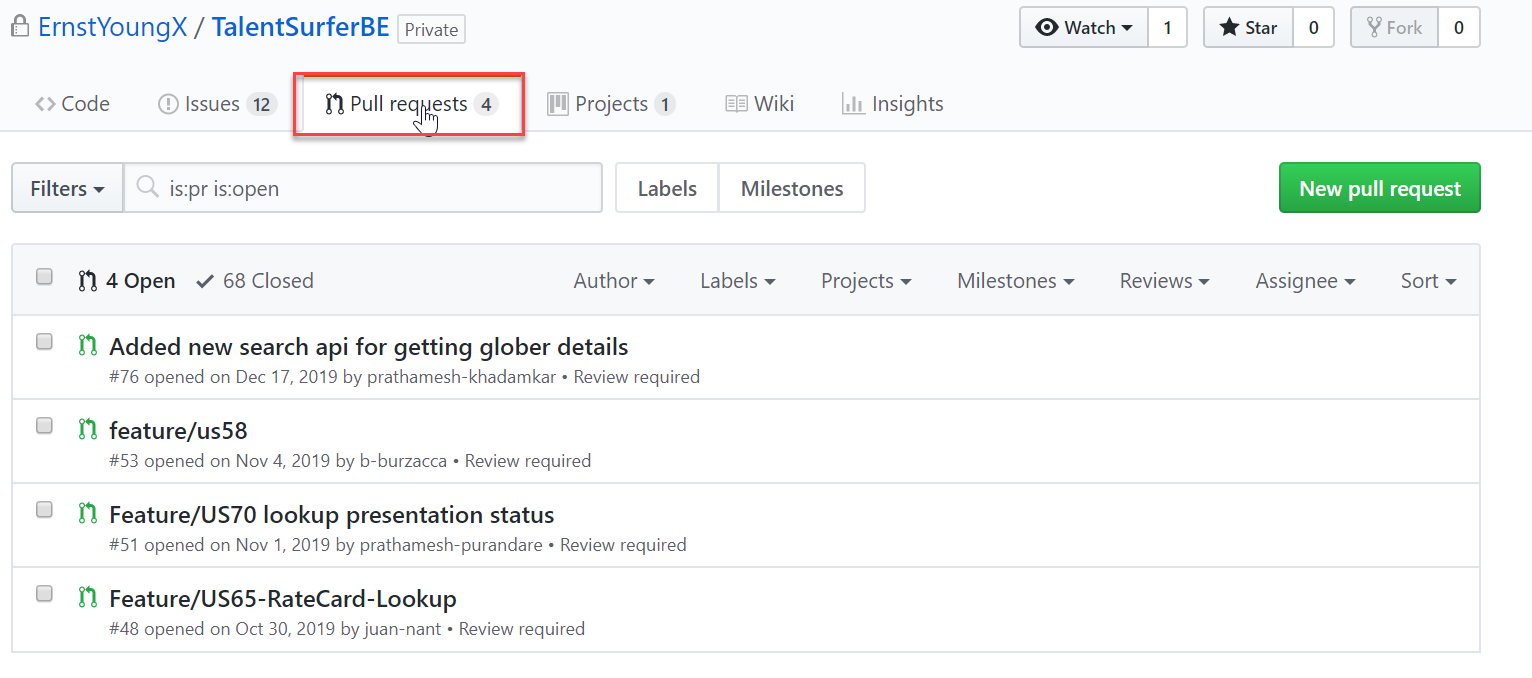
* Click on “New pull request” Button
* Select the base branch which is the develop branch.



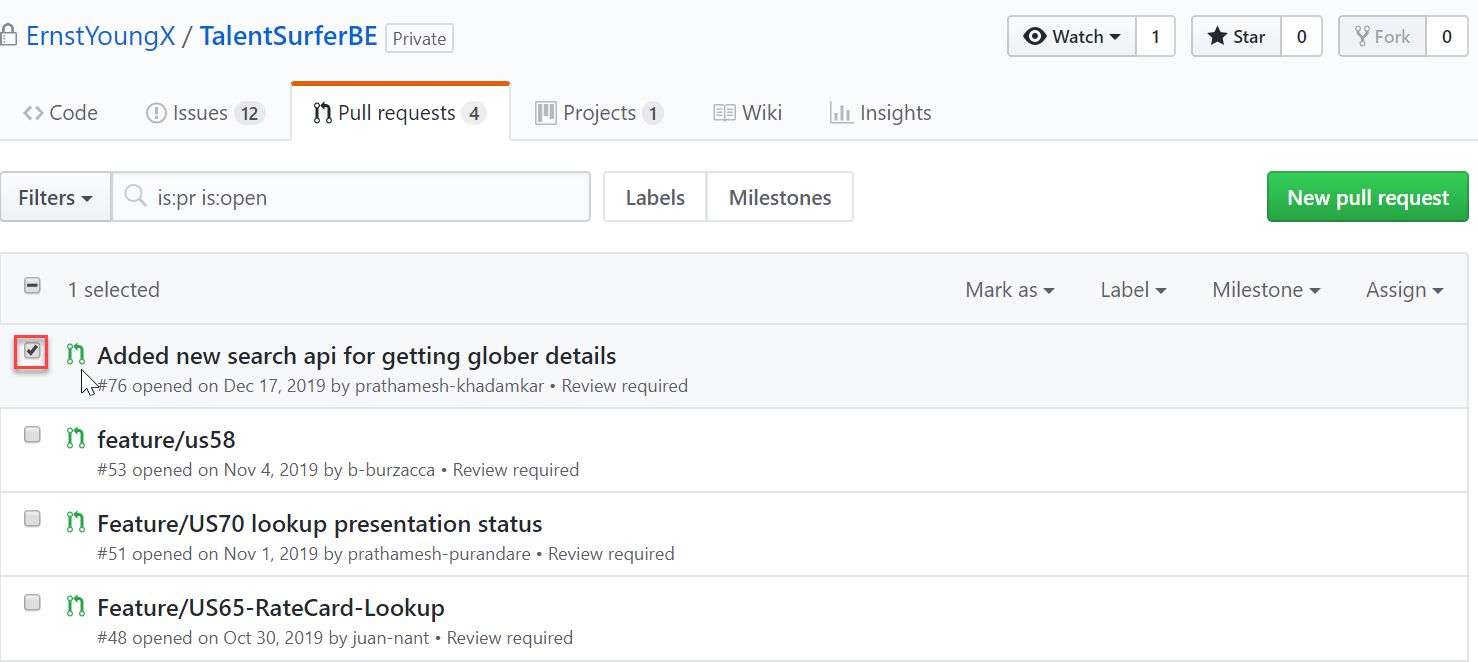
* Select the reviewers



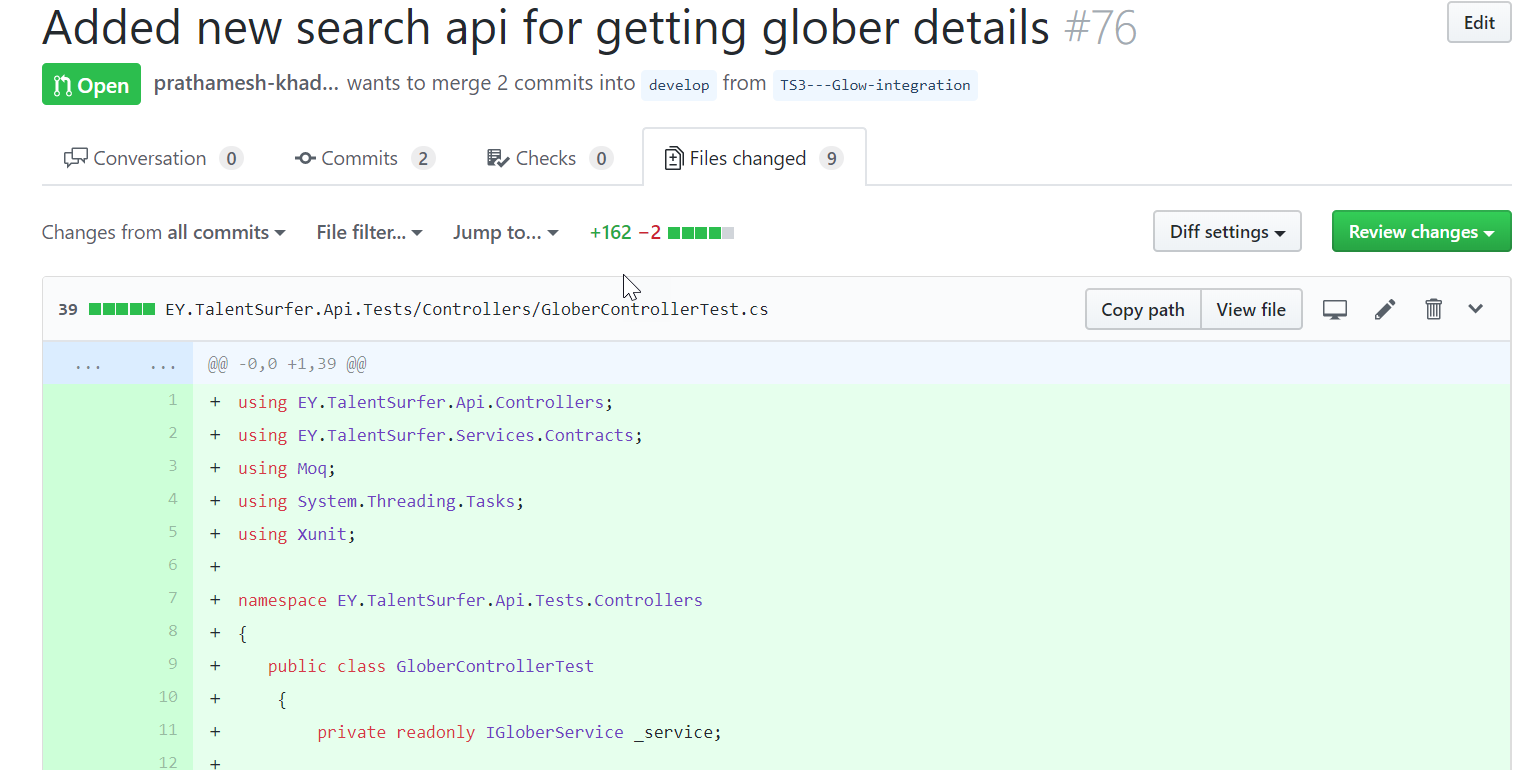
* Add comments and click on the “Create pull request” button.
* After this the reviewers receive an email about the creation of the pull request.
* The reviewers must enter to github and click ok the “pull requests” option.



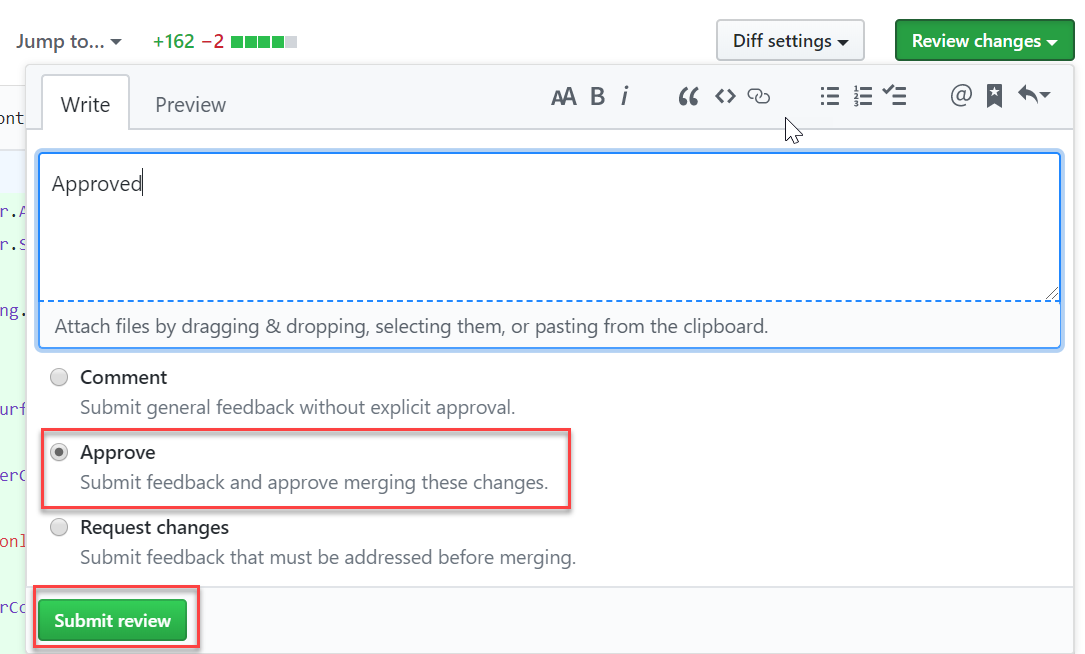
* Select the pull request to approve



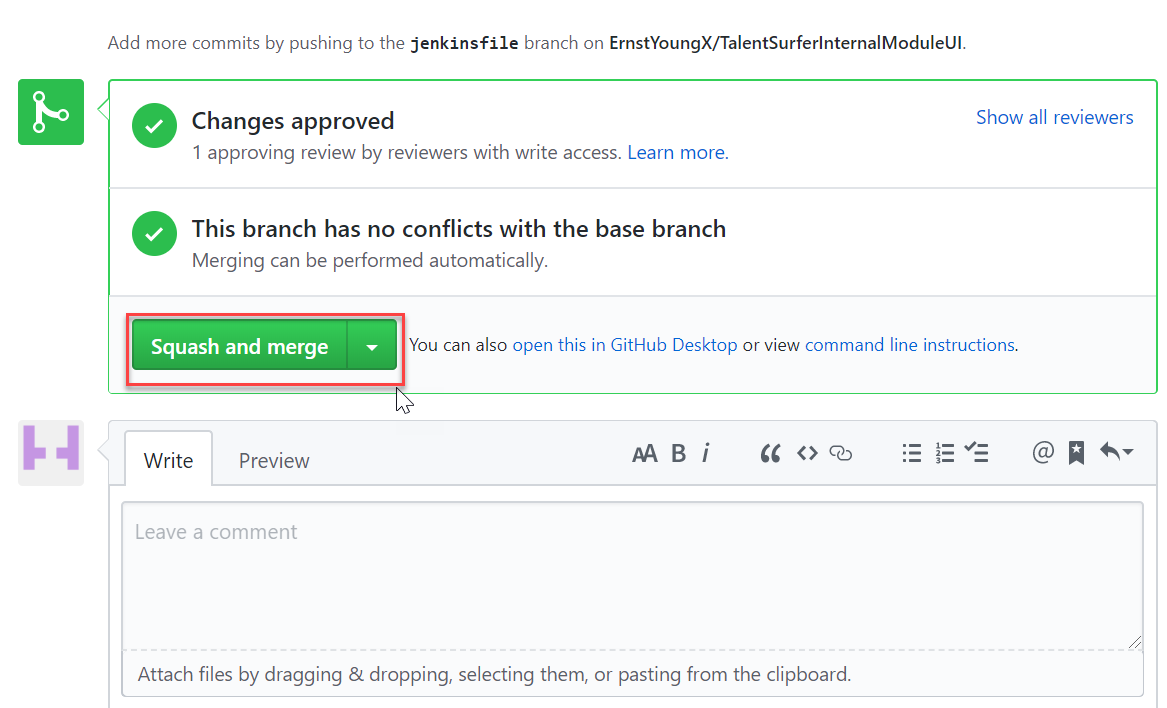
* Once there, click on the “Files changed” tab, and check the changes

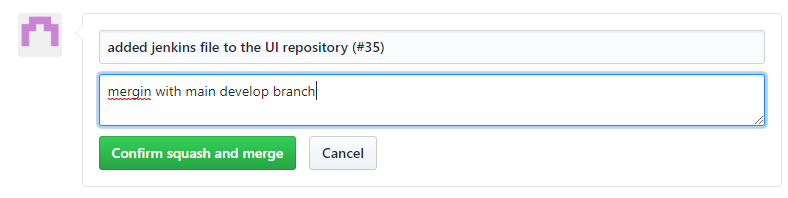


* When the reviewer has checked the files and changes, click on the “Review changes” button.
* Select the Comment, Approve or Request Changes action and add the comments and click on the “Submit review” button.



* After doing this, whoever made the pull request can do the squash and merge to the develop branch if only the branch hasn’ t conflicts to resolve.





* Finally, we must delete the branch in which we were working.

