**What Is Git?**

GIT – initially developed by Linus Torvalds is a version control system that is used to manage the source code of a software application.

It facilitates many developers to work collectively at the same time, on the same file without providing any hindrance to the others.

**What Is GitHub?**

Let us consider a situation for Example. Suppose you are working in a team to develop a software application and two members of a team are working on a specific module. Let us consider them as A and B.

For that module, the developer A will have a copy in his PC and developer B will have a copy on his PC. Now at this point, if developer B changes the code in the module, then developer A wouldn’t know the changes made by the developer B and vice versa.

To overcome this problem, we have a platform which is called GitHub. Github is a Web hosting service where the source code of a software application can be uploaded/saved through the git software (installed on your local PC) or you can directly copy the code from your PC and save on the server.

Features of GitHub

**Enlisted below are the various features of GitHub.**

* **Distributed:** GitHub provides a distributed network, which means it provides a backup of the code. Thus if in case the central server crashes, the coder has its copy in the local repository. It saves each version or each copy of the changed code.
* **Compatible:** Suppose if you are using any other version control system like SVN and you want to switch to GitHub, then you can easily do that without creating new code again.
* **Branching:**Branching is a unique feature provided by GitHub. The developer needs to pick a section of the code from the remote branch, in that the branch developer can make changes, merge or delete codes within a few seconds.
* **Secure:** GitHub uses security features of SHAI( encrypted hexadecimal code) for any changes /commit made by the developers. Thus, it maintains the confidentiality of the developed project.

Basic Terminologies

**Let us become familiar with the basic terminologies associated with GitHub.**

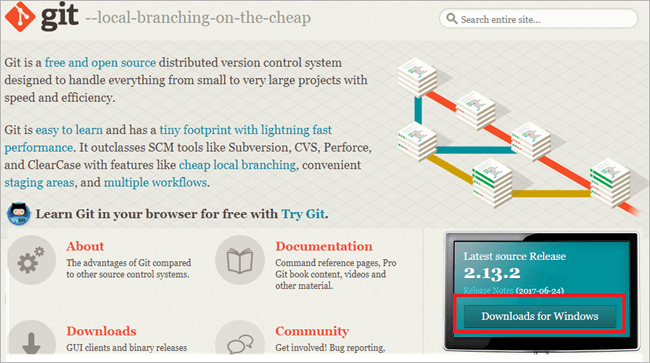
1. **Pull Request:** If you have made changes in code/script, to show the other collaborators you send a pull request.
2. **Repository:** You can simply, treat it as a storage area of your workplace that contains all your documentation files and the history of changes.
3. **Fork:**It is a copy of other's repository in your account in which you can make changes and it won't affect the original code.
4. **Commit:**Whatever the changes you make in your files will come under commit. Every change is saved under a particular name or ID which is also called “revision”.
5. **Branching:** When you extract a portion /section of code from the main or remote track of your software, then it is called ‘branch' and the process is known as Branching.

### How To Install Git?

**Follow the below steps to install Git:**

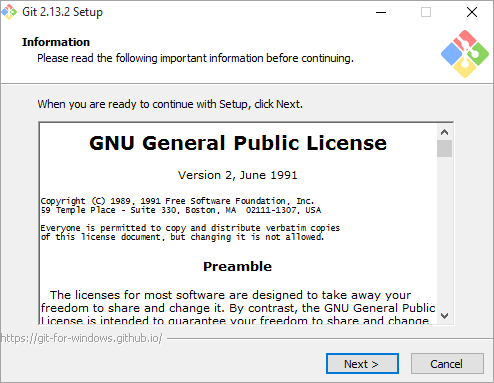
**#1)** Navigate to https://git-scm.com/.

**#2)** Click on the **download button**, as shown in the below image. This will download the latest stable version of Git in your system.

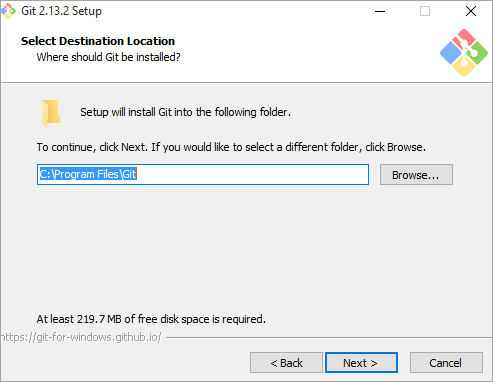


**#3)** Double click on the **downloaded setup file** to launch the installation process.

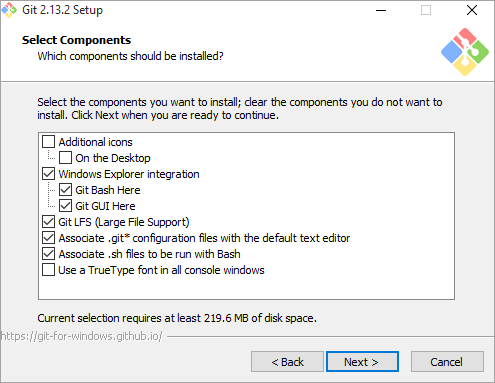
**#4)** Accept the **GNU General Public License**. Click on the Next button.



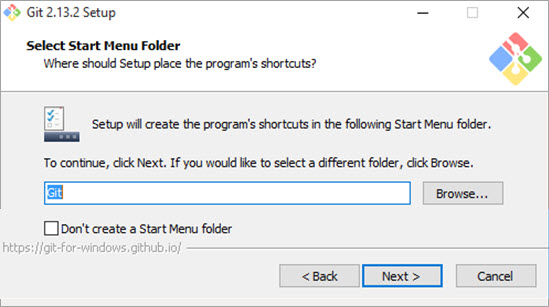
**#5)** Provide the **installation location** where you want to install Git. Click on the **Next button**.



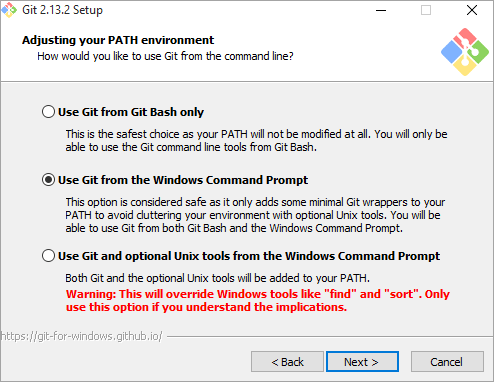
**#6)** Select the following components as shown in the below image to install. Click on the Next button.



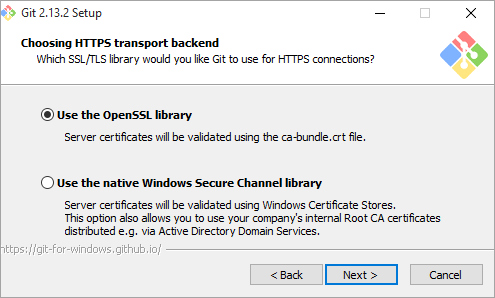
**#7)** On this wizard, there is an option to create a shortcut. By default, the Start Menu folder where the shortcut icon will be created is pre-selected. Provide the location if you want to change, else leave it unchanged. Click on the Next button.



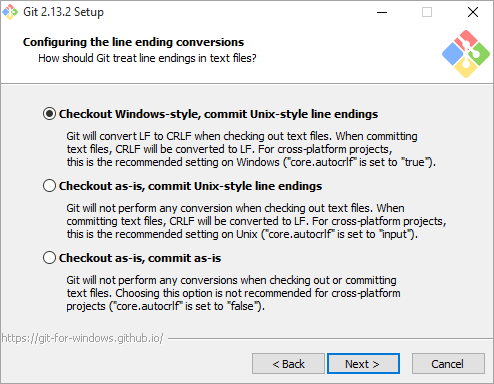
**#8)** On this page of the wizard, we are given an option on how to use Git from the command line. Select the **second option** and by selecting this we will be able to use Git from the Windows command prompt. Click on the Next button.



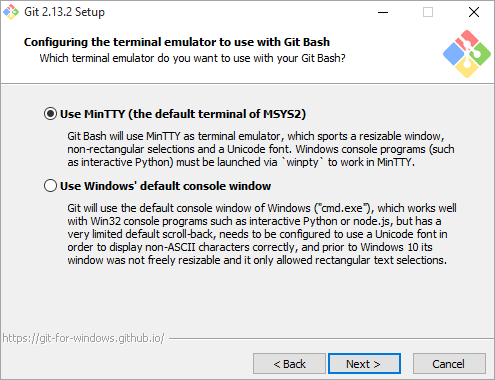
**#9)** The following screen is displayed on the wizard. Click on the Next button.



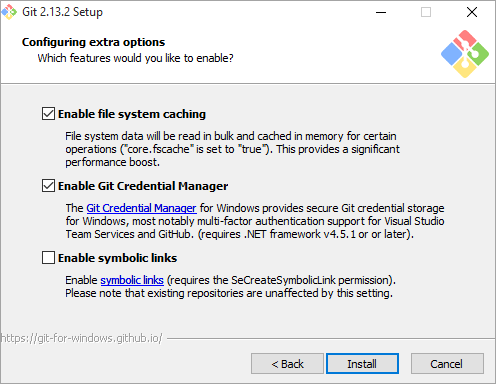
**#10)** On the next page of the wizard, there is a provision to configure line ending conversion. Leave the default settings and click on the Next button.

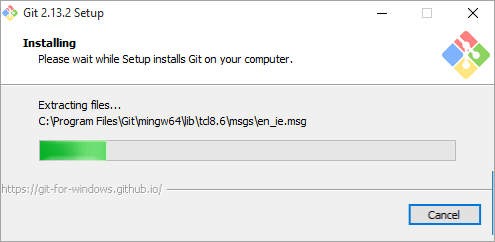


**#11)** On the next page of the wizard, select the terminal emulator which you want to use with Git bash. Leave the default settings and click on the Next button.

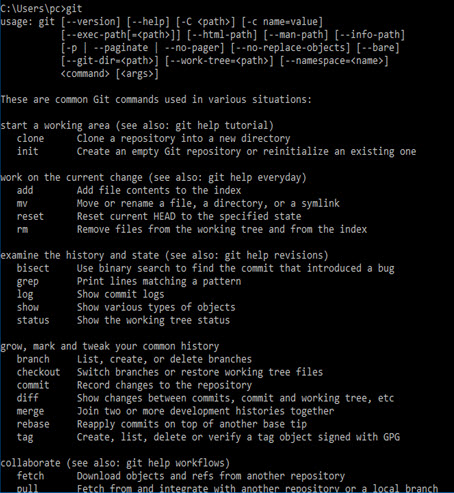


**#12)** On the last page of the wizard, there is an option to enable the extra features provided by Git. Select the options as displayed on the image and click on the Install button.



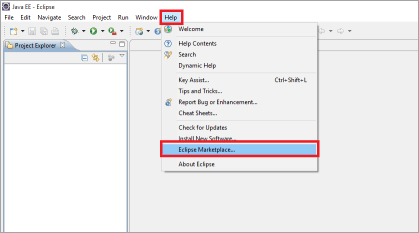


**Note:** Once the installation process completes, we need to verify if it was executed successfully or not. To do that, open the Command Prompt and type “Git” and press the “Enter” key. If the screen displayed below matches your output then it means that the installation process was successful.

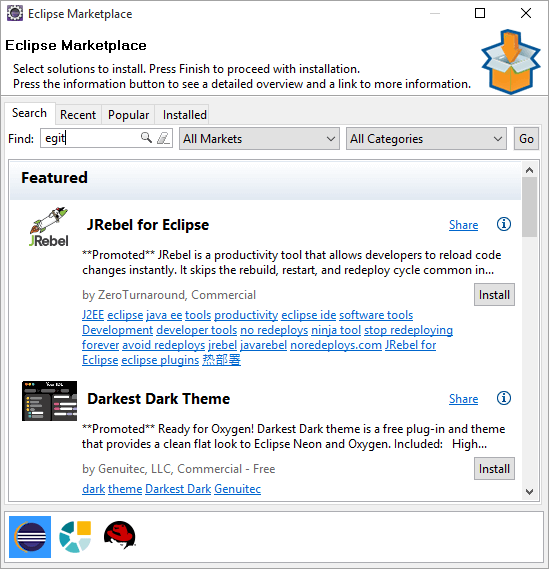


### Set Up Eclipse with Git Plugin

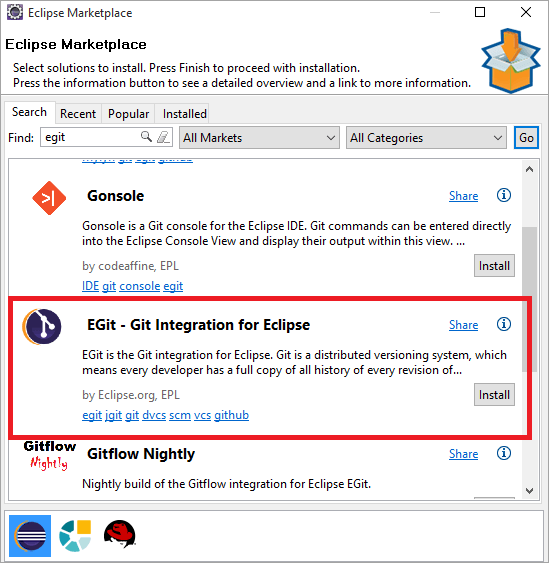
**#1)** Launch Eclipse and navigate to **Help => Eclipse Marketplace**.



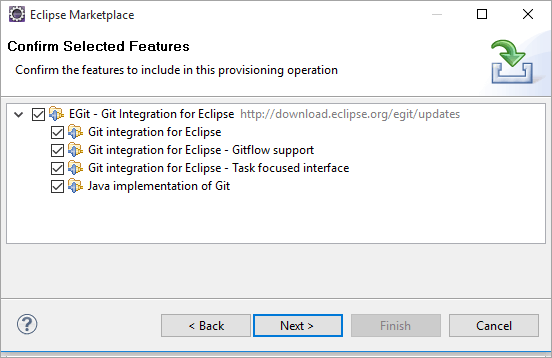
**#2)** The following screen will be displayed as shown in the below image.



**#3)** Type “EGit” in the find section as shown in the below image.  
**#4)** Click on Go.  
**#5)** Click on the install button to Install **“EGit – Git integration for Eclipse”**.



**#6)** Select all the options as displayed in the below image. Click on the Next button.



**#7)** Accept the license agreement and click on the Finish button.

Thus, you have successfully installed the Git plugin on your PC.

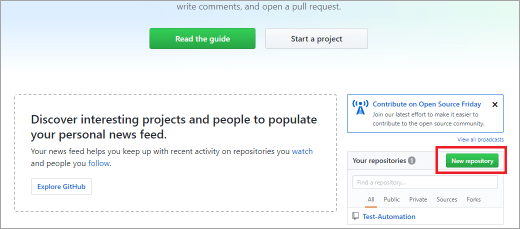
### Create Repository on GitHub

Before learning the process of creating a repository, let us understand what a repository is.

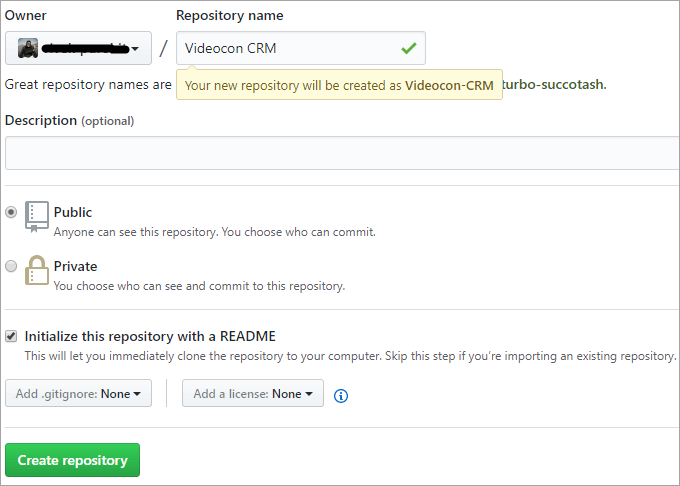
A Repository in context with GitHub means a central location where all data, files, etc., can be stored. It is particularly used to efficiently co-ordinate the activities of a project. You can add java files, screenshots, videos, Excel sheets, documents, etc., in short, all that which your project needs can be added to a repository.

**Follow the below steps to create a repository on GitHub:**

1. Navigate to https://github.com/.  
2. Complete the sign-up process.  
3. Log in with valid credentials.  
4. Click on **“New Repository”** as shown in the below image.



5. Enter the name of the repository in the **“Repository name”** text box.  
6. Provide a description (optional) of the repository.  
7. Click on the **“Create Repository”** button.



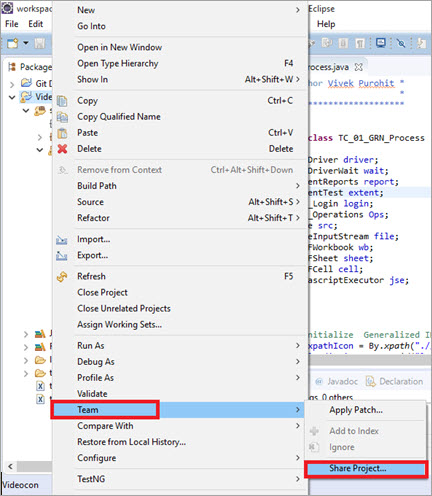
Thus a repository gets created.

## ****Selenium Integration With GitHub Using Eclipse****

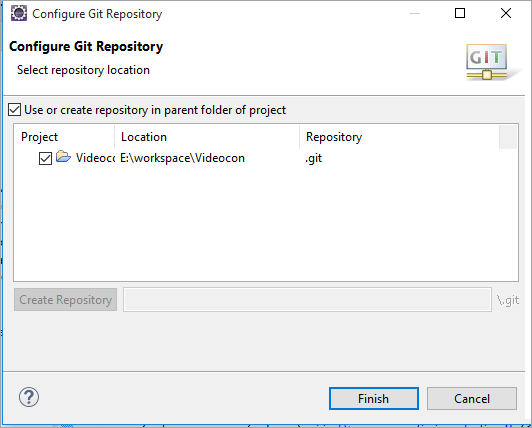
**Given below are the steps involved in the Integration of Selenium Automation Script with GitHub Using Eclipse IDE**

**#1)** To integrate Selenium with GitHub, launch Eclipse IDE and navigate to the Selenium Automation project which is to be synced with GitHub.

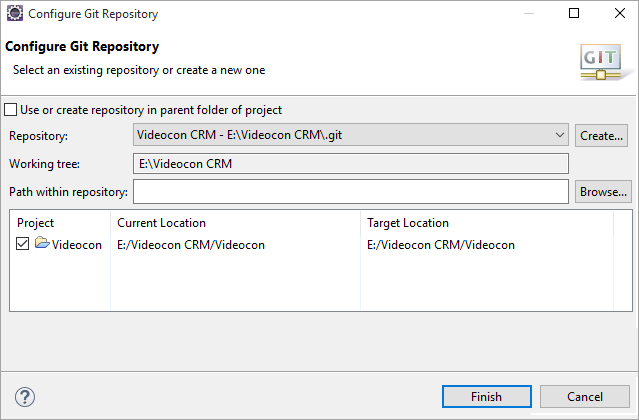
**#2)** Right-click on the project and navigate to **Team => Share Project**.



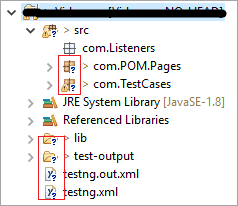
**#3)** Select the Repository from the drop-down. If no data is displayed in the dropdown, then click on Create.



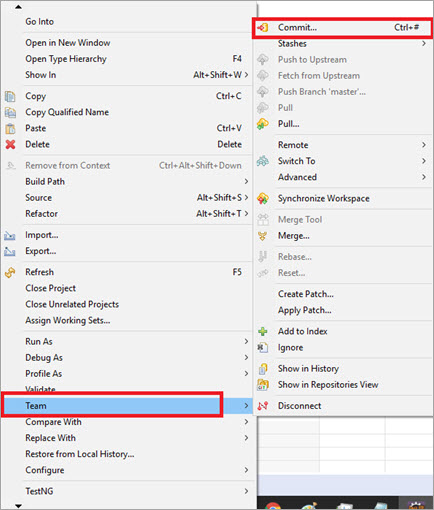
**#4)** Click on the Selenium Automation project which you want to integrate with GitHub. Click on the **Finish** button.



**#5)** Notice the following change in the structure of your Selenium project.

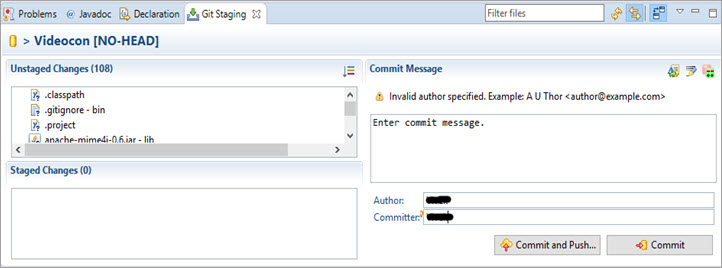


**#6)** Right-click on the project which was configured with the Git Repository. Navigate to**Team => Commit**.



**#7)** Right-click on the Selenium automation project and navigate to **Team => Add to Index**.

**#8)** Enter the commit message and click on the **Commit button**.

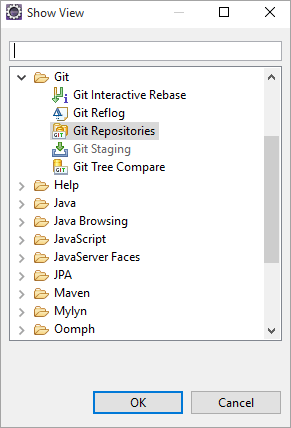


This will add all your Test Cases files to staged changes.

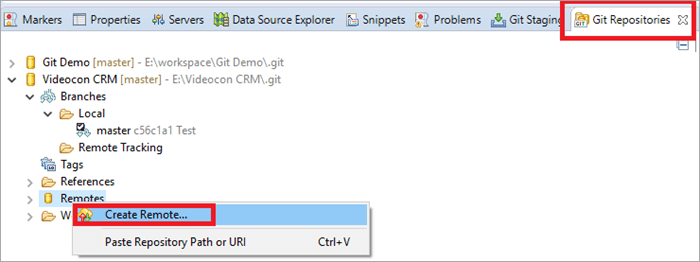
**#9)** Open Git repository tab in Eclipse.

**Note:** If the tab does not open by default, then follow the below steps.

**a)** Navigate to **Windows => Show view => Other**.  
**b)** Under the Git folder select, **Git repositories and Git Staging** and click on the OK button.



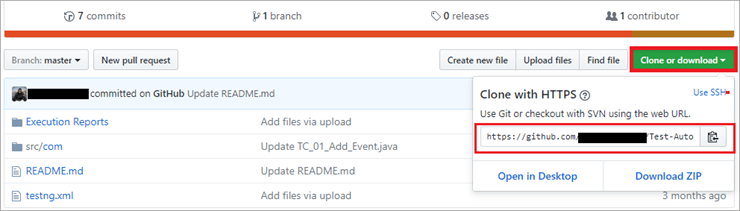
**#10)** Under the Git Repository, right-click on remote and navigate to create remote.



**#11)** A new pop-up window will open, provide the remote name. Leave the other settings unchanged and click on the OK button.

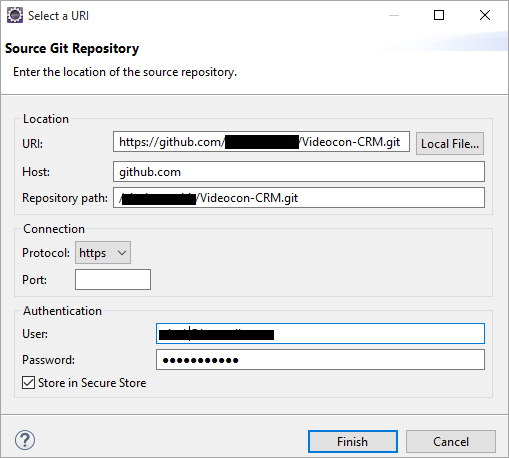
**#12)** Another pop-up window will open, provide the URL of the GitHub repository which can be copied by following the below-mentioned steps:

**a)** Navigate to the **created repository** on GitHub.  
**b)** Click on Clone or download as shown in the below image.

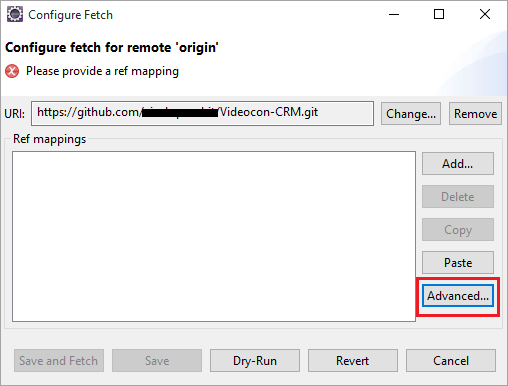


**c)** Copy the URL.

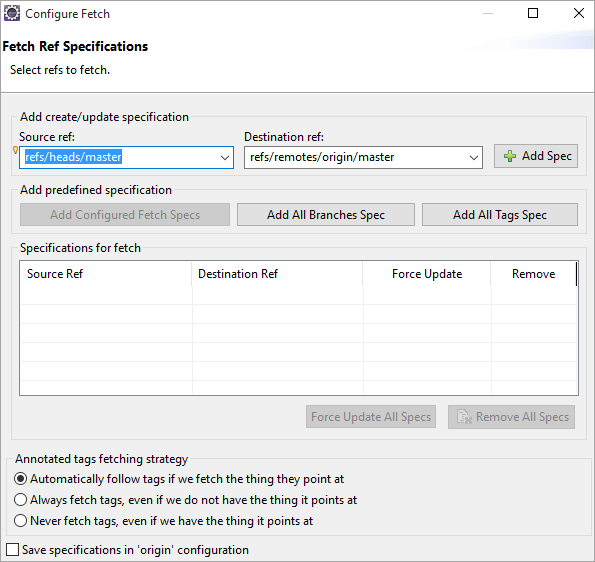
**#13)** Provide the copied URL and the other details including your login credentials of the GitHub account as shown in the below image, and click on the Finish button.



**#14)** Once the configuration process is done, we need to select the branch in which we will commit changes. Click on the **Advanced button** as shown in the image.

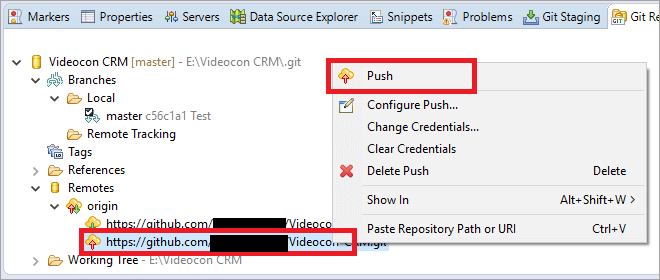


**#15)** Select your branch and click on the **Add Spec button**.

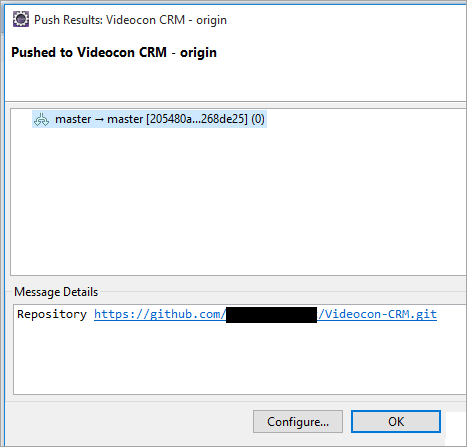


**#16)** Finally, click on the Finish button. Now under Remote, you will find a folder which is the name you provided in Step 11.

**#17)** Right-click on the URL with a red arrow, and click on **Push**.



**#18)** Thus, all the changes that were made and the Test Cases of your Selenium Project will get committed to the repository.



**#19)** Verify the updates in your GitHub account.