**What is GitHub?**

Git Hub is a Collaboration platform. It is built on top of git. It allows you to keep both local and remote copies of your project. A project which you can publish it among your team members as they can use it and update it from there itself.

**Advantages of Using Git Hub For Selenium.**

* When multiple people work on the same project they can update project details and inform other team members simultaneously.
* Jenkins can help us to regularly build the project from the remote repository this helps us to keep track of failed builds.

## ****WHAT IS A SOFTWARE REPOSITORY?****

A software repository is a central storage system where all the revised version of the code is stored, retrieved and changes made when it is required. Colloquially, a repository is called repo. Examples are Github, Bitbucket, Stash, etc.

### ****TYPES OF REPOSITORY****

A repository is of two types which are listed below:

* **Public:** A public repository is accessible to all without any restriction.
* **Private:**A private repository has limited access based on the permission granted by the administrator. We require an authorization code to access private repository.

### ****WHAT IS GITHUB?****

GitHub is a project hosting service to work in a collaboration with team. It is also a version control system (VCS); In simple word, GitHub is a central repository where we commit our code. It is built on the Git which is a version control software.

### ****DIFFERENCES BETWEEN GIT AND GITHUB****

Some of the major differences between Git and GitHub are listed below:

|  |  |
| --- | --- |
| **Git** | **GitHub** |
| 1. It is a version control software | 1. It is built at the top of the Git |
| 2. It is a software | 2. It is a website, a brand name and service provider |
| 3. Local installation file | 3. Cloud-based platform |
| 4. Primarily a command line tool. Eg: Git Bash | 4. Web-administered tool |
| 5. The desktop interface is called as Git GUI | 5. The desktop interface is called as GitHub Desktop |

### ****TERMINOLOGIES USED WHILE USING GITHUB REPOSITORY****

Some of the terminologies used while working with GitHub are as below:

#### ****REPOSITORY****

* It is itself a repository
* A GitHub repository should have the license file and README file as well
* It is used to keep source code and project files like Documents, data, etc

#### ****BRANCH****

* A branch created at GitHub repository is treated as the next version of the code
* The default branch is called- master
* A newly created branch is the copy of the master by default
* We merge branch to the master branch after making changes to the code

#### ****COMMITS****

* Any changes in the code are termed as commits
* Every commit should go with the appropriate comment

#### ****PUSH AND PULL REQUESTS****

* After commit, we push code from the Eclipse or Git SVN or Git Bash
* We can directly commit code at GitHub Repository (No Push required) through its portal
* With pull request, we download the latest version of the code to our local repo
* The difference in changes are shown by colors on a pull request

#### ****CLONE****

* Clone used to create the local version of the central repository

### ****HOW TO CREATE A GITHUB REPOSITORY?****

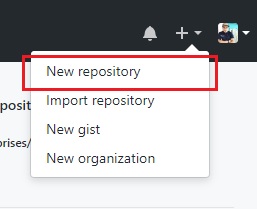
Let’s discuss the steps to create a GitHub repository. We can either create a public or private repository. The private repository is the paid service as it provides restricted code sharing access. Here are some of the steps to create a new GitHub repository:

#### ****STEP# 1: CREATE AN ACCOUNT AT GITHUB PORTAL****

Go to [www.github.com](https://www.github.com/) and create a new account there.

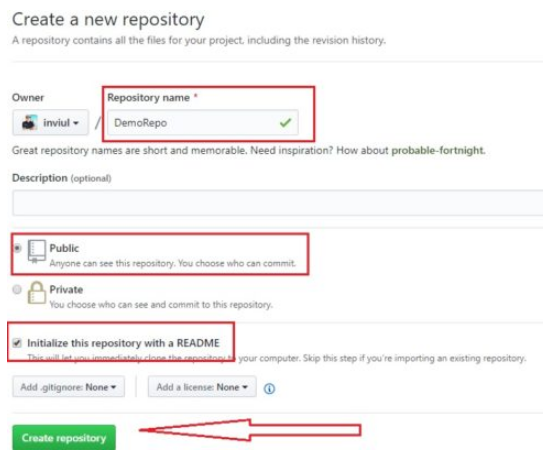
#### ****STEP# 2: ADD NEW REPOSITORY****

Find your account at the top right corner. There you see ‘plus (+)’ button. Click on it and then click on New repository option. See image below:



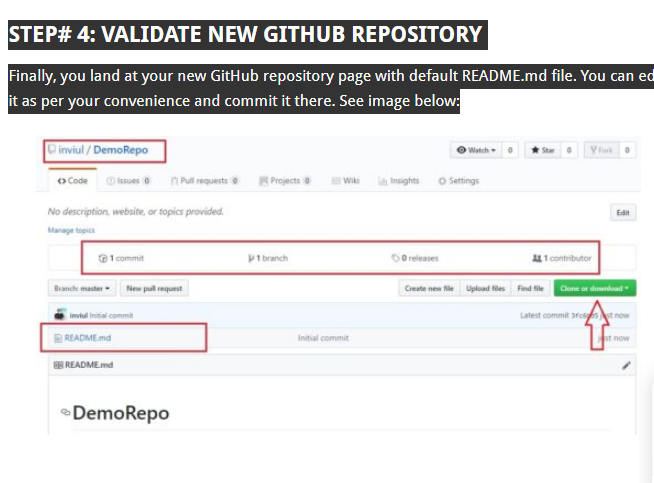
#### ****STEP# 3: CREATE A NEW REPOSITORY****

You land at Create a new Repository page. Here you fill details like Repository name, type of repository, check README file initialization then finally click on Create repository. See image below:



#### ****STEP# 4: VALIDATE NEW GITHUB REPOSITORY****

Finally, you land at your new GitHub repository page with default README.md file. You can edit it as per your convenience and commit it there. See image below:

This is how we create a new GitHub repository. Let’s discuss the integration of the GitHub repository with our Selenium project via Eclipse.

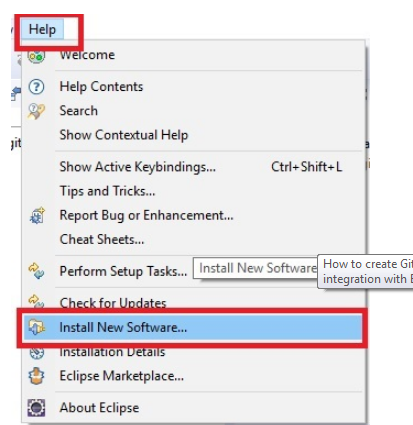
### ****HOW TO INTEGRATE GITHUB REPOSITORY WITH ECLIPSE FOR SELENIUM PROJECT?****

Following steps guide you through the way to integrate Eclipse with your GitHub repository.

### ****#1. INSTALLATION OF EGIT PLUGIN****

#### ****STEP# 1: INSTALL EGIT PLUGIN IN ECLIPSE****

Go to Help Menu of Eclipse then click on Install new software. See image below:



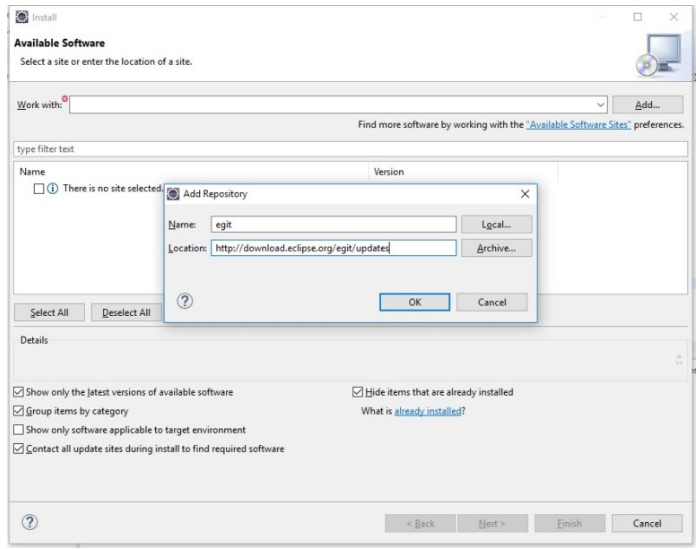
#### ****STEP# 2: FIND THE EGIT PLUGIN AND INSTALL IT****

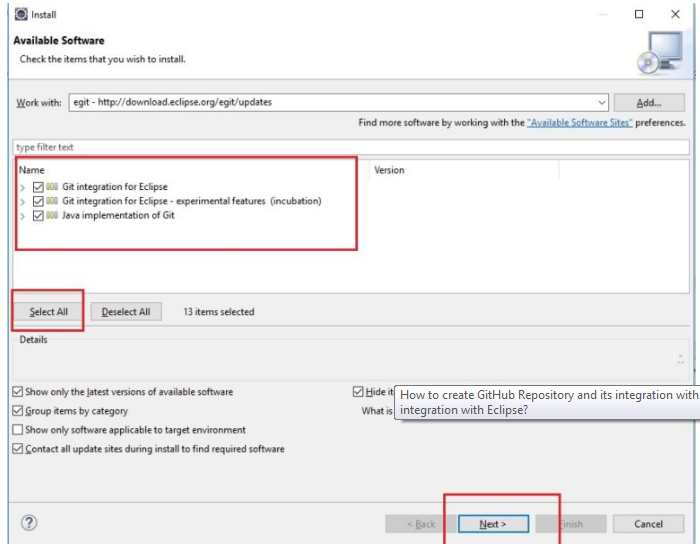
You will land to Installation window. Click on Add button and give name and Location of the plugin.

Name = Egit

Location = http://download.eclipse.org/egit/updates

See image below:





#### ****STEP# 3: REVIEW LICENCE AND PROCEED WITH INSTALL REMEDIATION PAGE****

You will get Remediation page twice, click Next and proceed further. Then accept the license and Finish installation. Eclipse will prompt to restart, go for it. See image below.

<http://www.inviul.com/selenium-github-repository/>