**GIT Installation and setup – Step by Step - For Configuration Management and Version Control**

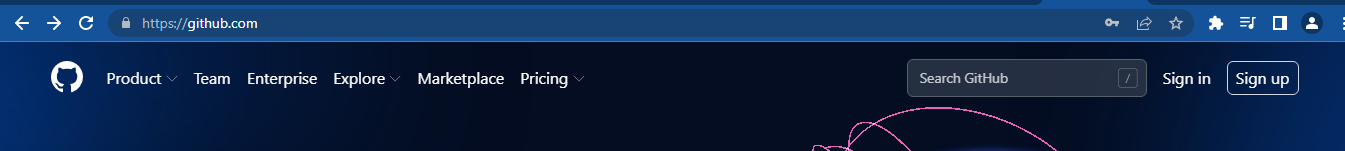
There are two parts to the installation.

**Part 1 Server-side view:**  First part is if we want to create our own repository as admin, we need to create a GitHub account with user name and password. This is optional step once customer admin already created the repository in corporate development projects

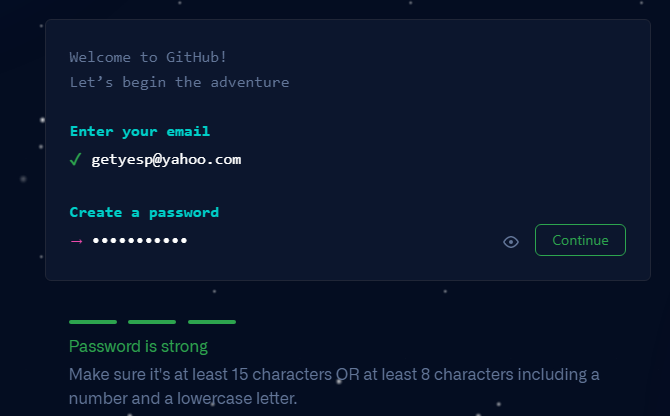
**Part 2 client side or developer view:** Secondly client-side view to clone the repository and get the status of files on the git server which is also called github.com

Part 1 **Server-side view**

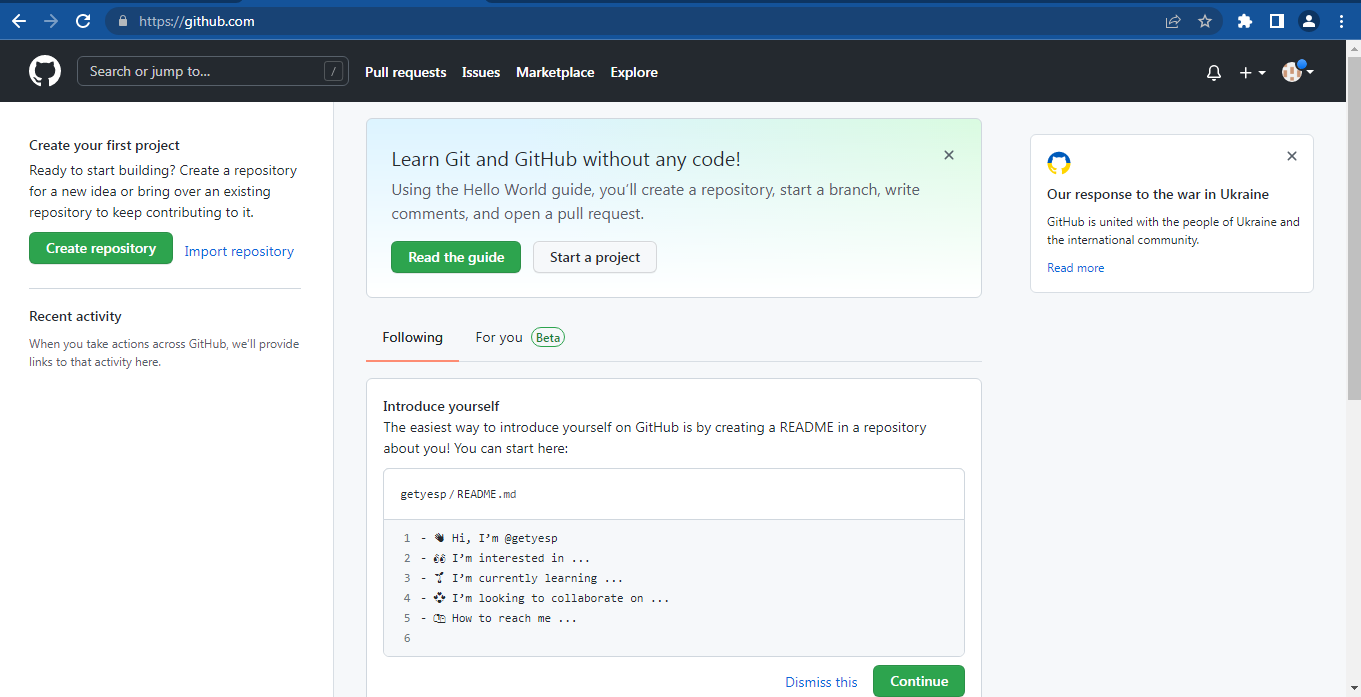
Go to git hub site. - <https://github.com/join> - Fill in the details.



1. Create GitHub account by providing email and password. Activate account



2. Once git account is created login to create repository. For the real time projects, this step will be done by customer and we don’t need to do. We can just clone repository and customer admin will create a repository for us. But for the students, you have to create repository.



3. Follow below steps to create a repository.

Click on create repository. By default, GitHub gives you URL with your git user name and creates repository by user name. For example, if my user is getyesp,

https://github.com/getyesp/getyesp/

3. We can create another repository as firstproject and git URL will be as below

https://github.com/getyesp/ firstproject /

4. We can add project source code files here. However, the best practice is to add it from client view of repository on developer box as shown in part 2 below.

$ git clone <https://github.com/getyesp/getyesp.git>

Cloning into 'getyesp'...

warning: You appear to have cloned an empty repository.

$ cd getyesp/

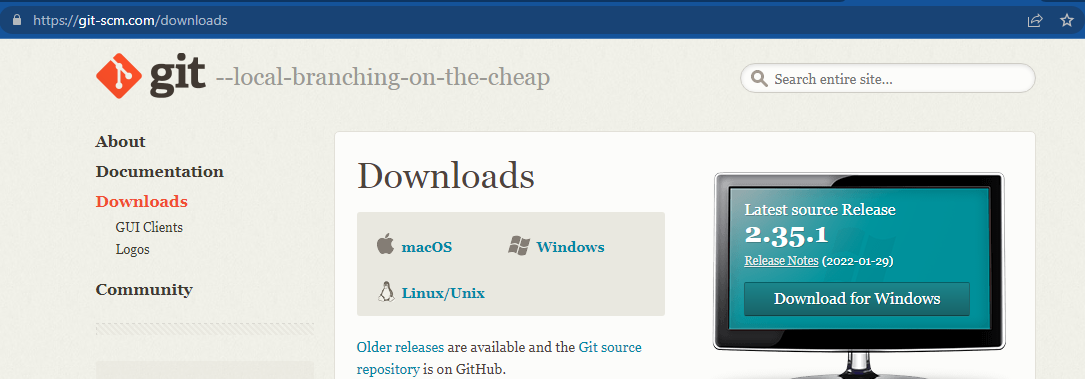
$ ls

/d/getyesp (main)

**Part 2 Client-side view or Developer view:**

**Developer View or Client-side view of Git**

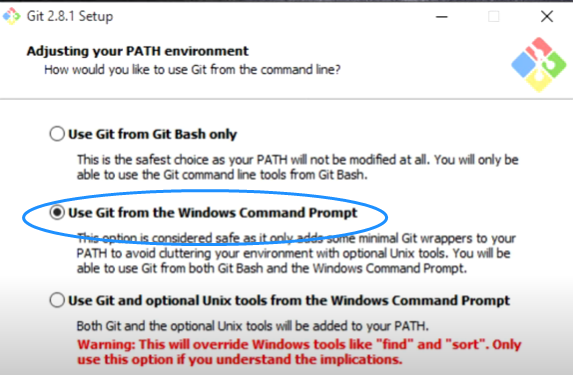
1. Go to URL: https://git-scm.com/downloads

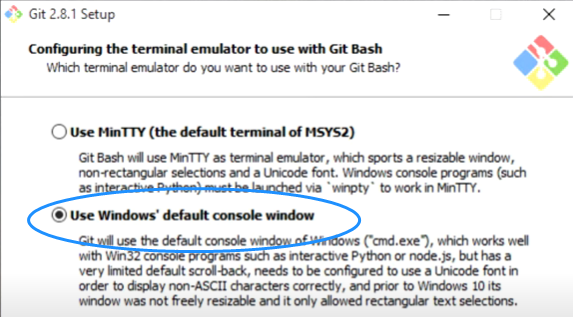


2. Click on Latest version of windows which is 2.35.1.

3. Select 64-bit version 

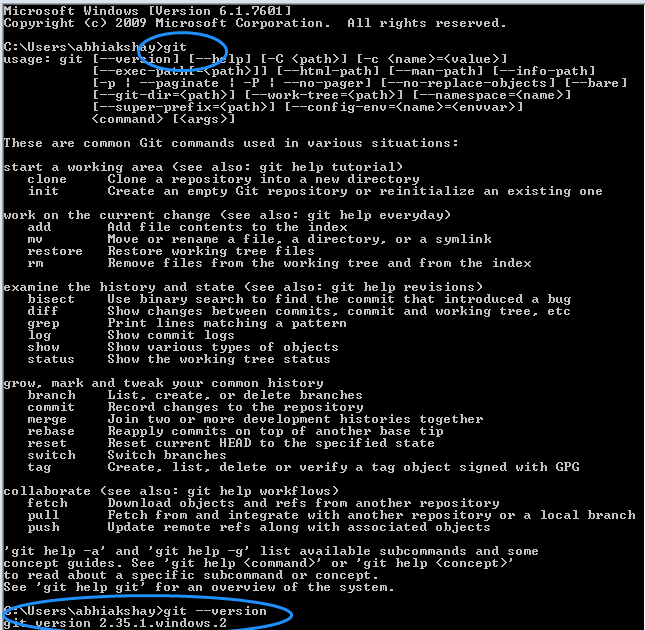
4. Double click on exe file downloaded. Go with default settings. However, it is recommended to go with minor changes during installation as shown below.



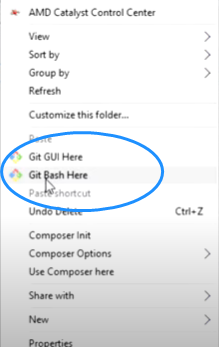


5. Once we click finish, check installation done or not using below commands

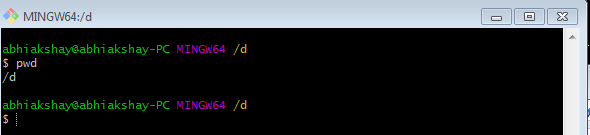
* git
* git --version



6. Go to any directory now and right-click and open git bash as shown below

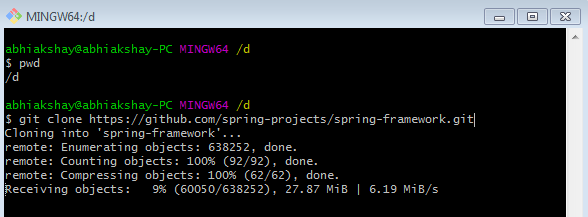


7. Git bash opens a command terminal where it simulates Linux kind of shell as shown below



8. Use git clone to clone required repository. For example, to clone spring use below command

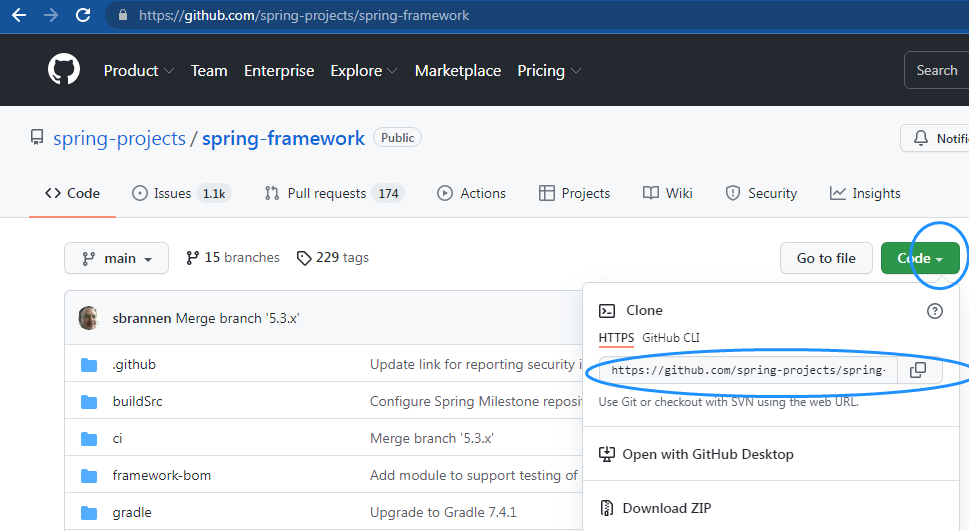
git clone https://github.com/spring-projects/spring-framework.git



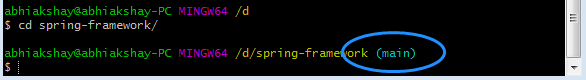
To find out repository name https://github.com/spring-projects/spring-framework.git

That was cloned above. We need to go to git server URL where spring is uploaded like Nexus repository.

For example, https://github.com/spring-projects/spring-framework



9. Once we clone repository, we can find the git projects by going to the directory



Here main indicates the branch name. We can create a feature branch for each developer as needed

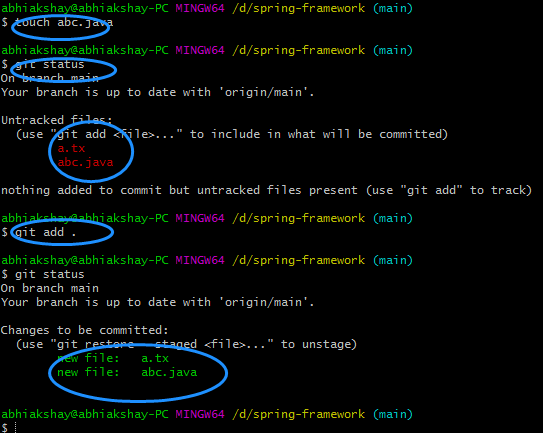
By default, cloned repository will point to main branch which is master branch. We can use git checkout abc to switch to feature branch by name abc.

Below are the operations you perform on the git branch.

touch abc.java -> to create a file by name abc.java

git add . - > to add all files to repository

git status - > to find the files that are added or modified or yet to be added to repository



git commit -m "initial files added"

git push -> to add files to remote repository

**Trouble shooting steps**

If we encounter issues like not able to recognize user, use below commands to

**Setting config**

git config --global user.name getyesp

git config --global user.name getyesp@yahoo.com

**Setting up Proxy for corporate n/w access limitations**

**Proxy**

git config --global http.https://domain.com.proxy http://proxyUsername:proxyPassword@proxy.server.com:port

git config --global http.https://domain.com.sslVerify false