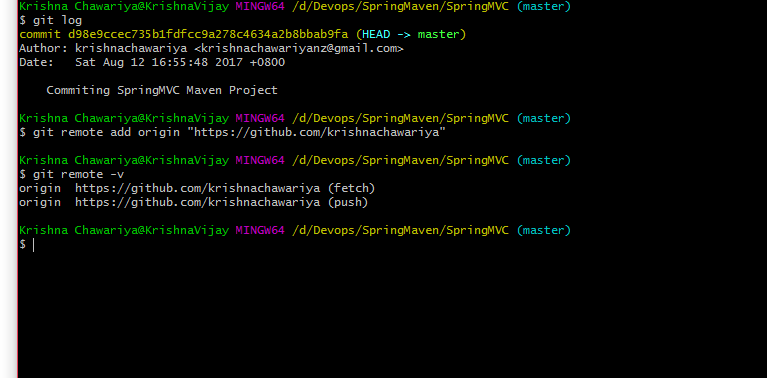
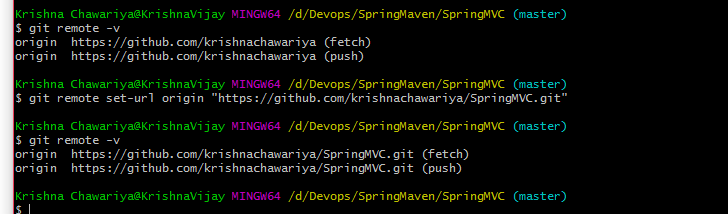
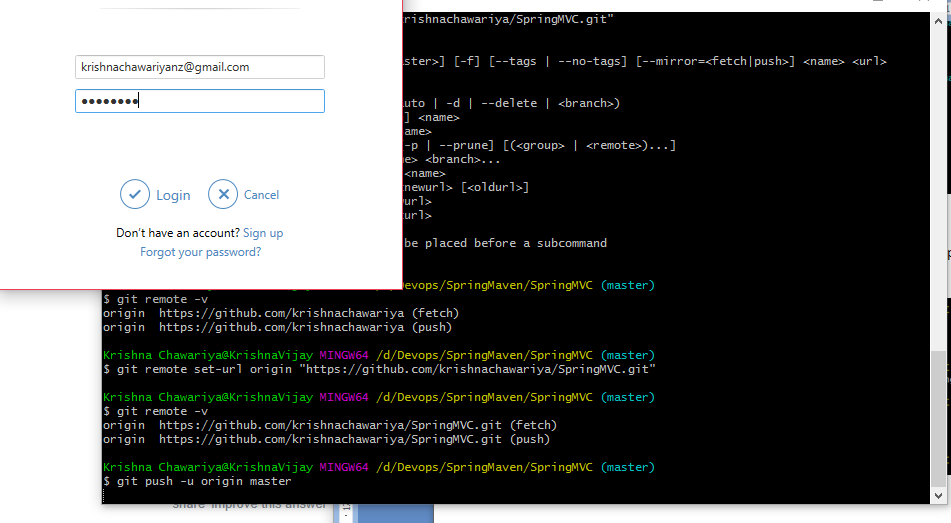
Add local repository to github:



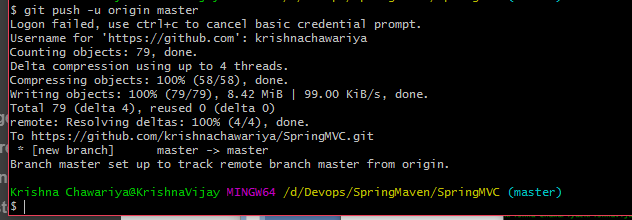
Map the local repo to remote repo

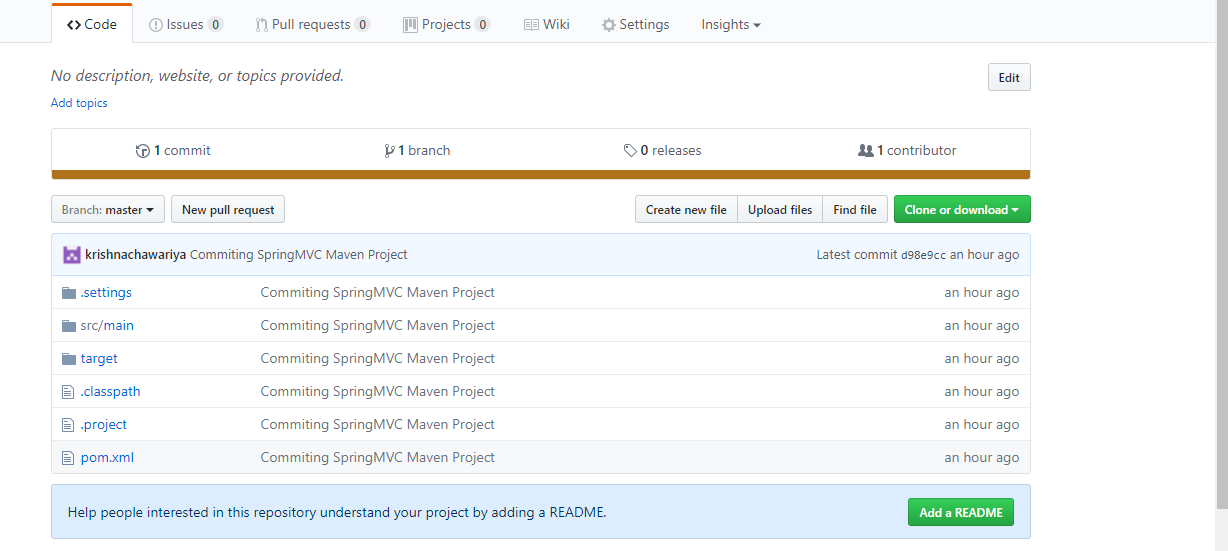
If we link our local repo by mistakenly to other remote repo, set the appropriate remote url using below command





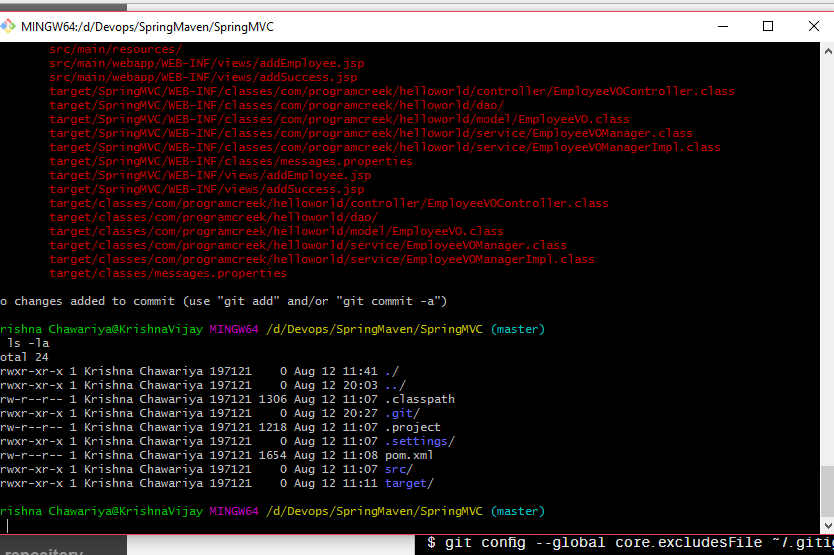
Once Push completed successfully

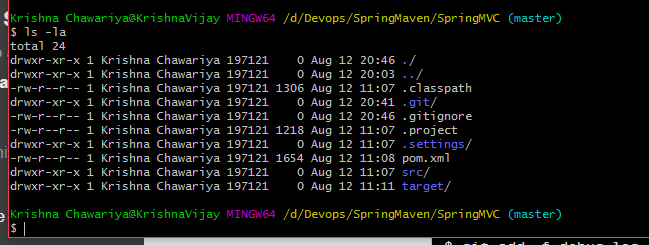


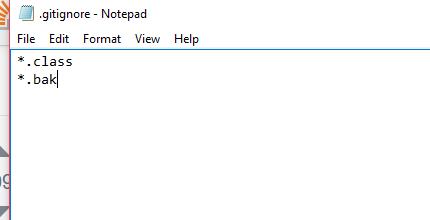


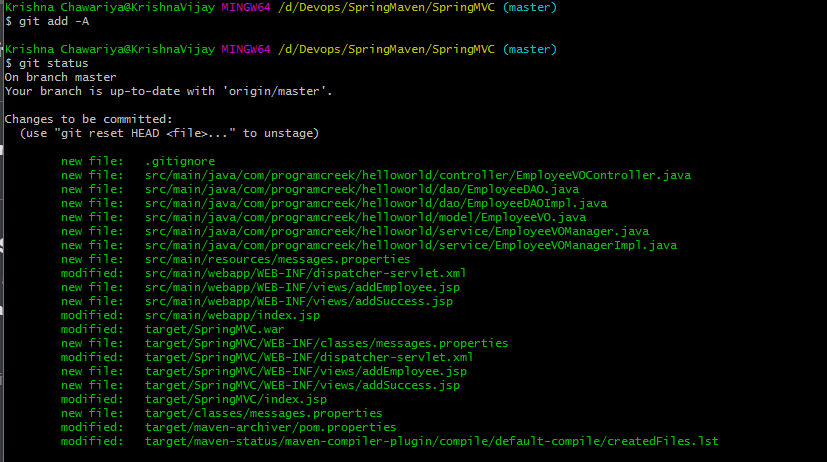
.gitIgnore

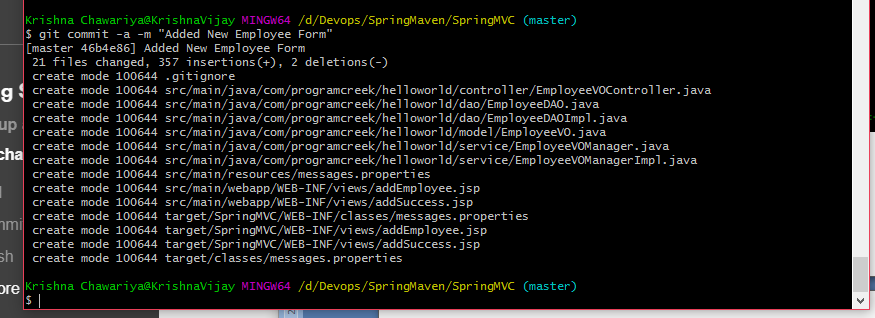
remove .class file

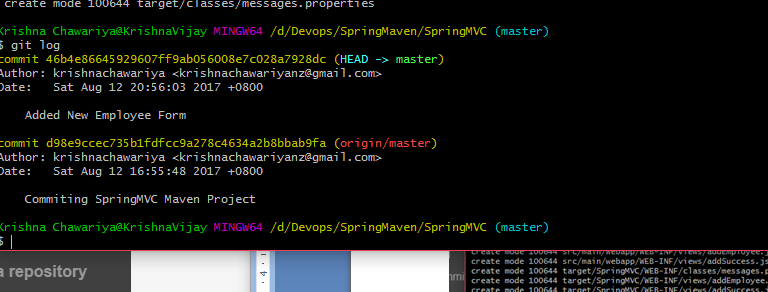




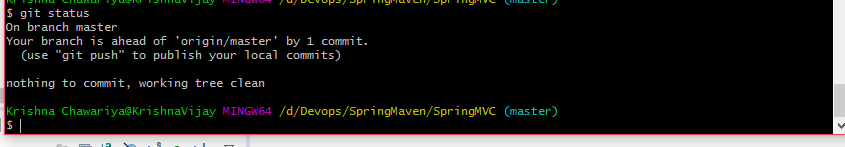


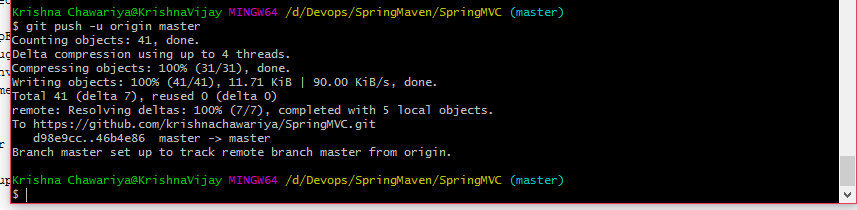


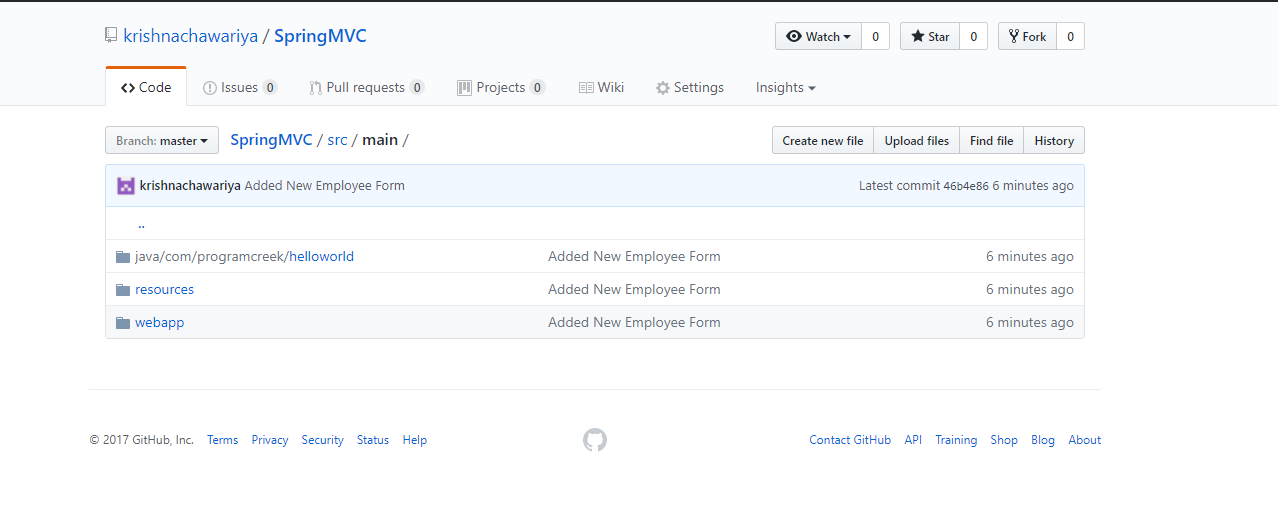




Before pushing changes to origin







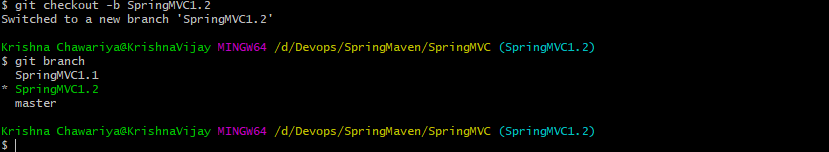
**Git Branching & Merging:**

**Created 2 branches under master in order to work parallel**

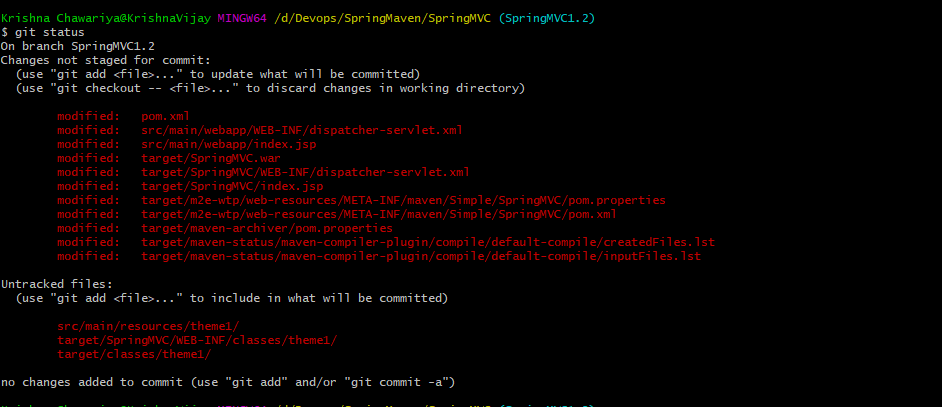
**syntax:**

**git branch <branchname>**

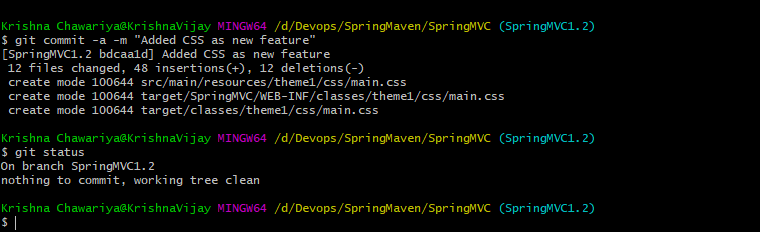
**git checkout -b <branchname>**

****

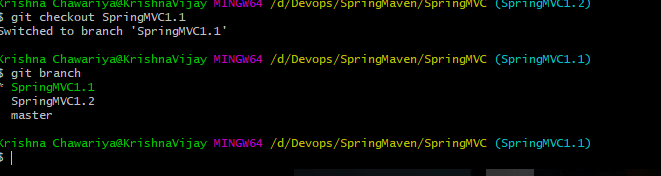
**1.2 contains modified and untracked files**

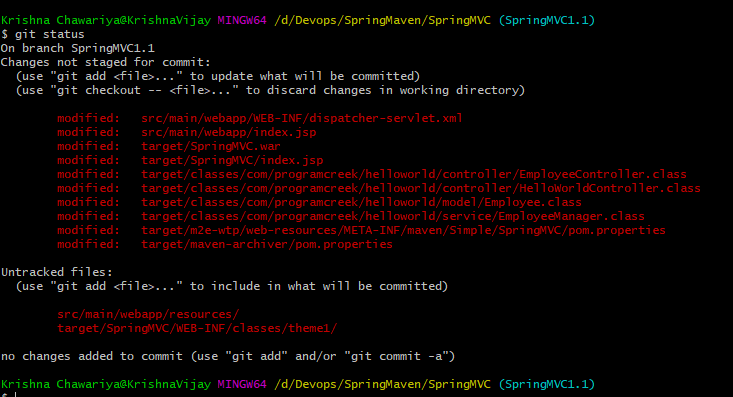
****

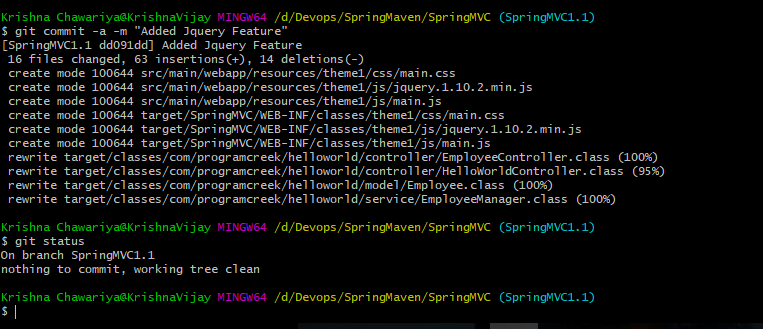
**Add and commit the files for 1.2 before switching to 1.1**

****

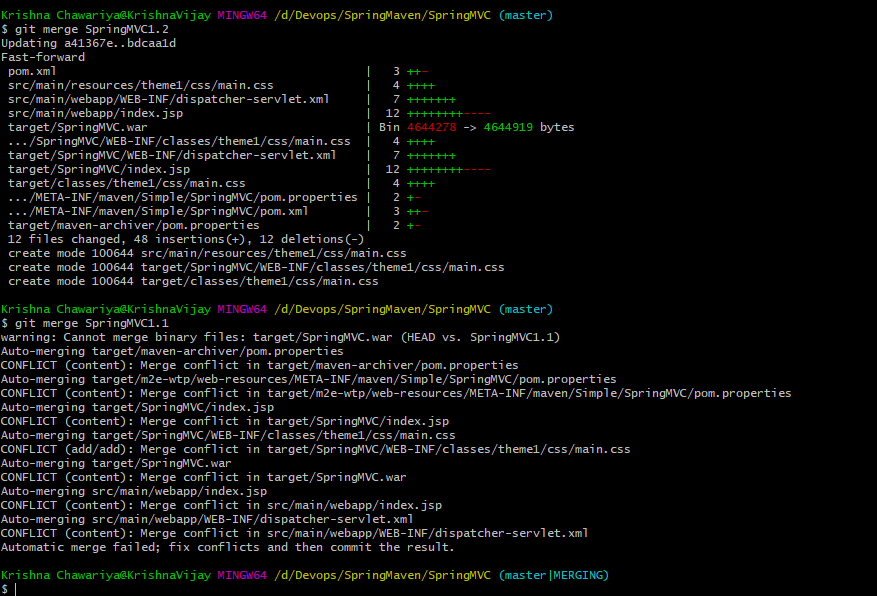
**Now switch to 1.1**

****

****

****

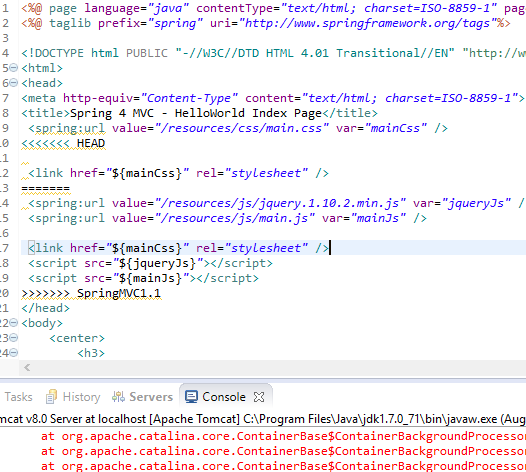
**Now checkout master, Merge 1.1 & 1.2 branches**

****

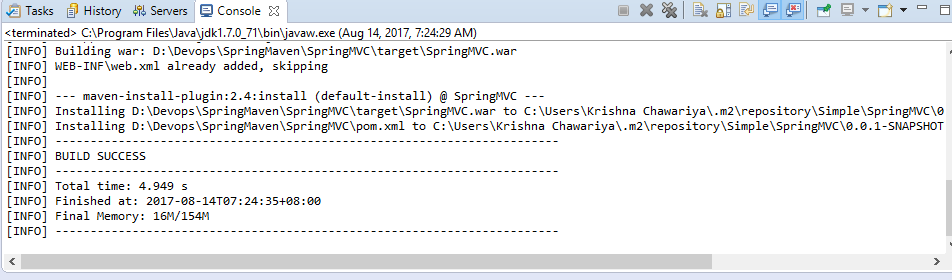
**Above screen shot, clearly shown the conflicts as i've performed the changes on common files**

**When you open the file in your text editor, you'll see the changes from the HEAD or base branch after the line <<<<<<< HEAD. Next, you'll see =======, which divides your changes from the changes in the other branch, followed by >>>>>>> BRANCH-NAME.**

**Decide if you want to keep only your branch's changes, keep only the other branch's changes, or make a brand new change, which may incorporate changes from both branches. Delete the conflict markers <<<<<<<, =======, >>>>>>> and make the changes you want in the final merge.**

****

**Ensure build is success after resolving conflicts**

****

****