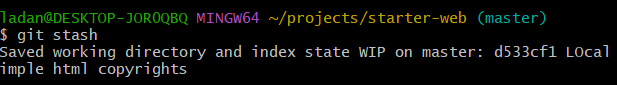
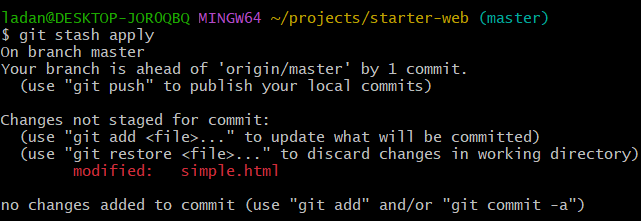
**Stashing**

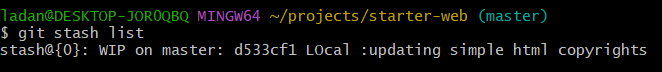
1. **Simple Stashing**

* Stash is used when we have simple.html modifies but not committed and urgently we need another file to committed. So, in this situation we need to save the changes of simple.html we use git stash command. **$git stash**



* Now change the file that was urgently needed and commit it.
* Now in order to get back the stash that we have done we need to apply our stash with command **$git stash apply**

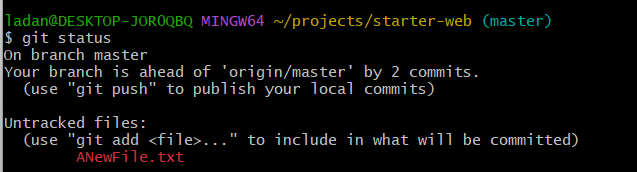
* Commit the simple.html and do command **$git stash list**



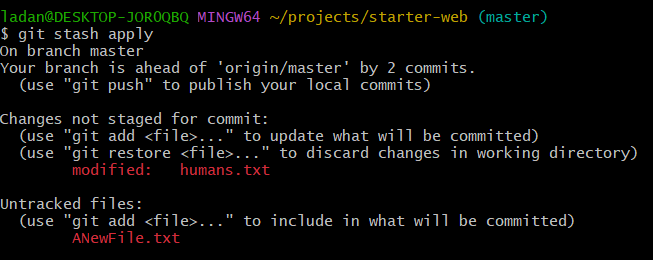
* Now we don’t need it anymore so use **$git stash drop**

1. **Stashing Untracked files and using pop**

* Git stash command only gets the tracked file it doesn’t get untracked files so if we create new file from bash then it will not be tracked first of all so it will not stash that file.



* Now apply stash to both untracked and tracked.



* Now drop the last stash by $git stash drop.
* One of the option is we can add new file to git staging area so that git will start tracking the file an issue will be resolved.
* If we don’t want to add that file and still want to stash it

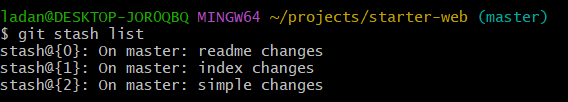
**$git stash -u**

* If we want to do apply and drop In one command **$git stash pop**

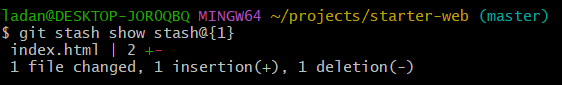
1. **Managing Multiple Stashes**

* When we use multiple stash we need to specify save and message in command in order to identify easily.

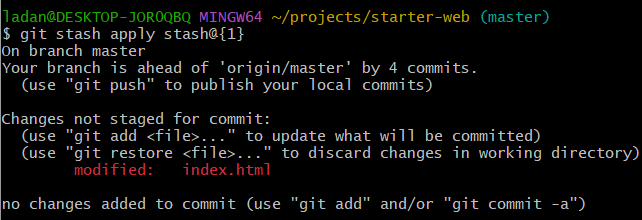
**$git stash save “index changes”**

****

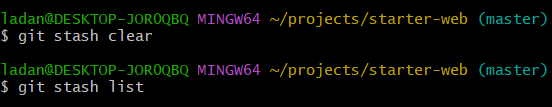
* Stash index is zero means readme was stashed at last.
* **$git stash show stash@{1}**

****

* To apply index changes command **$git stash apply stash@{1]**

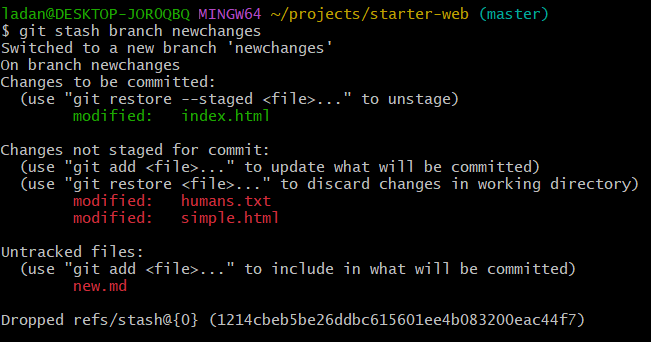
****

* Similarly, for drop and use **$git stash clear** for clearing all stash.



1. **Stashing into branch**

* Change the files human.txt, simple.html, index.html. and then add index.html to staging area.
* Add new file new.md
* After doing this we realize that this belongs to feature branch not master branch so to correct this is to use git stash.
* So, use $git stash -u to save files.
* **$git stash branch newchanges**

****

* Several things happened here first new branch is created and switched to it, second stash is applied and at end stash is dropped. So, stash list will be empty.
* Now remove the new file and add and commit the other files.
* Now we will integrate the changes in master branch.
* Now merge the newchanges branch $git merge newchanges by going to master branch.

1. **Section cleanup and push to GitHub**

* Pull and push the commands.