

# Module 3: Git, Jenkins and Maven Integration

---

Demo Document – 3

edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Demo – 3: Stashing, Rebasing, Reverting and Resetting

- To Create a stash of your current working directory  
Syntax: `git stash save 'message'`
- To list all the saved stashes  
Syntax: `git stash list`
- Stash list uses a stack structure to save the list

```
amr@amr-VirtualBox:~/Documents/demo$ git checkout master
Switched to branch 'master'
amr@amr-VirtualBox:~/Documents/demo$ git stash list
stash@{0}: On feat1: Saving changes made to the RepoFile
amr@amr-VirtualBox:~/Documents/demo$
```

- Saved stashes can be applied at anytime on any branch  
Syntax: `git stash apply <stash id>`
- After applying a stash it can still be accessed elsewhere because it remains in the stash

```
amr@amr-VirtualBox:~/Documents/demo$ git stash apply stash@{0}
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   RepoFile

no changes added to commit (use "git add" and/or "git commit -a")
amr@amr-VirtualBox:~/Documents/demo$
```

- Pop command can be used to apply the most recent stash and removing it from the stash stack  
Syntax: `git stash pop`

```
amr@amr-VirtualBox:~/Documents/demo$ git stash list
stash@{0}: On master: Updated new stash
stash@{1}: On feat1: Saving changes made to the RepoFile
amr@amr-VirtualBox:~/Documents/demo$ 
amr@amr-VirtualBox:~/Documents/demo$ git stash pop
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   RepoFile

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (4bbae06658e7435e16ea9a29f00c254526c1e788)
amr@amr-VirtualBox:~/Documents/demo$ git stash list
stash@{0}: On feat1: Saving changes made to the RepoFile
amr@amr-VirtualBox:~/Documents/demo$
```

- Stashes can be deleted from the stash list

Syntax: `git stash drop <stack id>`

```
amr@amr-VirtualBox:~/Documents/demo$ git stash drop stash@{0}
Dropped stash@{0} (6275777018dcd9bd801d02172bb45feb9538cebf)
amr@amr-VirtualBox:~/Documents/demo$
```

- The entire stack can also be deleted using one command

Syntax: `git stash clear`

```
amr@amr-VirtualBox:~/Documents/demo$ git stash clear
amr@amr-VirtualBox:~/Documents/demo$
```

Rebasing, reverting and resetting:

- To Rebase a branch

Syntax: `git rebase <rebase branch>`

```
amr@amr-VirtualBox:~/Documents/demo$ git rebase master
First, rewinding head to replay your work on top of it...
Fast-forwarded feat2 to master.
amr@amr-VirtualBox:~/Documents/demo$
```

- Revert or **undo** the changes made in the previous commit

- New commit is created without the changes made in the other commit  
Syntax: `git revert <commit id>`
- Old commit still resides in the history

```
amr@amr-VirtualBox:~/Documents/demo$ git revert HEAD
[feat2 a50d898] Revert "revert edit"
 1 file changed, 1 deletion(-)
amr@amr-VirtualBox:~/Documents/demo$
```

- Reset command can be used to undo changes at different levels
- Modifiers like `--hard`, `--soft` and `--mixed` can be used to decide the degree to which to reset  
Syntax: `git reset <modifier> <commit id>`

```
amr@amr-VirtualBox:~/Documents/demo$ git reset v1.5
Unstaged changes after reset:
M      RepoFile
amr@amr-VirtualBox:~/Documents/demo$
```