

# Guide to **git stash** command



@vikasrajputin

art by:  
*iqonic.design on iconscount*

# 1. Introduction

- The **git stash** command **saves our unfinished work** to a temporary location without having to commit
- Later, when we come back to the same work, we can resume our work from where we've left it





## 2. **Create** Stash

→ ***git stash*** or ***git stash push***

Stash all unstaged files.

→ ***git stash push <file-path>***

Stash only specified (unstaged)files.

→ ***git stash -m <stash-message>***

Stash all unstaged files with messages.



→ ***git stash --include-untracked***

Stash all untracked files

→ We can also use all the these  
Commands together as well

→ ***git stash <file-name> -m <message>  
--include-untracked***

Stashes the given file with all  
untracked files, with a message.





### 3. **View** Stash

→ ***git stash list***

List all stash

→ ***git stash show <stash-id>***

Show a list of all files in a stash

Use ***-p*** to see file differences,

Use ***--include-untracked*** to see all untracked files

Use ***--only-tracked*** to see all tracked files



## 4. **Use** Stash

### → ***git stash pop***

Applies the top stash in the list and removes it from the list

### → ***git stash pop <stash-id>***

Applies the given stash and removes it from the stash list

### → ***git stash apply <stash-id>***

Applies a given stash but does not remove it from the list





❤️ **Thanks** for reading !

For more content on  
**Java & Backend** Development,  
follow me on below handles



**Vikas Rajput**  
@vikasrajputin

