# GitHub account

**Github**

<https://github.com/login>

<https://github.com/amtmody/demo.git>

**My git account**

# Resources

git commands: https://www.atlassian.com/git/tutorials/comparing-workflows/

git flow: <http://danielkummer.github.io/git-flow-cheatsheet/>

git flow: <http://nvie.com/posts/a-successful-git-branching-model/>

# GITHUB Commands

## Creating a new repository

1. Create a repository in GitHub (https://help.github.com/articles/creating-a-new-repository/)
2. mkdir project
3. cd project
4. git init
5. git add . (or git add mavendemo-parent mavendemo-webapp my-app to add multiple folders)
6. git commit -m “new repository”
7. Switch to the remote repo or branch you want to commit these changes (see switching remote repository)
8. git push -u origin master

Example 1 (file add)

C:\Tech\Maven\Git\demo [master ≡]> echo 'test 123' > text.txt

C:\Tech\Maven\Git\demo [master ≡ +1 ~0 -0 !]> git add text.txt

C:\Tech\Maven\Git\demo [master ≡ +1 ~0 -0 ~]> git commit -m "added text.txt"

C:\Tech\Maven\Git\demo [master ↑]> git push origin master

Example 2 (directory add)

git init

git add test

git commit -m "All files message"

git push origin master

Example 3 (file change)

Changing a file (changed file pom – orig.xml)

git add "test\\my-app\\pom - orig.xml"

git commit -m "POM file"

git push origin master

## Cloning existing repository

git clone <https://github.com/username/your-repo.git>

E.g.:

C:\Users\amody\Documents\GitHub> cd C:\Tech\Maven\Git

C:\Tech\Maven\Git> git clone https://github.com/amtmody/demo.git

C:\Tech\Maven\Git> cd demo

C:\Tech\Maven\Git\demo [master ≡]> cat README.md

# demo

To add the cloned repository following creating a new repository section.

## Creating branch

git checkout -b feature-1

# you are now in a branch, you can edit and create new files

C:\Tech\Maven\Git\demo [master ≡]> git branch newfeature

C:\Tech\Maven\Git\demo [master ≡]> git checkout newfeature

* Switched to branch 'newfeature'

git add .

git commit -am “new feature”

C:\Tech\Maven\Git\demo [newfeature]> echo "new feature" > feature.txt 🡪 creating file

C:\Tech\Maven\Git\demo [newfeature +1 ~0 -0 !]> git add feature.txt

C:\Tech\Maven\Git\demo [newfeature +1 ~0 -0 ~]> git commit -m "added feature.txt"

C:\Tech\Maven\Git\demo [newfeature]> git push origin newfeature

### Merging branch to master

git checkout master

git merge feature-1

git push

C:\Tech\Maven\Git\demo [newfeature]> git checkout master

* Switched to branch 'master'

C:\Tech\Maven\Git\demo [master ≡]> git merge newfeature

C:\Tech\Maven\Git\demo [master ↑]> git push origin master

### List all branches

git branch -a

C:\Tech\Maven\Git\demo [newfeature]> git branch

master

\* newfeature

$ git branch -a

\* master

testFeature

remotes/origin/master

remotes/origin/testFeature

### Deleting branch

git branch -d feature-x 🡪 delete the branch from local

git push origin --delete feature-x 🡪 delete it from repository

### Switch branch

git checkout feature-x

Switch to master branch

git checkout master

E.g.: C:\Tech\Maven\Git\demo [newfeature]> git checkout master

## Listing Current Remote repositories

git remote -v

C:\Tech\Maven\Git\demo [master ≡]> git remote -v

origin https://github.com/amtmody/demo.git (fetch)

origin https://github.com/amtmody/demo.git (push)

## Switching remote repository

# in case your remote repository changes, or you want to switch from HTTPS->SSH or SSH->HTTPS

git remote remove origin

git remote add origin git@github.com:yourlogin/your-repo.git

e.g.:

git remote add origin [git@github.com:amtmody/demo.git](mailto:git@github.com:amtmody/demo.git)

git remote add origin git@github.com:amtmody/test.git

git remote add origin https://ghdp1.devops.iconectiv.com/NPC/npc\_uae\_temp\_bin.git

# Iconectiv Code Access

**Git Extension and IDE plugin**

Install Git client and IDE plugin:

1. install git client: https://code.google.com/p/gitextensions/

Note: select install MSysGit and KDiff3

select OpenSSH as SSH client

At configuring the line ending conversions page:

Select checkout as-is, commit Unix-style line endings

2. install git eclipse plugin

Post installation setup:

1. Create ssh key: ssh-keygen -t rsa

• Passphrase is not needed

• If you need access from multiple machines, just copy the id\_rsa and id\_rsa.pub files

2. send me your id\_rsa.pub file located in ~/.ssh directory (Please rename the pub to match your email, for example echang.pub)

3. Start gitextension program, open Tools -> Settings page, go to Git Config section and set name, email. mergetool (kdiff3) etc.

**Iconectiv Git repository structure:**

1. Project repository: git@icsnjnmsbuild1.ics.iconectiv.com:csc/repository-name, where repository-name includes:

csca-persist.git

csca-rest.git

csca-web.git

csca-cli.git

csca-common-rest.git

csca-core.git

csca-cscapi.git

csca-datacleaner.git

csca-email.git

csca-jobs.git

csca-netsuite-connector.git

csca-ns-import.git

csca-parent.git

csca-pdf-utils.git

csca-reports.git

csca-scheduler.git

csca-security.git

csca-servicedesk.git

csca-snmp.git

csca-utils.git

csca-wsdlgen.git

csca-development-docs.git

csca.git

2. Developer repository: git@icsnjnmsbuild1.ics.iconectiv.com:dev/your\_name/repo\_name

gitolite - multiaccess and multicontrol

Every developer can create their own repository

gitextensuib

OpenSSH for Git

gitK for history

mergetool

Pull = fetch + update

Sourcetree for grouping

# Troubleshooting

Adding id\_rsa and id\_rsa.pub

<https://help.github.com/articles/error-permission-denied-publickey/>

<http://abdelraoof.com/blog/2017/10/24/fixing-permission-denied-publickey-github/>

I am not sure how many modules are there in NMS Thailand repository. But for each module, by using the latest revision before the specific date, the code can be checked out to that point.

E.g.

$ git rev-list -n 1 --before="2018-06-27 13:37" origin/master

699762324e437e79cc284edbf441f1cea0ae1057

$git checkout 699762324e437e79cc284edbf441f1cea0ae1057

Here are the Git commands which are being covered:

* **git config**
* **git init**
* **git clone**
* **git add**
* **git commit**
* **git diff**
* **git reset**
* **git status**
* **git rm**
* **git log**
* **git show**
* **git tag**
* **git branch**
* **git checkout**
* **git merge**
* **git remote**
* **git push**
* **git pull**
* **git stash**

So, let's get started now!!

## **Git Commands Examples**

### git config

Usage: git config –global user.name “[name]”

Usage: git config –global user.email “[email address]”

This command sets the author name and email address respectively to be used with your commits.

Git Config Command - Git Commands - Edureka

### git init

Usage: git init [repository name]

This command is used to start a new repository.

GitInit Command - Git Commands - Edureka

### git clone

Usage: git clone [url]

This command is used to obtain a repository from an existing URL.



### git add

Usage: git add [file]

This command adds a file to the staging area.

Git Add Command - Git Commands - Edureka

Usage: git add \*

This command adds one or more to the staging area.

Git Add Command - Git Commands - Edureka

### git commit

Usage: git commit -m “[ Type in the commit message]”

This command records or snapshots the file permanently in the version history.



Usage: git commit -a

This command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.

Git Commit Command - Git Commands - Edureka

### git diff

Usage: git diff

This command shows the file differences which are not yet staged.



 Usage: git diff –staged

This command shows the differences between the files in the staging area and the latest version present.



Usage: git diff [first branch] [second branch]

This command shows the differences between the two branches mentioned.



### git reset

Usage: git reset [file]

This command unstages the file, but it preserves the file contents.



Usage: git reset [commit]

This command undoes all the commits after the specified commit and preserves the changes locally.

Git Reset Command - Git Commands - Edureka

Usage: git reset –hard [commit]  This command discards all history and goes back to the specified commit.

Git Reset Command - Git Commands - Edureka

### git status

Usage: git status

This command lists all the files that have to be committed.



### git rm

Usage: git rm [file]

This command deletes the file from your working directory and stages the deletion.

Git Rm Command - Git Commands - Edureka

### git log

Usage: git log

This command is used to list the version history for the current branch.



Usage: git log –follow[file]

This command lists version history for a file, including the renaming of files also.



### git show

Usage: git show [commit]

This command shows the metadata and content changes of the specified commit.



### git tag

Usage: git tag [commitID]

This command is used to give tags to the specified commit.



### git branch

Usage: git branch

This command lists all the local branches in the current repository.

Git Branch Command - Git Commands - Edureka

Usage: git branch [branch name]

This command creates a new branch.

Git Branch Command - Git Commands - Edureka

Usage: git branch -d [branch name]

This command deletes the feature branch.

Git Branch Command - Git Commands - Edureka

### git checkout

Usage: git checkout [branch name]

This command is used to switch from one branch to another.

Git Checkout Command - Git Commands - Edureka

Usage: git checkout -b [branch name]

This command creates a new branch and also switches to it.

Git Checkout Command - Git Commands - Edureka

### git merge

Usage: git merge [branch name]

This command merges the specified branch’s history into the current branch.

Git Merge Command - Git Commands - Edureka

### git remote

Usage: git remote add [variable name] [Remote Server Link]

This command is used to connect your local repository to the remote server.

Git Remote Command - Git Commands - Edureka

### git push

Usage: git push [variable name] master

This command sends the committed changes of master branch to your remote repository.



Usage: git push [variable name] [branch]

This command sends the branch commits to your remote repository.



Usage: git push –all [variable name]

This command pushes all branches to your remote repository.



Usage: git push [variable name] :[branch name]

This command deletes a branch on your remote repository.



### git pull

Usage: git pull [Repository Link]

This command fetches and merges changes on the remote server to your working directory.



### git stash

Usage: git stash save

This command temporarily stores all the modified tracked files.

Git Stash Command - Git Commands - Edureka

Usage: git stash pop

This command restores the most recently stashed files.



Usage: git stash list

This command lists all stashed changesets.

Git Stash Command - Git Commands - Edureka

Usage: git stash drop

This command discards the most recently stashed changeset.

Git Stash Command - Git Commands - Edureka

Want to learn more about git commands? Here is a [Git Tutorial](https://www.edureka.co/blog/git-tutorial/" \t "_blank) to get you started. Alternatively, you can take a top-down approach and start with this [DevOps Tutorial.](https://www.edureka.co/blog/devops-tutorial)