

## Sunil Kumar Gunasekaran

Quality automation Professional Research Labs

# Git Maven Evaluation – Documentation

### GIT BRANCH RULES

A Kickstarting feature card creation -

Start your feature/jiracard from master

Note down passwords in excel file and checkin files after removing passwords

B Pull latest changes from stash -

```
git pull
```

```
git checkout branch name
```

```
git merge master
```

C Stage and commit changes -

```
git add -A
```

```
git commit -m "Changes to feature JIRA"
```

```
git push -u origin "feature/jiracard"
```

D Inter branch file transfer -

Keep feature branch file folder

sub folder path folder names

keep exact files

switch between branches using VCS or Git Bash

E Clone register nad package

```
git clone ssh://git@devops-tools.abccorporation.com/occ/room-ui.git
```

```
mvn clean install -Dmaven.test.skip=true
```

```
git config --global user.name "your name here"
```

```
git config --global user.email "your email here"
```

To Clone project: `git clone ssh://git@devops-tools.abccorporation.com/occ/room-ui.git`

GIT PULL REQUEST GREP - GIT PULL REQUEST GREP - GIT PULL REQUEST GREP

Keep the pull request staged and Diff it

Keeping unknown files in remote from tampering your PULL REQUEST

`git rm -f <file location>` This gets the file deleted and staged

`git commit -m "Remote file deleted"`

`git push -u origin "feature/feature1"`

Start from master checkout feature/feature1 for every new feature

Place the <path to file> in .gitignore and place .gitignore in gitignore to avoid unnecessary files from getting detected in GIT tool



swiftsunil / August 2, 2016 /

Sunil Kumar Gunasekaran / Proudly powered by WordPress