CITI AUTOMATION CODE COMMIT WORKFLOW

***1.*** Always get latest code update from upstream (Remote upstream repository) before starting for coding for a day. **Origin** is your **fork**. You always **fetch** from **upstrem** but **push** to **origin**.

Command **: *git fetch upstream***

***2.*** Get latest code update and merge them into local master repository :

Command **: *git rebase -i upstream/master***

**Note :** *We will get rebase successful msg in case of success.*

**3.** If successfully rebased then start coding into your local master.

**Note** : *In case of successful git rebase, we will skip below step* ***4****,****5*** *and* ***6****.*

***4.*** If not successfully rebased (because of extra code coded in same files). Save our extra coded code with **git stash .**

Command **: *git stash save***

***5.*** Now again rebase upstream master repository code with same command:

Command **: *git rebase -i upstream/master***

***6.*** Now we need to get back our stashed(stored) code into local repository with **git pop command**:

Command**: *git stash pop stash@{0}***

**Note:** *stash@{0} means latest stored stash code.*

***7.*** To commit the code we will use ***git gui*,** which will provide us UI for **staged** and **unstaged** view and **entire changed code directory structure** so that picture will be more clear to us.

***8.*** To commit code we will move desired files from **unstaged** to **staged** area

Note: Through CMD line we will use: ***git add file\_name***to do same but here we could achieve this with just **click on desired files**

***9.*** Once you are done with adding desired files to commit into **staging area ,**you are ready to commit the code. Just set appropriate comment within comment section and click on **commit.**

Note : You can achieve same thing through CMD with : ***git commit -m "msg\_as\_commit\_text"***

***10.*** Now push your code to your **origin master(forked repo)** by clicking on push button and then selecting **proper origin master** branch to push local code.

Note : With CMD use: ***git push origin master***  to push into forked repository.

***11.*** Now make a Pull Request to **upstream/master** to push your code into upstream master repository.

Note : Use bitbucket server UI to create **PR** to **upstream/master.**

***12.***After code approval by **Team lead/Admin**, he/she will runs **merges** changes into master branch using **bitbucket UI interface.**

**Upstream Repo**

Git fetch upstream

Git rebase -i upstream/master

Pull Request

**Forked/Origin repo**

**Local repo**

Git push origin master