//====================================================================================

GIT FLOW (Github full workflow)

**By: HM Emrul**

Developer

Skype: emrul87

//====================================================================================

**BRANCHES OF GIT FLOW CONTROL**

1. Master branch
2. Develop branch
3. Feature branch
4. Release branch
5. Hotfix branch
6. Bugfix branch

PRODUCTION RELATIONS BETWEEN DOMAIN AND GITHUB

1. Master branch will be linked with the public domain for public use. like: [www.emptycoder.com](http://www.abc.com)
2. Develop branch will linked with demo domain for testing purpose Like [www.demo.emptycoder.com](http://www.demo.emptycoder.com)

**STEPS TO FOLLOW TO COMPLETE THE FULL GIT LIFECYCLE.**

1. Create a new repository on you github account
2. Create a new branch develop on current repository
3. Go to settings options to change the repository name main to master and change default branch master to develop.
4. Go to your local pc and clone the master branch of the repository where you want to save as local using the following command.
   1. Git clone git-link [Enter]
5. Initialize git flow to start a git flow chain by using the following commadn.
   1. Git flow init [Enter]
   2. Yes [Enter]
   3. Yes [Enter]
   4. Yes [Enter until finish the input]
6. Now see you are in develop branch automatically if not in develop use the following command
   1. Git branch checkout develop [Enter]
7. Push all data to your develop branch for your updated code. Use the following command.
   1. Git push all [Enter]
   2. if this update your code to git develop branch then ok else use following command
      1. Git add .
      2. Git commit -m ‘First commit to develop branch from local develop branch’
      3. Git push origin develop [Enter]

FEATURE BRANCH

1. Create new feature branch by following command
   1. Git flow feature start new-feature [Enter]
   2. Now you are in/using feature/new-feature branch automatically if not use following command:
      1. Git branch checkout feature/new-feature
2. Write your code what you need for your project and push to new-feature branch by following command:
   1. Git add .
   2. Git commit -m ‘Completed or 25% completed of my new feature’
   3. Git push origin feature/new-feature [Enter]
3. If your assigned feature is completed you think, now you merge your new-feature branch with develop branch by creating a pull request.
4. Go to your github account and select your current repository.
5. Click on your pull request. Create a new pull request and follow below steps:
   1. Select develop branch as base and new-feature as compare branch
   2. Marge and squash branch.
   3. Delete new-feature branch
6. Go to your local repository
7. Switch branch feature/new-feature to develop and delete feature branch by following command:
   1. Git branch checkout develop
   2. Git branch -D feature/new-feature [Enter]
8. Copy updated data from git repository by pull request by following command:
   1. Git pull origin develop [Enter]
9. If you need to work on another feature follow the step: [8 to 15]

RELEASE BRANCH

1. Make a release point or release name by using following command:
   1. Git flow release start 0.0.1 [Enter]
   2. Git push [Enter] (for see the command for set release branch)
   3. Git push - - set-upstream origin release/0.0.1
2. Go to github repository and click on pull request to merge master with release branch by following steps:
   1. Base as release/0.0.1 branch and compare with master branch
   2. Only marge [Not marge and squash]
   3. Not delete master branch
3. Now your repository have three branch as bellow
   1. master
   2. develop
   3. release/0.0.1

HOTFIX BRANCH

1. If you think you need an update on the release branch for minor changes you may use hotfix branch to solve this issue.
2. Select master branch as base branch of hotfix branch by using following command:
   1. Git branch checkout develop [Enter]
3. Create a new hotfix branch from master branch by using following command:
   1. Git flow hotfix start new-hotfix [Enter]
4. Now you are in hotfix/new-hotfix branch if not use the following command:
   1. Git branch checkout hotfix/new-hotfix [Enter] (for use new-hotfix branch)
5. Write your code as you need. If if you think you need to update your local code to git than make a push using following command:
   1. Git add .
   2. Git commit - m ‘message of your push request on hotfix/new-hotfix branch’ [Enter]
   3. Git push origin hotfix/new-hotfix [Enter]
6. If you think your hotfix/new-hotfix is completed successfully you may merge and squash the hotfix branch with master branch by following steps:
   1. Goto your github repository and make a pull request and follow following steps:
      1. Master branch as base and hotfix/new-hotfix as compare
      2. Marge only and not delete hotfix/new-hotfix
      3. Complete pull
   2. Again make a pull request and follow following steps:
      1. Develop as base and hotfix/new-hotfix as compare
      2. Marge and squash then delete the hotfix/new-hotfix branch
      3. Complete pull request
7. Create a new release branch by following step [ 17 to 19]

BUGFIX BRANCH

1. If you found any bug on testing time and you need the update of code on the develop branch you may use bugfix branch to solve that issue.
2. Select develop branch by using following command:
   1. Git branch checkout develop [Enter]
3. Create a new hotfix branch by using following command:
   1. Git flow bugfix start new-bugfix [Enter]
4. Now you are in hotfix/new-hotfix branch if not, use the following command:
   1. Git branch checkout bugfix/new-bugfix [Enter] (for use new-bugfix branch)
5. Write your code to fix the bug. Then update your local code to github by make a push using following command:
   1. Git add .
   2. Git commit - m ‘message of your push request on bugfix/new-bugfix branch’ [Enter]
   3. Git push origin bugfix/new-bugfix [Enter]
6. If you think your bugfix/new-bugfix is completed successfully you may merge and squash the bugfix branch with develop branch by following steps:
   1. Goto your github repository and make a pull request and follow following steps:
      1. Develop branch as base and bugfix/new-bugfix as compare
      2. Marge only and not delete bugfix/new-bugfix
      3. Complete pull
   2. Again make a pull request and follow following steps:
      1. feature/new-feature branch (if have) as base and bugfix/new-bugfix as compare
      2. Marge and squash then delete the bugfix/new-bugfix branch
      3. Complete pull request

===================================================================

END OF GIT FLOW CONTROL INSTARACTION

#######################################################################