EPI USE Interns Technology Stack Setup Tutorial

Overview:

1.Github

2.Wildfly

3.JEE

4.Hibernate

5.Postgresql

6.Nginx

7.Webpak

1.Github

1. go to <https://github.com/> and create a new account.

2. go to <http://gitforwindows.org/> and download git bash

Git commands:

NB!!! ALWAYS PULL BEFORE YOU PUSH. This ensures you have the latest version and you are not working on an older version of a project.

|  |
| --- |
| DOWNLOAD GIT BASH |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| git commands: |
|  |

|  |
| --- |
| make sure pwd is desktop cd ../Desktop |
|  |

|  |
| --- |
| step 1 : git clone https://github.com/DawiePritchard/Brogramming.git |
|  |

|  |
| --- |
| step 2 : cd <repository> brogramming |
|  |

|  |
| --- |
| step 3 : pwd |
|  |

|  |
| --- |
| step 4: git fetch(all remote branches) |
|  |

|  |
| --- |
| step 5: git checkout <branch> (development(Switched to development branch)) |
|  |

|  |
| --- |
| step 6: git branch(list of local branches) |
|  |

|  |
| --- |
| step 7: git branch <branch name> (create branch) |
|  |

|  |
| --- |
| step 8: touch dawie.txt |
|  |

|  |
| --- |
| step 9: ls |
|  |

|  |
| --- |
| step 10: git status |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Comment on step 11 |
|  |

|  |
| --- |
| #step 11: git add . (green means staging area) always before commit |
|  |

|  |
| --- |
| #git add . is not such a good idea since it adds any and all new files, that may include temporaries and unintended files as well. |
|  |

|  |
| --- |
| #It is better to use 'git add -u', which only looks at existing files that are being tracked, which means only already tracked file's changes will be commited which is a good thing. Now, if you do decide git must track a new file, you can explicitly add it using 'git add thefile'. After running 'git add thefile', whenever you modify or delete it and call 'git add -u', git will stage that as part of the commit for you. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| step 12: git commit -m "<commit message>" (initial commit) |
|  |

|  |
| --- |
| step 13: important note if you changed stuff on a branch then git wont let you switch until you have committed |
|  |

|  |
| --- |
| #comment on step 13: unless you stash the changes, then you can change branches without having to commit anything. |
|  |

|  |
| --- |
| step 14: git push |
|  |

|  |
| --- |
| #comment on step 14: As a matter of taste, one could be more explicit in the way you push changes. For example, if experimental has changed and you want to push changes to the remote master branch you can call 'git push origin experimental', where origin indicates the remote branch. This higher granularity is usefull since you are always aware of what branch you intend to push changes, thereby avoiding accidental changes to master when that was not intended. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| IMPORTANT : ALWAYS PULL BEFORE YOU PUSH TO REMOTE |
|  |

|  |
| --- |
| step 15: git pull |
|  |

|  |
| --- |
| #comment on step 15: 'git pull' has a similar caveat of granularity as step 14. "See https://stackoverflow.com/questions/15797183/differences-between-git-pull-commands-when-pulling-from-origin" for more information on this. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| How merge works: |
|  |

|  |
| --- |
| switch to branch where you want to merge in(test stuff into dev so dev is branch) |
|  |

git merge testBranch