Frank's – Final Project Set-up

Once you have your repos and branches setup, you should start thinking about the project.

1. **One team member** should:
   a. Create the repo for the project in GitHub.
   b. Open a GitBash window
   c. Navigate to the folder where you have other GitHub repos
      DO NOT USE A FOLDER THAT ALREADY HAS A REPO IN IT.
         if you are not sure if it has a repo in in it:
            1. **Navigate to the folder you want to use for the final project repo**
            2. **Issue the command:** `git status`
            3. **If you see the message:** `fatal: not a git repository (or any of the parent directories)` **you are safe to clone the assessment repo into that folder**
   d. Clone the repo down to the folder you have chosen on your local machine.
   e. Copy the `Server-Starter-Project` to your final project repo folder.
   f. Push the final repo folder just created to GitHub. (`git add`, `git commit`, `git push`)
   g. Copy the GitHub repo URL.
   h. Send the GitHub URL to other team member(s).

2. **Other team members** (not the one that created the repo and pushed the project):
   a. Open a GitBash window
   b. Navigate to the folder where you have other GitHub repos.
      DO NOT USE A FOLDER THAT ALREADY HAS A REPO IN IT.
      (Use the instructions above if you are not sure if it already contains a repo or not)
   c. Copy the URL given to you by the team member that created the final project repo.
   d. Clone the final project repo down to your machine using the URL given to you.
   e. Issue the command '`ls`' to be sure you have successfully cloned the repo.
   f. Open the `Server-Starter-Project` in IntelliJ and be sure you have all the files pushed by the creator of the project.
   g. Exit IntelliJ

Frank's – Final Project Set-up

3. If you would like to use git branching in your project follow these steps.

    Use of git branches of optional; <u>Git Branching Tutorial</u>

    All team members:
    - i.   Go back to your GitBash window (open another one).
    - ii. Navigate to the final project folder (if you are not already there).
    - iii. Issue the command "`git branch`" to see all current branches of the repo. You should just see one called "`main`" (or "`master`" depending on how the repo was created).
    - iv. Create a new branch for your work: `git checkout -b` *branch-name*
    - v.   Save the branch to GitHub: `git push origin branch-name`
    - vi. Issue the command: `git branch`

        You should see your new branch listed with an "`*`" next to it
        and any other branches for the repo.
    - vii. Open the final project `Server-Starter-Code` in IntelliJ.
    - viii. Look in the lower right corner. You should see your branch name as the one you are working in.
        (if not: click the arrow next to the branch name; you should be able to select your branch)
    - ix. You will be working and saving your changes in your branch.

4. To save your changes from your branch to GitHub:
    - a. Verify you are in your branch: `git branch` (look for the name with the "`*`" next to it)
        (if you are not in branch you want to be in: `git checkout` *branch-you-want-to-be-in*)
    - b. `git add -A`
         `git commit -m'meaningful-comment-about-what-you-did'`
         `git push origin branch-name`

5. When you are ready to merge your changes back to base code:
    - a.   Navigate to the folder with the Final Project repo on your machine.
    - b. Be sure all the changes from your branch have been pushed.
    - c. Navigate to the "`main`" (or "`master`") branch: `git checkout main` (or `master`)
    - d. Issue the command: `git merge` *branch-name*
                        (*branch-name* name of the branch containing code to be merged)
    - e.   Verify all changes have been pushed: `git status`

    If you receive merge conflict errors contact your Instructor or Teaching Assistant to help you resolve them unless you are ABSOLUTLEY sure you can do it yourself.

    - f.   Notify your teammates you have merged your changes so they "add, commit and push" their changes and then pull yours: `git pull origin main (or master)`

Frank's – Final Project Set-up

Don't forget, you can always go to GitHub using the URL to verify your changes have been saved.

If you don't know the GitHub URL: `git remote -v` will show it to you when you are in the repo folder on your local machine.

Please do not hesitate to contact your instructor or Teaching Assistant if your unsure of any of this, or anything else, before you attempt it. It is much easier to avoid a problem by asking for help than trying to fix one.