Tutorial 15 – Poor Man’s Guide to GitHub Workflow

Contents

[1 Introduction 1](#_Toc112583574)

[2 Person 1 – Establish Initial Codebase 2](#_Toc112583575)

[3 Person 1 – Create a Work Branch & Begin Coding 3](#_Toc112583576)

[4 Person 2 – Create a Work Branch & Begin Coding 4](#_Toc112583577)

[5 Submission 5](#_Toc112583578)

# Introduction

As a last resort, assuming you cannot figure out how to establish a workflow where you utilize a local Git repository and push and pull from GitHub, using either the command line or an IDE, you can use the workflow in this tutorial for your project. With this approach, instead of pulling code from your branch, you will download a zip file with the code from your branch. And, instead of pushing your code back to GitHub, you’ll drag individual files. You will not be maintaining your code locally, with a Git repo, against best practices.

**To use this approach effectively, every time you make a committable change to a file (write a method, modify a method, write the shell for a class, *etc.*) you must drag the file into GitHub, type a commit message, and then commit.** In other words, you are going against best practices if you write a lot of code (what should be a number of commits) and then drag to GitHub and commit. These statements are also true if you are using a workflow that utilizes a local Git repo; however, it is easier to commit using an IDE.

# Person 1 – Establish Initial Codebase

1. **Establish local code base.** Do the following:
2. Create a repo on GitHub (mine is *poor-man*)
3. On the Code page for the repo, choose: *uploading an existing file*



1. Drag a Java file into the site (I used *BasketballPlayer.java* from Tutorial 14)



1. Type a commit message and choose: Commit changes.

|  |  |
| --- | --- |
|  |  |

# Person 1 – Create a Work Branch & Begin Coding

1. Create a branch in GitHub (mine is named *dave*)
2. Choose: Code, Download ZIP



1. Unzip and load code into your favorite IDE. For this tutorial, I’m just going to use Notepad
2. Make some small change to the *BasketballPlayer* class and then save.
3. In GitHub, choose: Code, Add file, Upload files



1. Drag your local *BasketballPlayer.java* into the dialog, type a commit message, and choose: Commit changes.



1. In GitHub, choose: Code, Add file, Upload files
2. Drag another Java file into the dialog (I used *BasketballPlayerTest.java*), type a commit message, and choose: Commit changes.
3. Do a pull request and merge your branch back into the *main* branch.
4. Make a screen shot showing the result, something like:



# Person 2 – Create a Work Branch & Begin Coding

The process is exactly the same as for [Person 1](#_Person_1_–).

# Submission

1. **Do the following:**
2. Place the screen shot in the *HW VCS* document in the appropriate place.
3. The image should easily readable without zooming in or out.