Java Basics

Exercise 2: Basic Walkthrough

SOFTWARE GUILD

Credits and Copyright

Copyright notices

The Learning House
427 S. 4th Street #300
Louisville KY 40202

Lesson 4: Git Quick Start

Java Cohort

Exercise 2: Basic Walkthrough

Introduction

In this exercise we will walk through the basics of how to add a file to a repository, save the file as part of the repository and push the change to Bitbucket. This walkthrough can be used for many of the pre-work exercises as they require you to add a file to your repository and push them to Bitbucket.

These exercises will use the swcguild repository. If you wish to follow this walkthrough with another repository, this exercise assumes the repository is created on Bitbucket and is cloned to the local file system. If you have not completed these steps please refer to details on creating a repository and also how to clone the repository locally.

Steps

1 Create a file in the desired directory. This directory should be inside the directory you created for the repository during the git clone command. In many of the pre-work exercises you will be asked to create a specific directory structure. Do so directly in the repository directory on your local machine.

Ex. ..\swcguild\Pre-Work\Certificates\HTML\Codeacademy

In the above “swcguild” is the repository directory and each additional folder was created within that directory. The file can then be placed in the desired directory.

Running a git status at this point will show that the directory is added and untracked.

```
git status
On branch master Initial commit
Untracked files:
(use "git add <file>..." to include in what will be committed)

Pre-Work/

nothing added to commit but untracked files present (use "git
add" to track)
```

2 Now that the file is in the directory we need to tell git to track the file and stage the change. This can be accomplished using our git add command.

```
git add –all
```

Running git status shows the file being tracked but not yet committed. Now we see the actual file added to the repository and not just the directory.

```
git status
On branch master Initial commit
Changes to be committed:
(use "git rm --cached <file>..." to unstage)

new file:  Pre-
Work/Certificates/HTML/Codeacademy/certificate.jpg
```

3 Now we can commit the change using git commit. Remember to add the -m "<message>" for good measure and better description in the history.

```
git commit -m "saved my HTML CodeAcademy certificate"
[master (root-commit) 869cb64] saved my HTML CodeAcademy
certificate
1 file changed, 0 insertions(+), 0 deletions(-) create mode
100644 Pre-
Work/Certificates/HTML/Codeacademy/certificate.jpg
```

Now git status will show there are no changes to be committed but it does hint that we may need to push the changes to the remote repository.

```
git status
On branch master
Your branch is based on 'origin/master', but the upstream is
gone. (use "git branch --unset-upstream" to fixup)

nothing to commit, working directory clean
```

4 All that is left is to push the changes to Bitbucket using the git push command.

```
$ git push -u origin master
Password for 'https://vjpudelski@bitbucket.org': Counting
objects: 3, done.
Writing objects: 100% (3/3), 253 bytes | 0 bytes/s, done. Total 3
(delta 0), reused 0 (delta 0)
To   https://vjpudelski@bitbucket.org/vjpudelski/gitpractice.git
* [new branch]  master -> master
Branch master set up to track remote branch master from origin.
```

And now git status has nothing to say except we are up to date.

```
git status
On branch master
Your branch is up-to-date with 'origin/master'. nothing to
commit, working directory clean
```

This process can be followed for all of the pre-work. Just make sure you start with the swcguild repository and add the files to the correct directory. You do not need to run git status with every command but certainly can to get exposure to what is happening at each step. Git status was used here to help illustrate the steps of the exercises.