

Java Basics

Exercise 3: Merging



SOFTWARE-GUILD

Credits and Copyright

Copyright notices

Copyright © 2016 by The Learning House.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed "Attention: Permissions Coordinator," at the address below.

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

Lesson 4: Git Quick Start

Java Cohort

Exercise 3: Merging

Introduction

In this exercise we will walk through the development process used by two developers. For the sake of keeping things clear let's call those developers Wise and Ward. In this example Ward will start a new repository and begin modifying documents. Early in the process Ward will ask Wise to join the development and make his contributions. As the two developers work together you will see the conflict this causes in the code and the resulting strain on their ability to check files in. The goal of this exercise is to introduce the idea of encountering a conflict committing files and how to merge your changes once you have encounters such a situation. This is a common issue in scenarios where more than one developer will be working in the same solution at the same time.

Note: This example is a precursor to the paired programming you are likely to see in the cohort. We will keep the process simple using only a single branch and instead of code files we will be modify text files. Regardless this is good practice and the topics here will be expanded on in the cohort.



Task 1:

Ward is ready to begin development and is going to create a new repository on git hub. We will create a new repository to keep this separate from the rest of the pre-work exercises.

Steps

- 1 Create a repository on Bitbucket named "GitPractice" to store the files that are used in this practice exercise. Follow the steps in Exercise 1 including the code provided by Bitbucket to get this repository setup. This would include creating a folder called "Ward" in the repos folder. Calling the folder Ward rather than GitPractice so that we can create a second copy to simulate the other developer.
 - ```
git init
git remote add origin
https://<user>@bitbucket.org/<user>/gitpractice.git
```

## Task 2:

---

**With the repository created, Ward is ready to create the first file and check in the initial file to the repository and push those changes to Bitbucket.**

---

### Steps

- 1 Create a text file in the Ward directory and name it file1.txt
- 2 Add some text to the file using your favorite text editor (perhaps Notepad or Notepad++ for you Windows users, while my Linux friends can use gEdit).

**Example Text:** Ward: This is a text file

- 3 Save the file.
- 4 Next Run the git status command. You may need to change the directory within in git bash if it is not already on the Ward directory.

```
git status
```

The output of git status shows that the file is untracked and not a part of our repository.  
On branch master

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
file1.txt
```

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

- 5 Use the git add command to add the file to the repository and stage the changes for commit.

```
git add -all
```

Running git status again, we can see that the file is tracked and ready to be committed.

```
$ git status
```

```
On branch master Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage) new file: file1.txt
```

- 6 Use git commit command to save the change to the repository.

```
$ git commit -m "initial commit - adding file1"
```

```
[master (root-commit) cd375ca] initial commit - adding file1
1 file changed, 1 insertion(+) create mode 100644 file1.txt
```

In this instance, git status shows that we have nothing to commit and isn't yet tracking on our remote repository. Therefore, we need to push to the remote and set that we are going to track.

```
$ git status
```

```
On branch master
```

```
nothing to commit, working directory clean
```

- 7 To push the changes to Bitbucket we can use the git push command and authenticate to Bitbucket using our username and password that was used to create the account earlier.

```
$ 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.
```

Our changes will now be visible in Bitbucket from the website.

## Source

|                                                                                               |                                                                                     |                                             |
|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------|
|  master ▾  |  | GitPractice /                               |
|  file1.txt | 25 B                                                                                | 4 minutes ago initial commit - adding file1 |

### Task 3:

---

Now Wise is now going to join the development effort. We will simulate this by creating a separate directory for his repository and modify the file already in the directory. Once modified we will jump to Ward's file and modify that. Let's see what happens when he tries to push the file.

---

#### Steps

- 1 This time we will run the git clone command and make sure you also set a directory name and we will call the new directory "Wise".

---

**Note:** make sure the directory is set to the \_repos directory



---

```
git clone
https://vjpuudelski@bitbucket.org/vjpuudelski/gitpractice.git wise
```

- 2 Open the file1.txt file and modify the file and save.

**Example Text:** Wise: Yes, this is a text file

Once the changes are saved the git status command will show the uncommitted file again.

```
$ git status
```

```
On branch master
```

```
Your branch is up-to-date with 'origin/master'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed) (use
"git checkout -- <file>..." to discard changes in working
directory)
```

```
modified: file1.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

- 3 Use the commit command to commit the change to the file.

```
$ git commit -a -m "added a new line to the file" [master
```

```
4dc020b] added a new line to the file
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```

Notice the "-a" on the commit and the lack of an add command call. This is because "-a" takes the place of add. Adding all the files that should be staged to the commit.

#### 4 Push the changes to Bitbucket

```
$ git push origin master
Password for 'https://vjpudelski@bitbucket.org': Counting
objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 0 bytes/s, done. Total 3
(delta 0), reused 0 (delta 0)
To https://vjpudelski@bitbucket.org/vjpudelski/gitpractice.git
cd375ca..4dc020b master -> master
```

---

**JUMP OVER TO WARD'S FILE NOW**



---

**Make sure you now have Ward's file open. The first thing to notice is that Ward's file doesn't have the line we added to Wise's. That is ok. Let's proceed anyway...**

---

#### 5 Modify file1.txt and save the edit as we have done in the past.

**Example Text:** Ward: adding a new line of text

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'. Changes not
staged for commit:
(use "git add <file>..." to update what will be committed) (use
"git checkout -- <file>..." to discard changes in working
directory)

modified: file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

#### 6 Use the commit command again and again we will add the -a to ensure the change is first staged and then committed.

```
$ git commit -am "second line from Ward added" [master 93a6b42]
second line from Ward added
1 file changed, 2 insertions(+), 1 deletion(-)
```

## 7 Now let's push the change up to Bitbucket

```
$ git push origin master
Password for 'https://vjpudelski@bitbucket.org':
To https://vjpudelski@bitbucket.org/vjpudelski/gitpractice.git
! [rejected] master -> master (fetch first) error: failed to
push some refs to
'https://vjpudelski@bitbucket.org/vjpudelski/gitpractice.git'
hint: Updates were rejected because the remote contains work that
you do
hint: not have locally. This is usually caused by another
repository pushing
hint: to the same ref. You may want to first integrate the remote
changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for
details.
```

---

UH OH! It was rejected!!!





## Task 4:

---

Remember when Wise made modifications, committed them and then pushed them up to Bitbucket. Well, Ward didn't pull those changes and now the file is out of sync with the remote repository and cannot be committed until Ward pulls these changes and merges them with the changes he wishes to make. To do this we need to pull those changes that are in the remote repository on Bitbucket. To get those changes we will start by pulling them down, merging changes and then recommitting and pushing to Bitbucket.


---

### Steps

- 8 Use the pull command to pull down the changes from Bitbucket. This will actually try and overwrite your files to what is currently on Bitbucket. You may be thinking, "but, we already made changes... won't we lose those?". Watch what happens...

```
$ git pull origin
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done. remote: Total 3
(delta 0), reused 0 (delta 0) Unpacking objects: 100% (3/3),
done.
From https://bitbucket.org/vjpudelski/gitpractice
cd375ca..4dc020b master -> origin/master
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

---

**CONFLICT!** things just keep getting worse, right? WRONG... This is ok and a normal part of the development cycle. Don't worry and watch how to resolve this conflict. 

---

- 9 Open the file1.txt again and the file will look something like this:

**Ward: this is a text file**

**<<<<<< HEAD**

**Ward: adding a new line of text**

**=====**

**Wise: Yes, This is a text file**

**>>>>>> 4dc020bc7d9c985cf74778aa64512f0f98c10364**

10 Modify the file to be something more like the below and save the file.

**Ward: this is a text file**

**Ward: adding a new line of text Wise: Yes, This is a text file**

After saving the file you can go to git bash and run git status

```
$ git status
```

```
On branch master
```

```
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commit each, respectively. (use "git
pull" to merge the remote branch into yours)
```

```
You have unmerged paths.
```

```
(fix conflicts and run "git commit")
```

```
Unmerged paths:
```

```
(use "git add <file>..." to mark resolution) both modified:
```

```
file1.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

11 Now you can run commit using the "-a" to stage the changes again.

```
$ git commit -am "fixed the merge conflict" [master ad2564d]
```

```
fixed the merge conflict
```

12 Now use git push again and push those changes up.

```
$ git push origin master
```

```
Password for 'https://vjpudelski@bitbucket.org': Counting
objects: 6, done.
```

```
Delta compression using up to 2 threads.
```

```
Compressing objects: 100% (4/4), done.
```


```
Writing objects: 100% (6/6), 584 bytes | 0 bytes/s, done. Total 6
(delta 1), reused 0 (delta 0)
```


```
To https://vjpudelski@bitbucket.org/vjpudelski/gitpractice.git
4dc020b..ad2564d master -> master
```

This exercise was an example of doing a merge. You can practice by making modifications and jumping between the two directories on your machine to get the hang of how this works. You will notice that your repository does create a fork and then merges together. This is because the merge creates an extra commit to bring the two branches (master and origin/master) in sync.

To view this fork you can use the git log command as mentioned above. You can also review the repository in bitbucket and see this change. On the repository page go to Commits page on the left hand side. It should appear like the following.

# Commits

 All branches ▾

|                                                                                   | Author                                                                                            | Commit                                 | Message                       |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|----------------------------------------|-------------------------------|
|  |  Victor Pudelski | <a href="#">ad2564d</a> <span>M</span> | fixed the merge conflict      |
|                                                                                   |  Victor Pudelski | <a href="#">93a6b42</a>                | second line from Ward added   |
|                                                                                   |  Victor Pudelski | <a href="#">4dc020b</a>                | added a new line to the file  |
|                                                                                   |  Victor Pudelski | <a href="#">cd375ca</a>                | initial commit - adding file1 |

```
$ git log --oneline --graph --decorate
* ad2564d (HEAD -> master, origin/master) fixed the merge conflict
| \
| * 4dc020b added a new line to the file
* | 93a6b42 second line from Ward added
| /
* cd375ca initial commit - adding file1
```

During the cohort we will look at other options to resolve these types of issues including commands such as rebase and what the differences are between rebase and merge.