InsideSherpa

JPMorgan Chase Software Engineering Virtual Experience

Software engineering task : making a patch file

Disclaimer

- This guide is only for those who did the setup locally on their machines.
- For those using REPL (*from module 1*), please use this guide instead

Prerequisites

- Set up should have been done. This means,your server and client applications should have been running with no problems without introducing any changes to the code yet. You can verify this if you get a similar result to any of the following slides that include a picture of the server and client app running together

- You should have made the necessary changes to the client script to make the application output correct values

Intro: What is a patch file

- You should know by now that git is a way for developers to manage code in a project especially if there’s other developers collaborating in that project too.

- A git **patch** file is just a file that you can apply to a repository to get the changes / modifications / additions another developer did on his / her machine onto your local machine. This isn’t the only way to do that ofcourse but this is a viable method for a head/lead developer to check your code first before merging it into the repository’s main/master branch.

How to make a patch file

- Fire up a terminal, enter the repository via the terminal you opened (*via the **cd** <repo_name_here> aka change directory command*) and do the following commands (*one line, one command*)

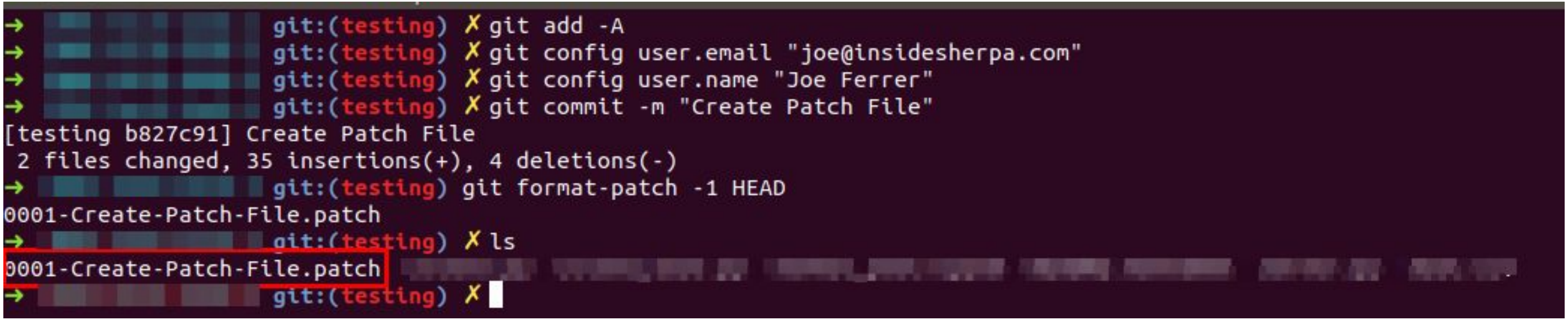
```
git add -A
git config user.email "<your_email_address>"
git config user.name "<your_name>"
git commit -m 'Create Patch File'
git format-patch -1 HEAD
```

note: make sure you use your personal email address and your real name

How to make a patch file

The final command, i.e. `git format-patch -1 HEAD`, should produce the .patch file you’d want to submit to complete this module. It will be located in the directory where you executed the command.

How to make a patch file



```
➜  [redacted] git:(testing) ✗ git add -A
➜  [redacted] git:(testing) ✗ git config user.email "joe@insidesherpa.com"
➜  [redacted] git:(testing) ✗ git config user.name "Joe Ferrer"
➜  [redacted] git:(testing) ✗ git commit -m "Create Patch File"
[testing b827c91] Create Patch File
 2 files changed, 35 insertions(+), 4 deletions(-)
➜  [redacted] git:(testing) git format-patch -1 HEAD
0001-Create-Patch-File.patch
➜  [redacted] git:(testing) ✗ ls
0001-Create-Patch-File.patch  [redacted]
➜  [redacted] git:(testing) ✗ █
```

- That’s how it would look like if you executed the commands properly
- *note: **ls** will only work for Linux and Mac environments. For Windows, use **dir** to list the files in the current directory so you can verify that you created the patch file*

And you’re done!

- Submit the patch file on the right module page