

# Journey from Monolithic to Modern CFML Development - Chapter 2: GitHub and VS Code

## [Journey from Monolithic to Modern CFML Development - Chapter 2: GitHub and VS Code](#)

Posted by [Andy Peterson](#)  
17 Apr 2020 10:30 AM

The goal of this series is to get team members familiar with using GitHub to collaborate on projects. In our camp, the vast majority of development consists of a 1:1 ratio of project to developer. However, I've tasked everyone on the team with learning this new approach, which is very convenient when having to justify, identify, and showcase the benefits of maintaining version control and other controls that keep our codebase safe on GitHub. So it is important to learn GitHub regardless of whether you are the star on your team or a lone star on no team.

In this chapter, we will create a branch of the Charity project and code directly in this branch using VS Code to interact with the repository.

Let me point out that up to and through this step, and even perhaps throughout this entire tutorial series, the tools used in this project are free. Neither my team nor I pay a dime for Ortus Products, Lucee or even Adobe CF Developer Edition, and VS Code. And even GitHub has a free version. And yes, could develop, and even deploy, with a free version of a relational database my MySQL. This fact alone is inspiring in that only the hardware costs would be required from a benefactor in order to teach low income kids how to code.

## **Step 2: GitHub and VS Code: Cloning a branch of the Charity project**

If you are following along, you should have a local Charity folder created from Chapter 1. Congrats! For us to work locally and to collaborate on the project, we will leverage GitHub as our repository and VS Code as the editor. By the end of this step, you will be using VS Code to pull down the master as well as a clone of a branch that you create.

There are basically two ways to work on your own versions of a code repository: forks and branches. Today we are going to learn about branches, but there is some configuration to get out of the way first, along with understanding the workflow. I am by no means an expert in either, but hopefully have found a direction moving forward.

Here is a high level overview of the following steps, from left to right, to eventually push your own version of code - your branch - to a repository – via a pull request that the code repo owner merges:

## Using VS Code

Following along in [this excellent video](#), you will have two new copies of the Charity project on your system: the master and a branch, as managed by GitHub and honored by VS Code. The master will be pulled using the first 9 steps. In step 10, I would like for you to clone your branch.

You will need to tailor these instructions for our project. As you follow the video, pay attention to these notes below.

**0:39** Step 1 : Install git on your system. This will show you whether you have git on your system, and if you do not, how to install it

**2:46** Step 2 : Create account on github - <https://github.com/> if you do not already

**3:27** Step 3 : Create a repository on github & copy url path to .git folder – Copy .git file <https://github.com/AndyPeterson/CharityProject.git>

**4:25** Step 4 : Goto VS Code and open project/folder note : make sure git is enabled from settings

**5:54** Step 5 : Goto source control section & click on git icon

6:52 Step 6 : Give commit message & Commit the changes – note that you probably won't have changes that need to be committed – let me know if it shows that you have U – uncommitted changes.

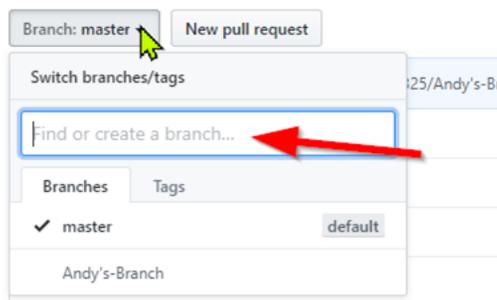
8:00 Step 7 : Add remote repo (github repo) – here: <https://github.com/AndyPeterson/CharityProject.git>

9:15 Step 8 : Push committed changes to github repo

You will now push to remote. Notice on 10:30 the reference to master branch bottom left hand corner.

10:03 Step 9 : Check changes in your GitHub Repository.

13:05 How to clone from GitHub – This is specifically to sync up your Charity branch. Now, I think you will want to clone your branch. So, create a branch first – call it JimmysBranch or whatever. and then clone your branch into Charity-JimmysBranch folder on your local.



14:50 Remove project from Git If you face any issues.

So now, you should have cloned your very own branch on your local. Please comment below on what issues you ran into. Thanks!

\*We use GitHub's private repositories, which comes with a fee.

Tags: [ColdBox](#), [ColdFusion](#), [ContentBox](#), [GitHub](#), [VSCode](#)  
[0 Comments](#)

---

Copyright © Andy Peterson. All rights reserved.

Powered by ContentBox v4.2.1+737