

Github: Branch, Merge, Revert and Rollback

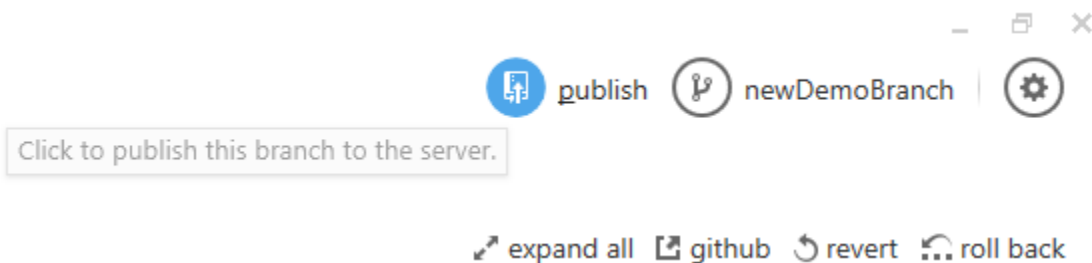
First, let's create a new branch. A new branch will give us a complete copy of the master branch so that we can make changes without altering the master branch. Select the highlighted branch button.



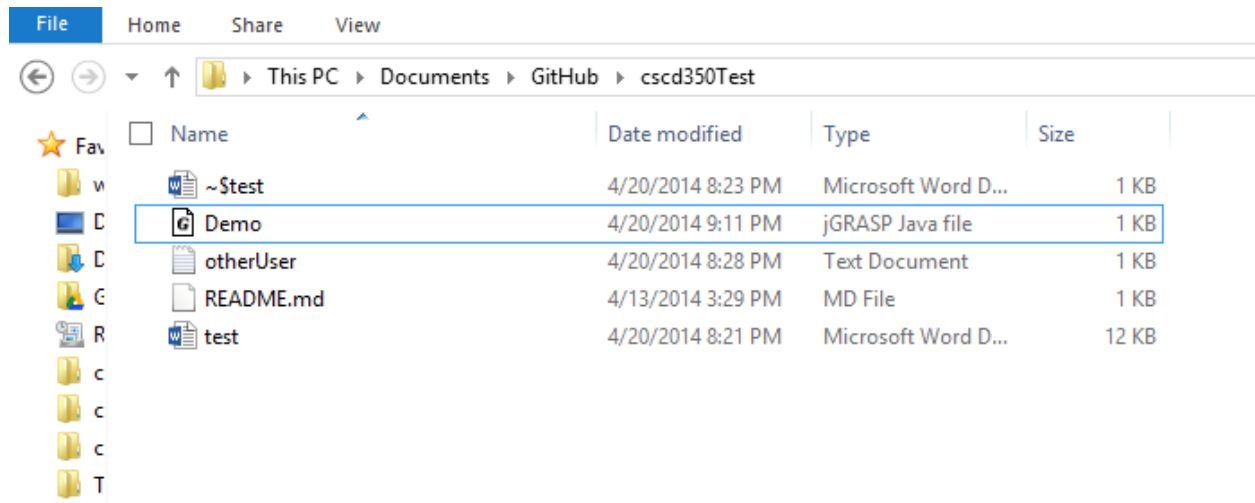
Enter a name for your new branch.

A screenshot of the 'Branches' dialog box in the GitHub interface. The dialog has a title 'Branches' and a subtitle 'manage / merge' with a branch icon. Below the title is a text input field containing 'newDemoBranch' and a close button (X). At the bottom of the dialog, there is a button that says '+ create branch: newDemoBranch'.

Next, click the publish button.



Now you have created a new branch with a complete copy of the current version. You will notice in your folder that there does not appear to be a new branch...

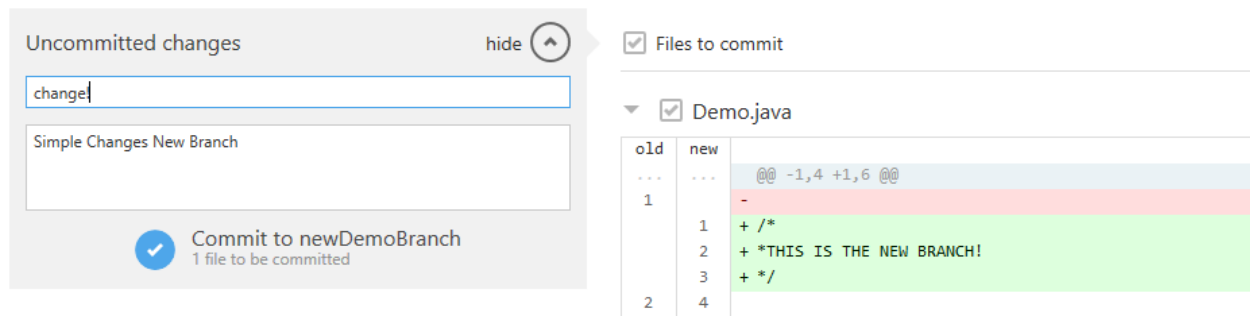


So this is how it works... When you are on a certain branch (indicated by the highlighted button), that is the branch that is currently represented in your Github folder. Any changes made to something in the folder at that time will be made to the current branch only. Let's go ahead and make a couple of changes to a simple java program. Here is the changes in the new branch.

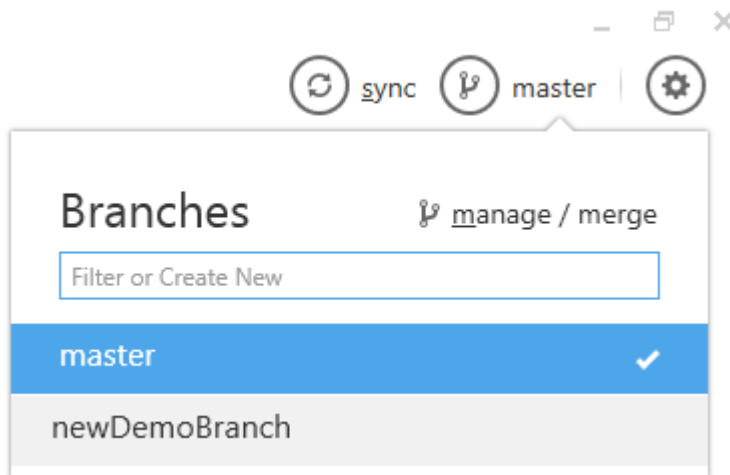
```
/*
*THIS IS THE NEW BRANCH!
*/

public class Demo
{
    public static void main(String [] args)
    {
        System.out.println("This program does nothing!");
    }
}
```

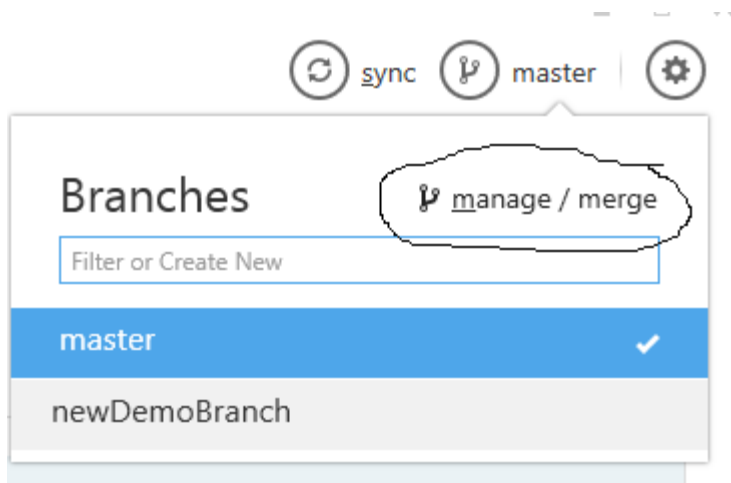
Now go ahead and commit these changes and sync.



Now switch back to the master branch. Select the master branch from this button.



Now, let's merge these two java files! Select the branch button and click the circled manage/merge button.



Now go ahead and drag and drop the new branch and master branch to the bottom of the screen, then click the merge button and sync.

Merge



Now if you open the java file you should see that they are now merged!

Now let's revert that change. Highlight the change on the left hand side and then select the revert option. Then sync.

The screenshot shows the code editor interface. At the top, a status bar indicates 'No uncommitted changes'. The main area shows a diff for 'simple change' by 'Mightygnome' (commit 375e1a9). The diff shows a change in 'Demo.java' where line 1 was changed from '...' to '-1,4 +1,6 @@'. On the left, the 'History' panel shows the 'simple change' commit. On the right, there are buttons for 'collapse all', 'github', 'revert', and 'roll back'. A tooltip above the 'revert' button says 'create a new commit that reverts the changes in this commit'.

Now if you open the java file, it should be back to how it was before the last commit.

```
public class Demo
{
    public static void main(String [] args)
    {
        System.out.println("This program does nothing!");
    }
}
```

Rollback works the same as revert although it leaves the changes in your working copy. This allows you to fix small mistakes and then re-commit.

Now that we are done with the branch we can delete it. Go back into manage/merge and simply click the garbage can and it is now gone.

The screenshot shows the branch management bar at the bottom of the editor. It has two tabs: 'newDemoBranch' and 'Mightygnome simple change'. The 'newDemoBranch' tab is active. To the right of the tabs is a trash can icon, a plus icon, and a minus icon. To the far right is a button labeled 'Unpublish'. A tooltip below the trash can icon says 'Delete this branch. This will delete the branch locally and on the server.'

