

Adding a remote repository:

To add a new remote repository, use the `git remote add` command on the terminal, in the directory your repository is stored at.

The `git remote add` command takes two arguments:

- A remote name, for **example**, `origin`
- A remote URL, for **example**, `https://github.com/user/repo.git`

```
git remote add <name_of_your_remote> <url>
```

```
git remote add origin https://github.com/MRGazi/browserTest.git
```

```
git push -u origin master
```

```
git remote remove master
```

In your Github fork, you need to keep your master branch clean, by clean I mean without any changes, like that you can create at any time a branch from your master. Each time that you want to commit a bug or a feature, you need to create a branch for it, which will be a copy of your master branch.

When you do a pull request on a branch, you can continue to work on another branch and make another pull request on this other branch.

Before creating a new branch, pull the changes from upstream. Your master needs to be up to date.

```
$ git pull
```

Create the branch on your local machine and switch in this branch :

```
$ git checkout -b [name_of_your_new_branch]
```

Push the branch on github : This branch will be visible in your remote repo.

```
$ git push origin [name_of_your_new_branch]
```

Example: `git push origin mrg001` (Note: Here `mrg001` is the branch name)

When you want to commit something in your branch, be sure to be in your branch. Add -u parameter to set-upstream.

You can see all the branches created by using :

```
$ git branch -a
```

Which will show :

```
* approval_messages  
master  
master_clean
```

Add a new remote for your branch :

```
$ git remote add [name_of_your_remote] [name_of_your_new_branch]
```

Push changes from your commit into your branch :

```
$ git push [name_of_your_new_remote] [url]  
git push -u origin master
```

Update your branch when the original branch from official repository has been updated :

```
$ git fetch [name_of_your_remote]
```

Then you need to apply to merge changes if your branch is derivated from develop you need to do :

```
$ git merge [name_of_your_remote]/develop
```

Delete a branch on your local filesystem :

```
$ git branch -d [name_of_your_new_branch]
```

To force the deletion of local branch on your filesystem :

```
$ git branch -D [name_of_your_new_branch]
```

Delete the branch on github :

```
$ git push origin :[name_of_your_new_branch]
```

The only difference is the: to say delete, you can do it too by using GitHub interface to remove branch: <https://help.github.com/articles/deleting-unused-branches>.

If you want to change default branch, it's so easy with GitHub, in your fork go into Admin and in the drop-down list default branch choose what you want.

If you want create a new branch:

```
$ git branch <name_of_your_new_branch>
```

After Change

```
git status
```

```
git add -A
```

```
git commit -m "commit message"
```

```
git status
```