**Contribute to IDP Open Source**

**Fork the Project to your local system**

1. Set up Git

* Login to GitHub with valid credentials. Set up the Git to your local system.

2. Create local Clone to your Fork

1. Fork the repository to have a local copy. This will create a copy of the repository in your own GitHub account and you’ll see a note that it’s been forked underneath the project name:

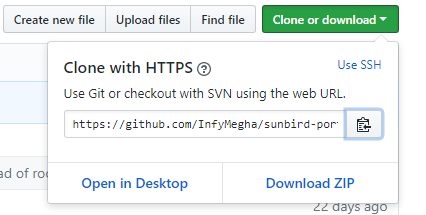


Right now, you have a fork of the sunbird-portal repository, but you don't have the files in that repository on your computer. Let's create a clone of your fork locally on your computer.

1. On GitHub, navigate to **your fork** of the sunbird-portal repository.



1. Under the repository name, click **Clone or download**.



1. In the Clone with HTTPs section, click  to copy the clone URL for the repository.
2. Open Git Bash. (It required to download in your system)
3. Type git clone, and then paste the URL you copied in Step 2. It will look like this, with your

GitHub username instead of YOUR-USERNAME:

git clone https://github.com/YOUR-USERNAME/Spoon-Knife

e.g. 

vii. Press **Enter**. Your local clone will be created.

e.g.one https://github.com/InfyMegha/sunbird-portal.git

Cloning into 'sunbird-portal'...

remote: Counting objects: 119011, done.

remote: Total 119011 (delta 0), reused 0 (delta 0), pack-reused 119011 eceiving

Receiving objects: 100% (119011/119011), 198.10 MiB | 1.07 MiB/s, done.

Resolving deltas: 100% (88620/88620), done.

Checking out files: 100% (1364/1364), done.

Now, you have a local copy of your fork of the Sunbird-Portal repository!

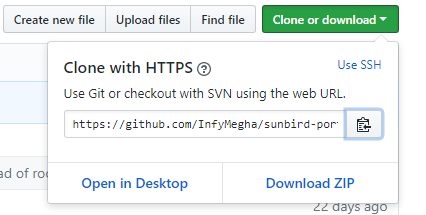
3. Configure GitHub to sync your fork with Original Sunbird-Portal repository

When you fork a project in order to propose changes to the original repository, you can configure Git to pull changes from the original, or upstream, repository into the local clone of your fork.

1. On GitHub, navigate to the Subnbird-portal repository.

Clone or download button

1. Under the repository name, click **Clone or download**

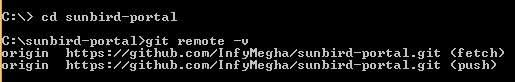


1. In the Clone with HTTPs section, click  to copy the clone URL for the repository.
2. Open Git Bash.
3. Change directories to the location of the fork you cloned in Step 2: Create a local clone of your fork.

* To go to your home directory, type just cd with no other text.
* To list the files and folders in your current directory, type ls.
* To go into one of your listed directories, type cd your\_listed\_directory.
* To go up one directory, type cd ...
* Type git remote -v and press **Enter**. You'll see the current configured remote repository for your fork.

1. git remote –v
2. origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)
3. origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

e.g.



1. Type git remote add upstream, and then paste the URL you copied in Step 2 and press **Enter**. It will look like this:

git remote add upstream https://github.com/octocat/Spoon-Knife.git

e.g. 

1. To verify the new upstream repository, you've specified for your fork, type git remote -vagain. You should see the URL for your fork as origin, and the URL for the original repository as upstream.

git remote -v

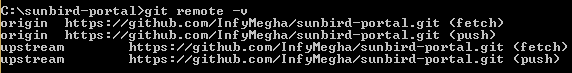
origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (fetch)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (push)

e.g.



Now, you can keep your fork synced with the upstream repository with a few Git commands.

**Syncing with Fork**

1. Open Git Bash.
2. Change the current working directory to your local project.
3. Fetch the branches and their respective commits from the upstream repository. Commits

to master will be stored in a local branch, upstream/master.

git fetch upstream

remote: Counting objects: 75, done.

remote: Compressing objects: 100% (53/53), done.

remote: Total 62 (delta 27), reused 44 (delta 9)

Unpacking objects: 100% (62/62), done.

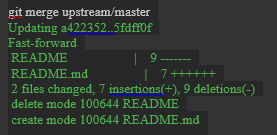
From https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY

\* [new branch] master -> upstream/master

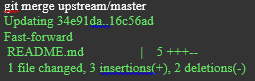
1. Check out your fork's local master branch.



1. Merge the changes from upstream/master into your local master branch. This brings your fork's master branch into sync with the upstream repository, without losing your local changes.



1. If your local branch didn't have any unique commits, Git will instead perform a "fast-forward":



**Configuring a Remote for Fork**

You must configure a remote that points to the upstream repository in Git to sync changes you make in a fork with the original repository. This also allows you to sync changes made in the original repository with the fork.

1. Open Git Bash.
2. List the current configured remote repository for your fork.

git remote -v

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

eg.



1. Specify a new remote upstream repository that will be synced with the fork.

git remote add upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git

e.g. 

1. Verify the new upstream repository you've specified for your fork.

git remote -v

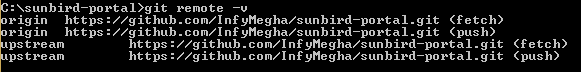
origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (fetch)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (push)

e.g.



**Merging an upstream repository into your work**

If you don't have push (write) access to an upstream repository, then you can pull commits from that repository into your own fork.

1. Open Git Bash.
2. Change the current working directory to your local project.
3. Check out the branch you wish to merge to. Usually, you will merge into master.

git checkout master

e.g.



1. Pull the desired branch from the upstream repository. This method will retain the commit history without modification.

git pull https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git BRANCH\_NAME

e.g.



1. If there are conflicts, resolve them. For more information, see "[Addressing merge conflicts](https://help.github.com/articles/addressing-merge-conflicts)".
2. Commit the merge.
3. Review the changes and ensure they are satisfactory.
4. Push the merge to your GitHub repository.

git push origin master

**Pushing to a Remote**

Use git push to push commits made on your local branch to a remote repository.

The git push command takes two arguments:

* A remote name, for example, origin
* A branch name, for example, master

For example:

git push  *<REMOTENAME> <BRANCHNAME>*

As an example, you usually run git push origin master to push your local changes to your online repository.

e.g.

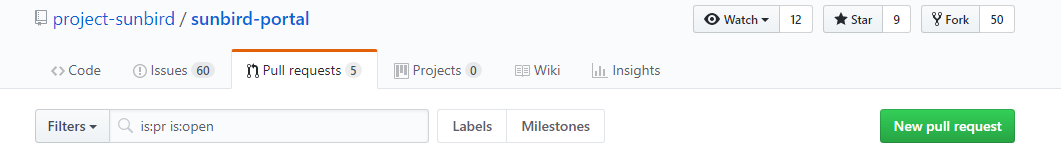


Creating a pull request from a fork

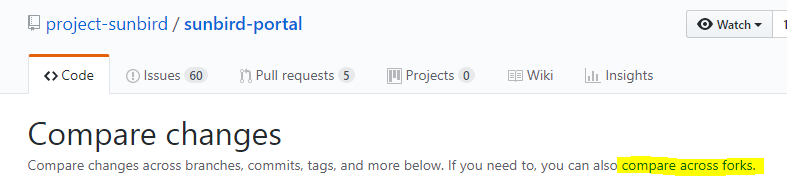
If you've forked a repository and made changes to the fork, you can ask that the upstream repository accept your changes by creating a pull request.

You can open a pull request to the upstream repository from any branch or commit in your fork. We recommend that you make changes in a topic branch, so that you can push follow up commits if you receive feedback on your pull request. You also have the option to give the upstream repository's maintainers the ability to make commits on your topic branch to update your pull request. If your pull request compares your topic branch with a branch in the upstream repository as the base branch, then your topic branch is also called the compare branch of the pull request. For more information about pull request branches, including examples, see "Creating a Pull Request."

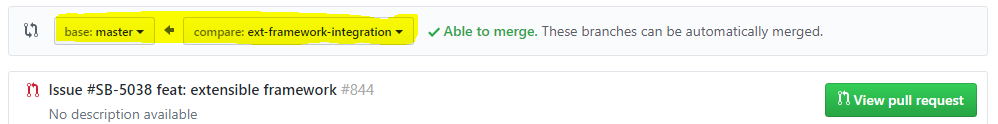
1. Navigate to the original repository you created your fork from.



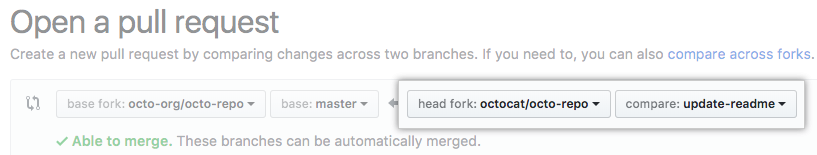
1. To the right of the Branch menu, click **New pull request**.



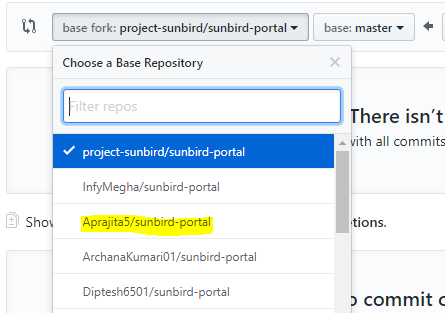
1. On the Compare page, click **compare across forks**.



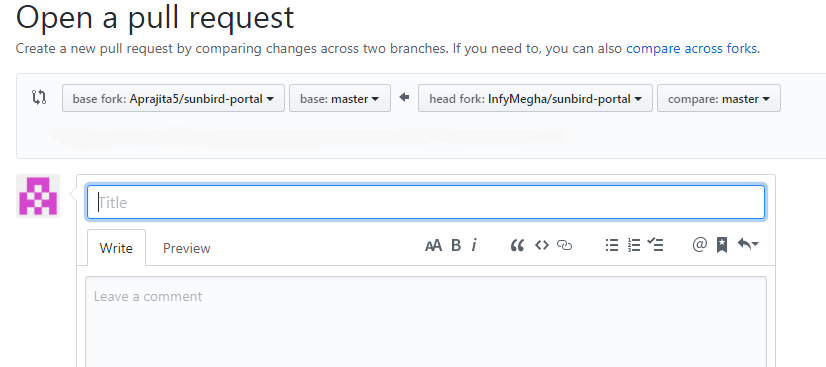
1. Confirm that the *base fork* is the repository you'd like to merge changes into. Use the *base branch* drop-down menu to select the branch of the upstream repository you'd like to merge changes into.



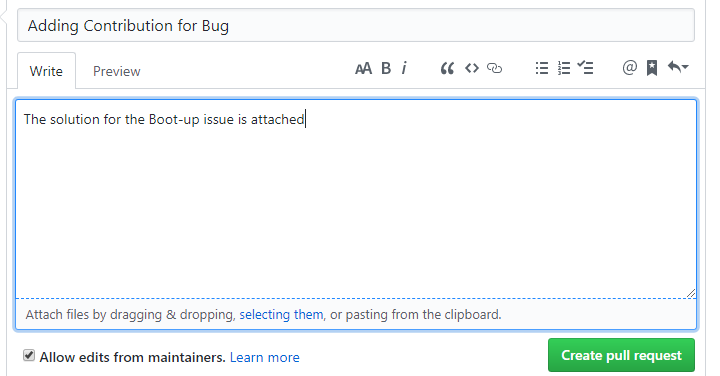
1. Use the *head fork* drop-down menu to select your fork, then use the *compare branch* drop-down menu to select the branch you made your changes in.



vi. Type a title and description for your pull request.



1. If you do not want to allow anyone with push access to the upstream repository to make changes to your PR, unselect **Allow edits from maintainers**.



1. Click **Create pull request**.