# Development

Start by forking the dolphinscheduler GitHub repository, make changes in a branch and then send a pull request.

## Set up your dolphinscheduler GitHub Repository

There are three branches in the remote repository currently:

- `master` : normal delivery branch. After the stable version is released, the code for the stable version branch is merged into the master branch.

- `dev` : daily development branch. The daily development branch, the newly submitted code can pull requests to this branch.

- `x.x.x-release` : the stable release version.

So, you should fork the `dev` branch.

After forking the [dolphinscheduler upstream source repository](https://github.com/apache/incubator-dolphinscheduler/fork) to your personal repository, you can set your personal development environment.

```sh

$ cd <your work direcotry>

$ git clone < your personal forked dolphinscheduler repo>

$ cd incubator-dolphinscheduler

```

## Set git remote as ``upstream``

Add remote repository address, named upstream

```sh

git remote add upstream https://github.com/apache/incubator-dolphinscheduler.git

```

View repository:

```sh

git remote -v

```

There will be two repositories at this time: origin (your own warehouse) and upstream (remote repository)

Get/update remote repository code (already the latest code, skip it).

```sh

git fetch upstream

```

Synchronize remote repository code to local repository

```sh

git checkout origin/dev

git merge --no-ff upstream/dev

```

If remote branch has a new branch `DEV-1.0`, you need to synchronize this branch to the local repository.

```

git checkout -b dev-1.0 upstream/dev-1.0

git push --set-upstream origin dev1.0

```

## Create your feature branch

Before making code changes, make sure you create a separate branch for them.

```sh

$ git checkout -b <your-feature>

```

## Commit changes

After modifying the code locally, submit it to your own repository:

```sh

git commit -m 'information about your feature'

```

## Push to the branch

Push your locally committed changes to the remote origin (your fork).

```

$ git push origin <your-feature>

```

## Create a pull request

After submitting changes to your remote repository, you should click on the new pull request On the following github page.

<p align = "center">

<img src = "http://geek.analysys.cn/static/upload/221/2019-04-02/90f3abbf-70ef-4334-b8d6-9014c9cf4c7f.png" width ="60%"/>

</p>

Select the modified local branch and the branch to merge past to create a pull request.

<p align = "center">

<img src = "http://geek.analysys.cn/static/upload/221/2019-04-02/fe7eecfe-2720-4736-951b-b3387cf1ae41.png" width ="60%"/>

</p>

Next, the administrator is responsible for \*\*merging\*\* to complete the pull request.