

Wikiprint Book

Title: Development

Subject: TracMeteo - drm4gDevelopment

Version: 9

Date: 05/23/2022 08:35:28 AM

Table of Contents

Development	3
Necessary Steps	3
Testing the DRM4G	3
Installing your version in a virtual environment	4
Committing changes	4
Creating a Pull Request	4

Development

The aim of this section will be to explain how to contribute to the development of the DRM4G.

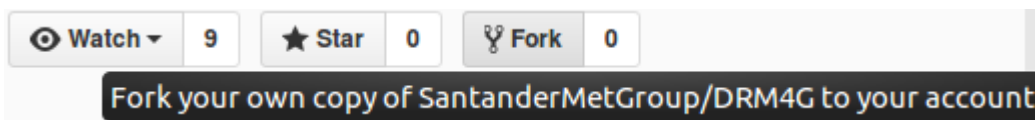
To make it easier and more accessible for anyone to play around with the DRM4G, we have decided to host our source code on **GitHub**.

- You can find the project [?here](#).

Necessary Steps

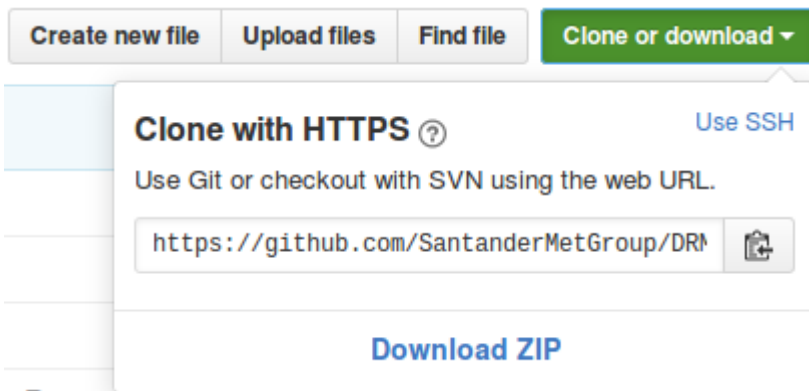
For those of you who wish to help but don't know how, the first thing you need is a [?GitHub account](#).

In our project's page hit the "**Fork**" button at the top right corner:



This will create a copy of our repository in your account where you'll develop your own feature or implement a bugfix that you may believe is necessary. You'll be submitting changes to this one until you are certain everything works properly, at which point you can request to have your changes integrated into the DRM4G's repository.

To continue, you'll need to setup a local repository where you'll be changing the code and doing your testing. To do this you'll need your repository URL, that can be obtained adding **.git** to your project page or by clicking on the "**Clone or download**" button:



In Linux operating systems:

- Open a terminal on the folder in which you wish your local copy of the repository to be stored (it's recommended to use an empty folder) and run the following commands:

```
git init #to initialize an empty Git repository
git remote add origin <your_repository_url> #to make your local repository point to your remote repository in GitHub
git fetch
git checkout develop #to create a local copy of the develop branch
```

To be consistent with our [gitflow](#), all you'll be able to do is create **feature or bugfix branches**, and you'll have to follow our naming conventions to do so.

- The naming convention will just be to create branches in lower case letters separated by underscores ("_") that describe what you're trying to accomplish with the branch.

With this, you will now have DRM4G's source code at your disposal.

From here you could create branches for every new feature you'd like to include to the DRM4G, for a more in depth tutorial on how to do that, click [?here](#).

Testing the DRM4G

Once you've made the changes you wanted to, you'll want to install your version to be sure that your new feature is working properly.

Just in case you would like to try out different versions, we recommend you use a [virtual environment](#) to test it.

- [?Here](#) you can find a tutorial on how to install a virtual environment, or you can look for one on your own.

Before you can install and try out your own version, you'll have to build your own package:

- Open a terminal in the folder where your repository is located.
- Run the command `python setup.py sdist`

This will create a distribution package under a folder called **dist**.

Installing your version in a virtual environment

Go to wherever you have your virtual environment, open a terminal and execute the following commands:

```
source bin/activate
export DRM4G_DIR = $PWD/conf
pip install path/to/drm4g/package
```

DRM4G_DIR is where the configuration files will be installed. More information [here](#).

And that's it. Now you can use and test your own version of DRM4G.

For other ways to install the DRM4G, you can check [here](#).

Committing changes

After you've tested that everything is in working order it's time to update your GitHub fork.

```
git add .
git commit -m "Description of the changes you've made"
git push origin develop
```

From here you'll have to create a **Pull request**.

Creating a Pull Request

#####First you should update and check that there are no conflicts#####

Go to your repository page on github