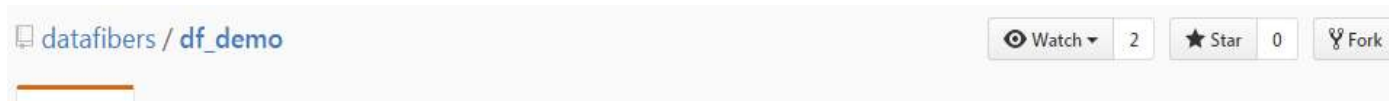


Use Github DataFibers Repository

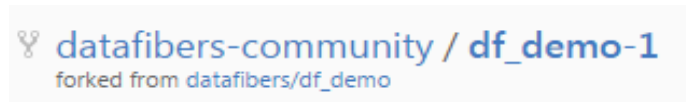
0. Download and Install Git for windows (Git-2.10.1-64-bit.exe)

<https://git-scm.com/download/win>

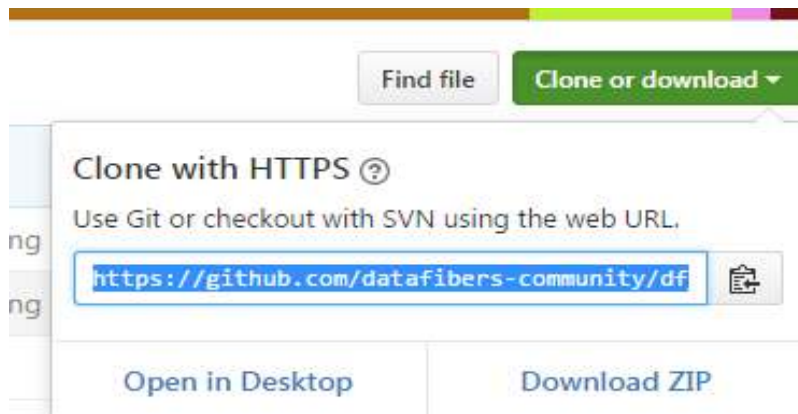
1. Login DataFibers Github, find out the repository, for exmaple df_demo. fork the repository for you own copy



in this case, df_demo-1 is forked from datafibers/df_demo,



2. Get the URL by clicking Clone or download button



https://github.com/datafibers-community/df_demo-1.git

3. Open Git Bash window

C:\Git\git-bash.exe

4. Goto your folder where you want to check out the code, Create a directory,

```
cd c:\DF_Demo
mkdir my_df_demo-1
```

5. Go to the directory

```
cd my_df_demo-1
```

6. Git clone to clone the repository

```
git clone https://github.com/datafibers-community/df_demo-1.git
```

```
$ git clone https://github.com/datafibers-community/df_demo-1.git
Cloning into 'df_demo-1'...
remote: Counting objects: 288, done.
remote: Total 288 (delta 0), reused 0 (delta 0), pack-reused 288
Receiving objects: 100% (288/288), 74.92 KiB | 0 bytes/s, done.
Resolving deltas: 100% (103/103), done.
```

7. The repository is in local drive and in there df_demo-1 folder is created. Go to df_demo-1 folder

```
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1
$ ls
df_demo-1/

ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1
$ cd df_demo-1

ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
```

```
$ ls
df-data-collector/  df-environment/  license.md  README.md
df-demo.iml        df-visualisation/ pom.xml

ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ |
```

8. Add a new file, for example myNote.txt

9. git status to find out the changes

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    myNote.txt

nothing added to commit but untracked files present (use "git add" to track)
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ |
```

10. Staging the files to be checked into repository

git add myNote.txt

```
$ git add myNote.txt
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ |
```

11. Set your account to default identity

git config --global user.email "you@example.com"

git config --global user.name "Your Name"

```
$ git config user.email "allen.zhangliugen@gmail.com"
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ git config user.name "Allen Zhang"
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
```

12. Commit the code change to local repository

git commit -m "Check in my changes" myNote.txt

13. To avoid conflict, use git fetch and git pull to get and/or merge the latest changes from remote repository

git fetch

git pull

```
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ git fetch
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/datafibers-community/df_demo-1
   b013903..cf40ec5  master       -> origin/master
   * [new branch]      development_AllenZhang_DFDemo1_201611 -> origin/development_AllenZhang_DFDemo1_201611

ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ git pull
Merge made by the 'recursive' strategy.
```

14. To publish the code change to remote repository

git push

```
ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
$ git push
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 370 bytes | 0 bytes/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/datafibers-community/df_demo-1.git
   cf40ec5..1d1507f  master -> master

ALZHANG@ALZHANG-PC MINGW64 /c/DF_Demo/my_df_demo-1/df_demo-1 (master)
```

15. If you check out a branch, any work done after checking out will be in this branch.

git checkout development_AllenZhang_DFDemo1_201611

```
$ git checkout development_AllenZhang_DFDemo1_201611
Already on 'development_AllenZhang_DFDemo1_201611'
Your branch is up-to-date with 'origin/development_AllenZhang_DFDemo1_201611'.
```