Data Scientist’s Toolbox Project Tutorial

By: Matt Marchand
Last Updated: 7/19/2014

I’m new to Git Bash and GitHub so I thought I’d write this tutorial as a reference for myself, though maybe it can help others, too.

This tutorial makes the following assumptions:

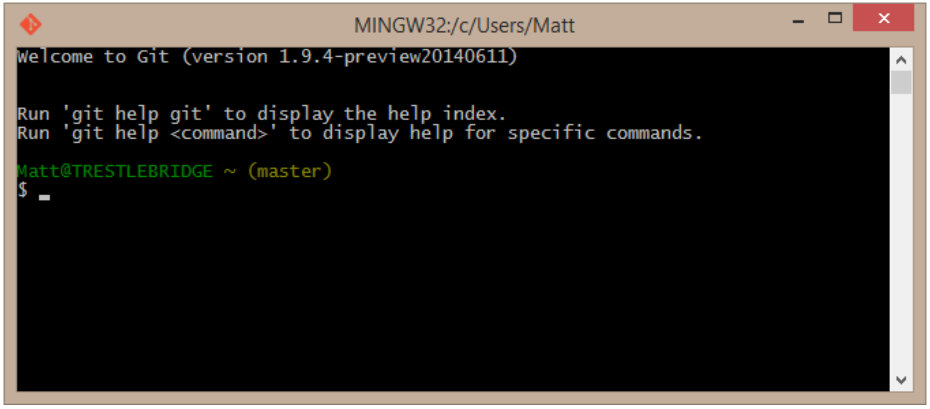
- You already watched all the lectures and did all the quizzes
- You already have GitBash installed
- You already have a GitHub account created
- You have already completed the course project phase where you created a repository (“repo” called “datasciencecoursera”

Note: I’m doing this on a Windows 8.1 PC. If you are using a Mac or a Linux system (or possibly with an older revision of Windows), you may see different folder structures get created in your local filesystem.

Start by creating the HelloWord.md file. It doesn’t matter where you put this file for now.

Part 1: Create and set up your local repo

1) Open the GitBash CLI application

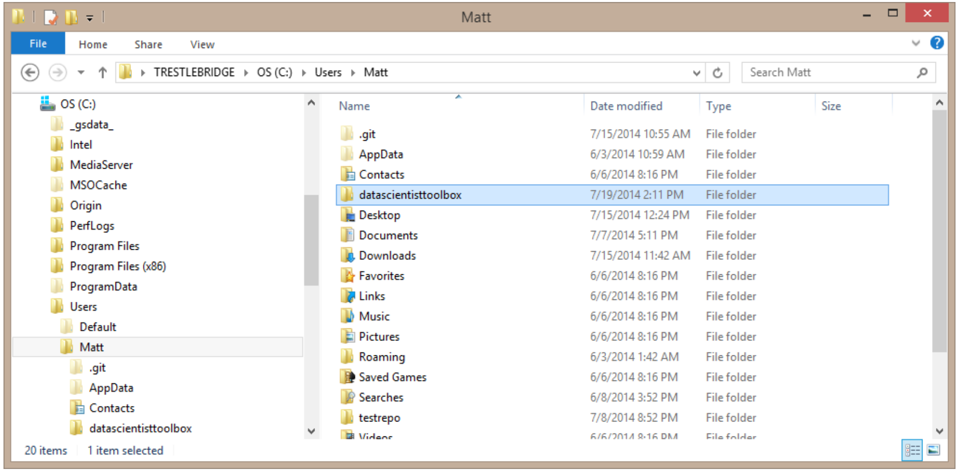


```
Welcome to Git (version 1.9.4-preview20140611)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Matt@TRESTLEBRIDGE ~ (master)
$ _
```

2) Create a directory on the GitBash command line. You can use whatever name you like, but I created one called datascientisttoolbox so I will use that here:

`mkdir ~/datascientisttoolbox`

Note that this creates a folder in your c:\Users\WINDOWSUSERNAME folder by default! You can change to any folder on your system that you want in GitBash first.

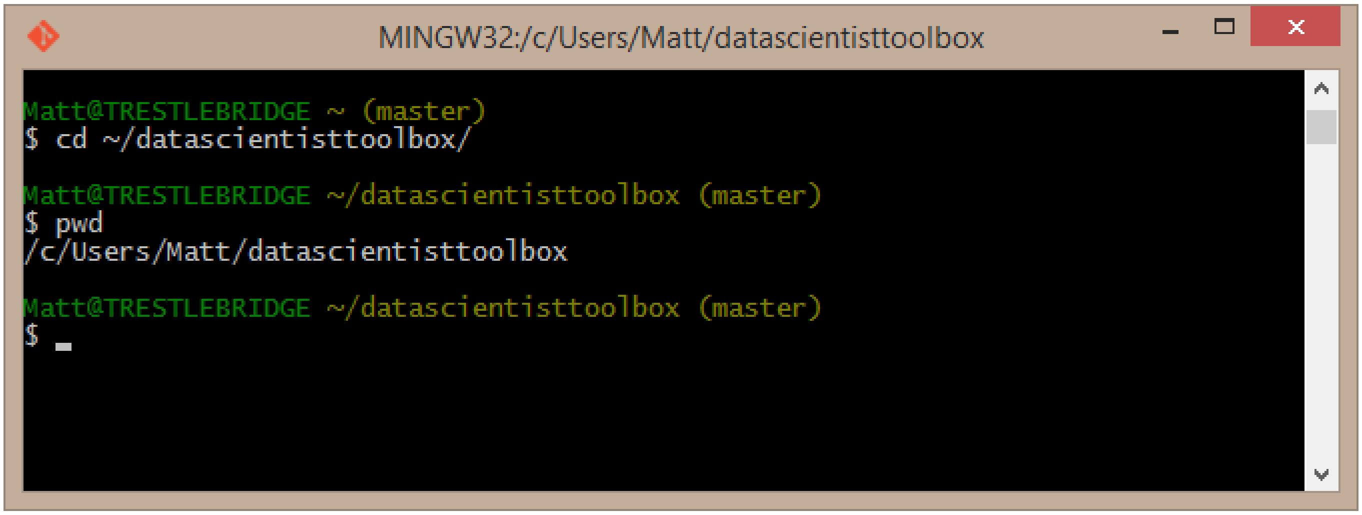


Name	Date modified	Type	Size
.git	7/15/2014 10:55 AM	File folder	
AppData	6/3/2014 10:59 AM	File folder	
Contacts	6/6/2014 8:16 PM	File folder	
datascientisttoolbox	7/19/2014 2:11 PM	File folder	
Desktop	7/15/2014 12:24 PM	File folder	
Documents	7/7/2014 5:11 PM	File folder	
Downloads	7/15/2014 11:42 AM	File folder	
Favorites	6/6/2014 8:16 PM	File folder	
Links	6/6/2014 8:16 PM	File folder	
Music	6/6/2014 8:16 PM	File folder	
Pictures	6/6/2014 8:16 PM	File folder	
Roaming	6/3/2014 1:42 AM	File folder	
Saved Games	6/6/2014 8:16 PM	File folder	
Searches	6/8/2014 3:52 PM	File folder	
testrepo	7/8/2014 8:52 PM	File folder	
Videos	6/6/2014 8:16 PM	File folder	

3) Change to that directory:

```
cd ~/datascientisttoolbox
```

This changes the GitBash CLI to be operating with **datascientisttoolbox** as the working directory. You can run **`pwd`** to verify this.

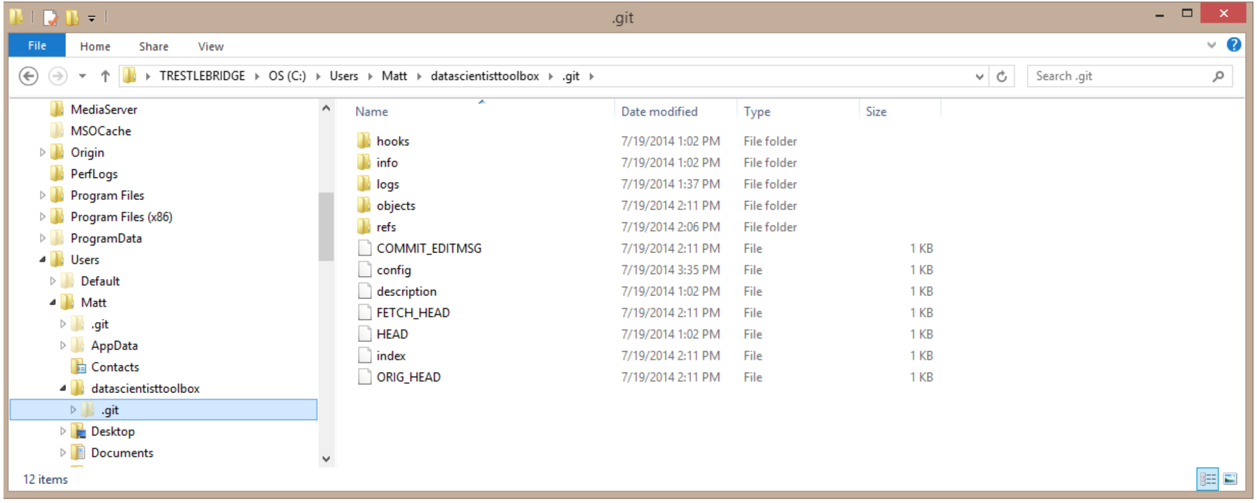


```
Matt@TRESTLEBRIDGE ~ (master)
$ cd ~/datascientisttoolbox/

Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ pwd
/c/Users/Matt/datascientisttoolbox

Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ _
```

4) Initialize the directory as a repo:

```
git init
```

This creates the .git repo substructure underneath the datascientist toolbox directory. The .git folder may be hidden, so if you want to see it you may need to enable the viewing of hidden files and folders in your File Explorer.



Name	Date modified	Type	Size
hooks	7/19/2014 1:02 PM	File folder	
info	7/19/2014 1:02 PM	File folder	
logs	7/19/2014 1:37 PM	File folder	
objects	7/19/2014 2:11 PM	File folder	
refs	7/19/2014 2:06 PM	File folder	
COMMIT_EDITMSG	7/19/2014 2:11 PM	File	1 KB
config	7/19/2014 3:35 PM	File	1 KB
description	7/19/2014 1:02 PM	File	1 KB
FETCH_HEAD	7/19/2014 2:11 PM	File	1 KB
HEAD	7/19/2014 1:02 PM	File	1 KB
index	7/19/2014 2:11 PM	File	1 KB
ORIG_HEAD	7/19/2014 2:11 PM	File	1 KB

5) Point the local repo to your remote repo. The `ACCOUNT_NAME` and `REPO_NAME` will be unique to you, although if you followed the project directions the `REPO_NAME` will simply be "`datasciencecoursera`":

```
git remote add origin https://github.com/ACCOUNT_NAME/REPO_NAME
```

```
example: git remote add origin https://github.com/MatticusCaesar/datasciencecoursera
```

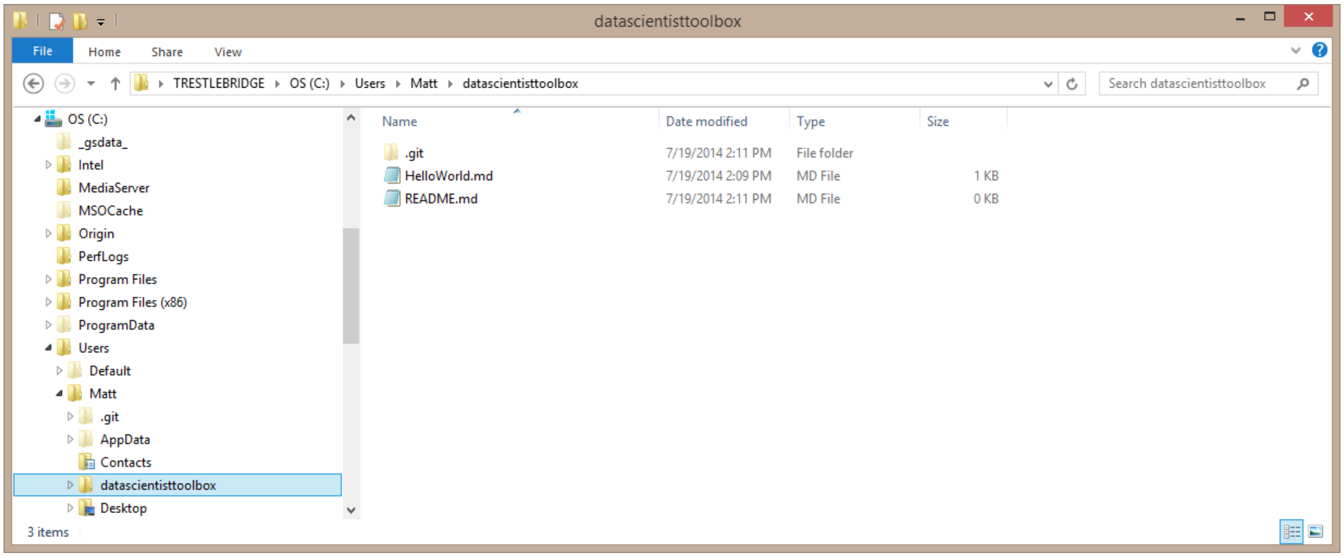
Part 2: Getting the file into the remote repo

6) Copy your **HelloWorld.md** file (or whatever other files you happen to want to upload/commit) into the datascientisttoolbox folder using File Explorer or whatever method you prefer.

7) Sync your remote repo with the local repo:

```
git pull https://github.com/ACCOUNT_NAME/REPO_NAME
```

This pulls all files to your local repository. If you did the earlier parts of the project correctly, you'll see that the **README.md** file now resides in your c:\Users\USERNAME\REPOSITORYNAME folder:



Name	Date modified	Type	Size
.git	7/19/2014 2:11 PM	File folder	
HelloWorld.md	7/19/2014 2:09 PM	MD File	1 KB
README.md	7/19/2014 2:11 PM	MD File	0 KB

8) Add the HelloWorld.md to the queue for committing. Use "." as shorthand for adding all files (a wildcard), or you can specify explicit filenames separated by a space:

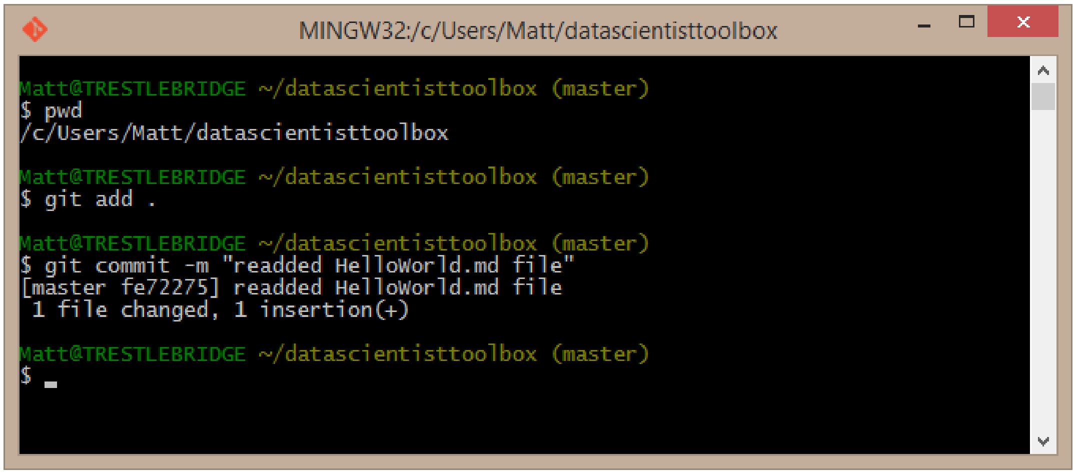
```
git add .
```

or

```
git add HelloWorld.md
```

9) Commit the changes. Use the –m parameter to add a "Change message".

```
git commit –m "I'm committing the HelloWorld.md file"
```

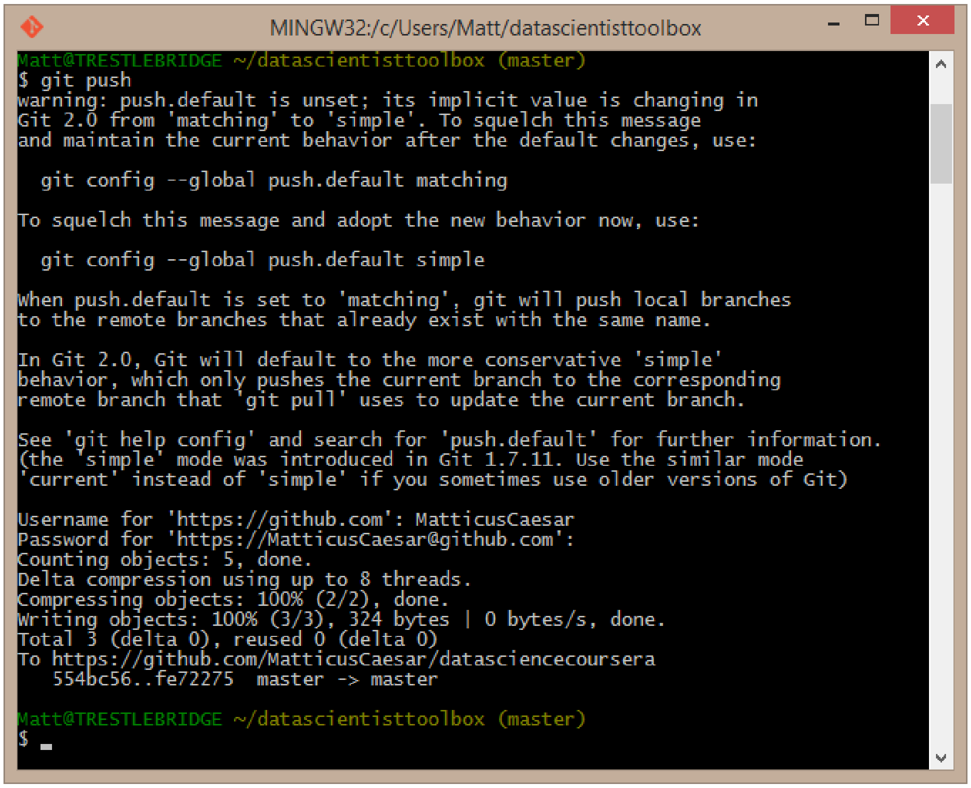


```
Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ pwd
/c/Users/Matt/datascientisttoolbox

Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ git add .

Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ git commit -m "readded HelloWorld.md file"
[master fe72275] readded HelloWorld.md file
 1 file changed, 1 insertion(+)

Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ _
```

10) Upload it all to the remote repo on GitHub:

```
git push
```

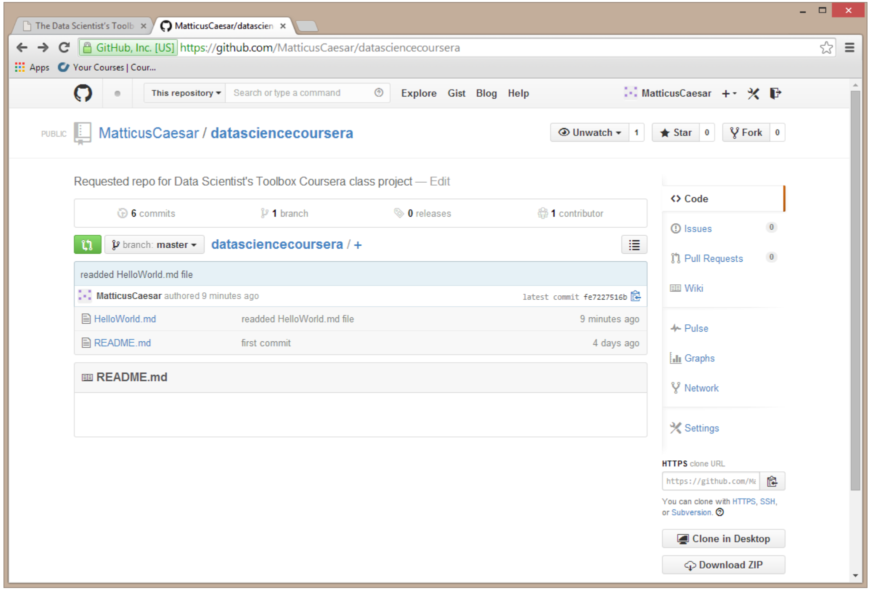
Note: You will be asked to authenticate to your GitHub account when you do this, so make sure you have your accountname and password handy:



```
Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': MatticusCaesar
Password for 'https://MatticusCaesar@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/MatticusCaesar/datasciencecoursera
   554bc56..fe72275  master -> master

Matt@TRESTLEBRIDGE ~/datascientisttoolbox (master)
$ 
```

Part 3 – Verifying success

That should be it! If you use your browser to visit your repository you'll see the uploaded file:



https://github.com/MatticusCaesar/datasciencecoursera

MatticusCaesar / datasciencecoursera

Requested repo for Data Scientist's Toolbox Coursera class project — Edit

6 commits · 1 branch · 0 releases · 1 contributor

branch: master · datasciencecoursera / +

readded HelloWorld.md file

MatticusCaesar authored 9 minutes ago · latest commit fe7227516b

HelloWorld.md	readded HelloWorld.md file	9 minutes ago
README.md	first commit	4 days ago

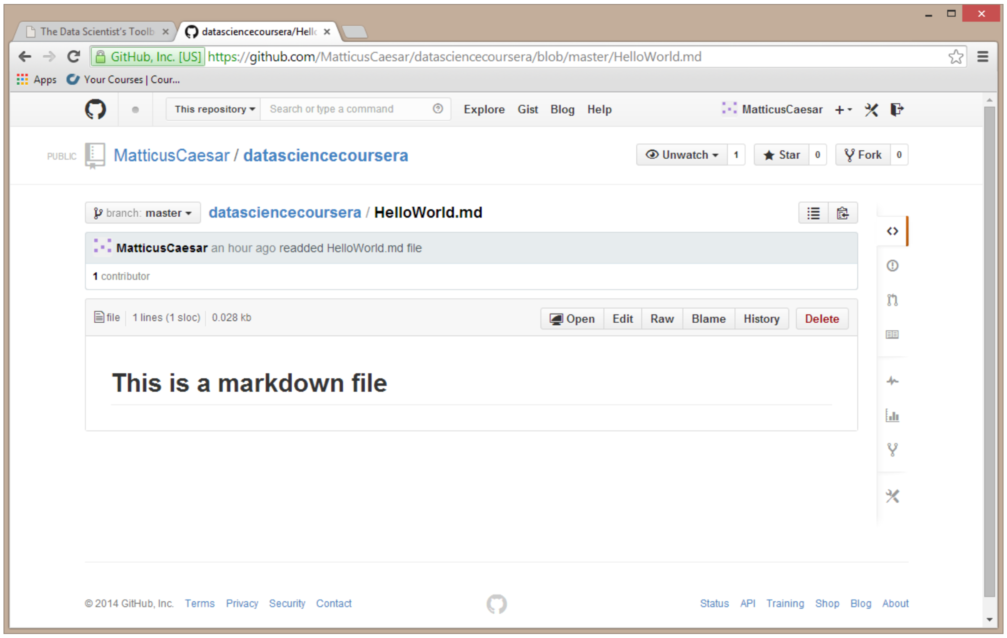
README.md

HTTPS clone URL

https://github.com/Ma

You can clone with HTTPS, SSH, or Subversion.

If you click the link for the HelloWorld.md file, you'll see the contents you put in.



https://github.com/MatticusCaesar/datasciencecoursera/blob/master/HelloWorld.md

MatticusCaesar / datasciencecoursera

branch: master · datasciencecoursera / HelloWorld.md

MatticusCaesar an hour ago readded HelloWorld.md file

1 contributor

file · 1 lines (1 sloc) · 0.028 kb

This is a markdown file

© 2014 GitHub, Inc.

You can now copy your URL listed in the browser to do that phase of the assignment on the Coursera Data Scientist's Toolbox class assignment page.