# Quantlets

Härdle Hu Petukhina Sterling Zinovyeva

Ladislaus von Bortkiewicz Professor of Statistics
BRC Blockchain Research Center
International Research Training Group
Humboldt-Universität zu Berlin
lvb.wiwi.hu-berlin.de
www.case.hu-berlin.de
irtg1792.hu-berlin.de

# Summary

- **Quantlets have to be submitted to a repository in the Quantlet Github Organization**
- **Comply to the styleguide of Quantlets!**
- **Contribute Quantlets as member or non-member**



# Outline



# Structure of Quantlet Folder 1

- Each Quantlet Q is a program that needs to be accompanied by a Metainfo.txt file, both files need to be in the same folder.
- Per folder only one Q. If you have multiple Qs structure them in GitHub like this:
  1. \RepositoryName\…\FolderName1\QuantletName1
  2. \RepositoryName\…\FolderName2\QuantletName2
- Give each Q a meaningful name:
  3. The name should start with the project / book / class abbreviation followed by a unique name.
- Each Q shall be executable, hence provide the input data! *If data are classified or too big, provide synthetic data of same structure or talk to the QTeam about alternative options.*
- Save output pictures in the same folder as png or jpg.



# Structure of Quantlet Folder 2

- If you have a project which includes multiple major steps, split them into multiple Qs. For example a Q for each of the following steps:
  1. Data collection / scraping / mining,
  2. Data preprocessing,
  3. Data exploration,
  4. Data visualisation.
- The README file is created automatically out of the
  5. Metainfo file,
  6. Pictures in the same folder (in alphabetical order),
  7. Code of the Q.
- Existing README files are not overwritten, if you make changes in the Metainfo file or the Quantlet, please delete the README.



# Metainfo.txt File 1

- **Required** Meta-Information:
  - **Name of Quantlet**: *Same name as the Quantlet without the program ending* (.r,.py,.m,...). Select a meaningful name!
  - **Published in**: Book / Paper/ Class / other place of publication
  - **Description**: Describe with at least 10 words what this Quantlet does, which techniques are used, what is the applied use case, for what purpose etc.
  - **Keywords**: At least 5 keywords.
  - **Author**: Name of the authors



# Metainfo.txt File 2

- **Optional** Meta-Information:
  - **See also**: mention related Quantlets, e.g. Quantlets of same project
  - **Submitted**: state the name and the time of the original submission
  - **Datafile**: All datafiles used by your code need to be listed here
  - **Input**: Should contain some new info, which is not written in other metainfo fields
  - **Output**: Should contain some new info, which is not written in other metainfo fields
  - **Example**: Should contain a list of generated plots and descriptions, which are not written in other metainfo fields



# Metainfo.txt File 3

- The Metainfo file has to be a YAML debuggable text file.
  - A template is provided on GitHub.
  - If the Metainfo file is NOT debuggable the Quantlet is not displayed on Quantlet.de. You can check yourself whether it is debuggable, e.g. on http://yaml-online-parser.appspot.com/
- YAML rules:
  - The colon ‘:’ separates the data field (left) from its description (right).
  - The dash ‘-’ enumerates list items. Avoid them in text.
  - Put texts in quotes (single or double), especially if the text is multiline or if special characters, e.g. ‘:’, ‘-‘, ‘_’, …, are used.



# README File

- The README file is the representation of the Quantlet, it contains
  - Name of Quantlet,
  - Metainfo file,
  - Graphics in the same folder (only JPEG or PNG),
  - Source code of Quantlet.
- The README file is automatically created for all Quantlets in the Github organisation.
  - README files are not overwritten, thus not updated.
  - If you need an updated README file, delete the README file, a new one is created automatically.



# Example Quantlets

- Class projects:
  - SPL_class_WS1617
  - DEDA_Class_SS2021
- Projects with multiple Qs, e.g. thesis:
  - FVC - Face Value of Companies
  - BLEM



# Special Cases

- R Shiny Apps:
  1. Program UI and Server in one R Script, e.g.

```
library(”shiny”)
ui        = shinyUI(…)
server  = shinyServer(function(input, output)){…}
app      = shinyApp(ui, server)
```

  2. See for example SVCJOptionApp.
- If you have a special case that is not listed below ask your supervisor and if necessary contact the Quantlet Team.



# Outline





# Member in GitHub Quantlet organization

- Check if you are member in the Quantlet organization (login to GitHub first): https://github.com/orgs/QuantLet/people
- If you are not, the owner (Prof. Härdle) or the Quantlet team (quantlet.wiwi@hu-berlin.de) can invite you.

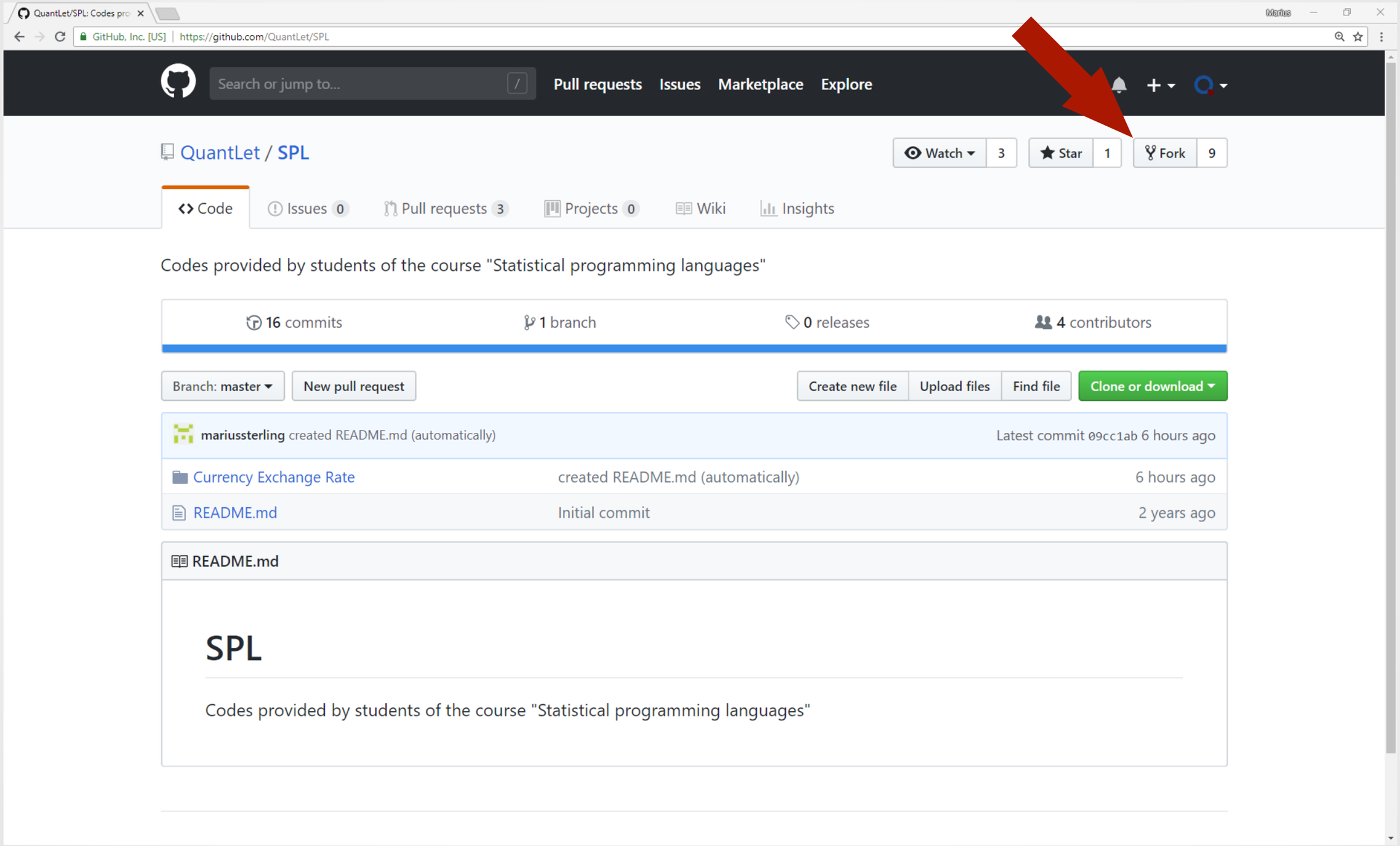


# Submission to existing repository

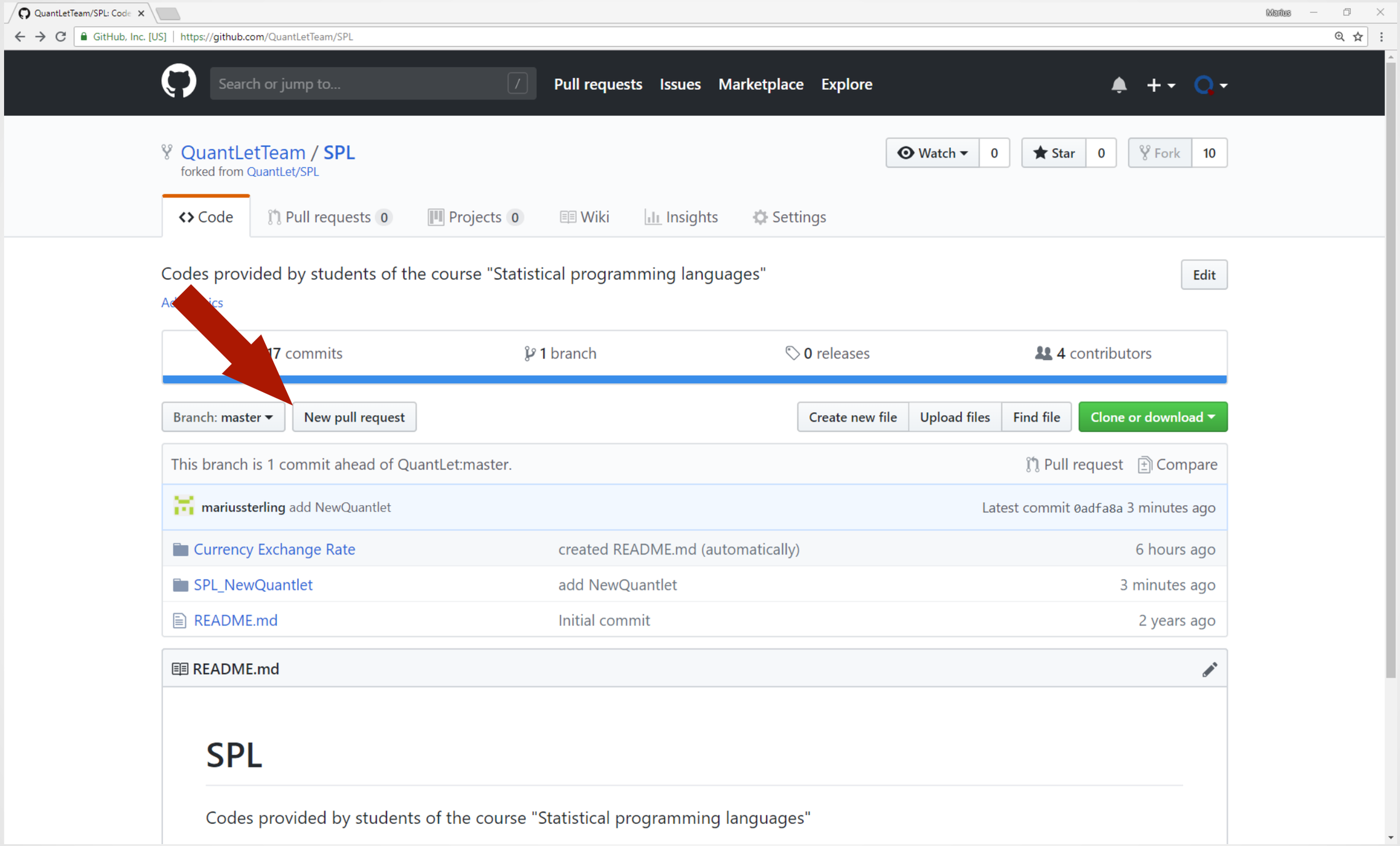
- There are 2 ways to submit to existing repositories:
- Directly:
  - You need write privileges (by default only the repository admin has such privileges),
  - the repository admin can grant write privileges,
  - alter the repository as if it was your own.
- Via pull request:
  - Step 1: Fork the repository,
  - Step 2: Commit changes,
  - Step 3: Create a pull request,
  - the steps will be explained in the following pages.



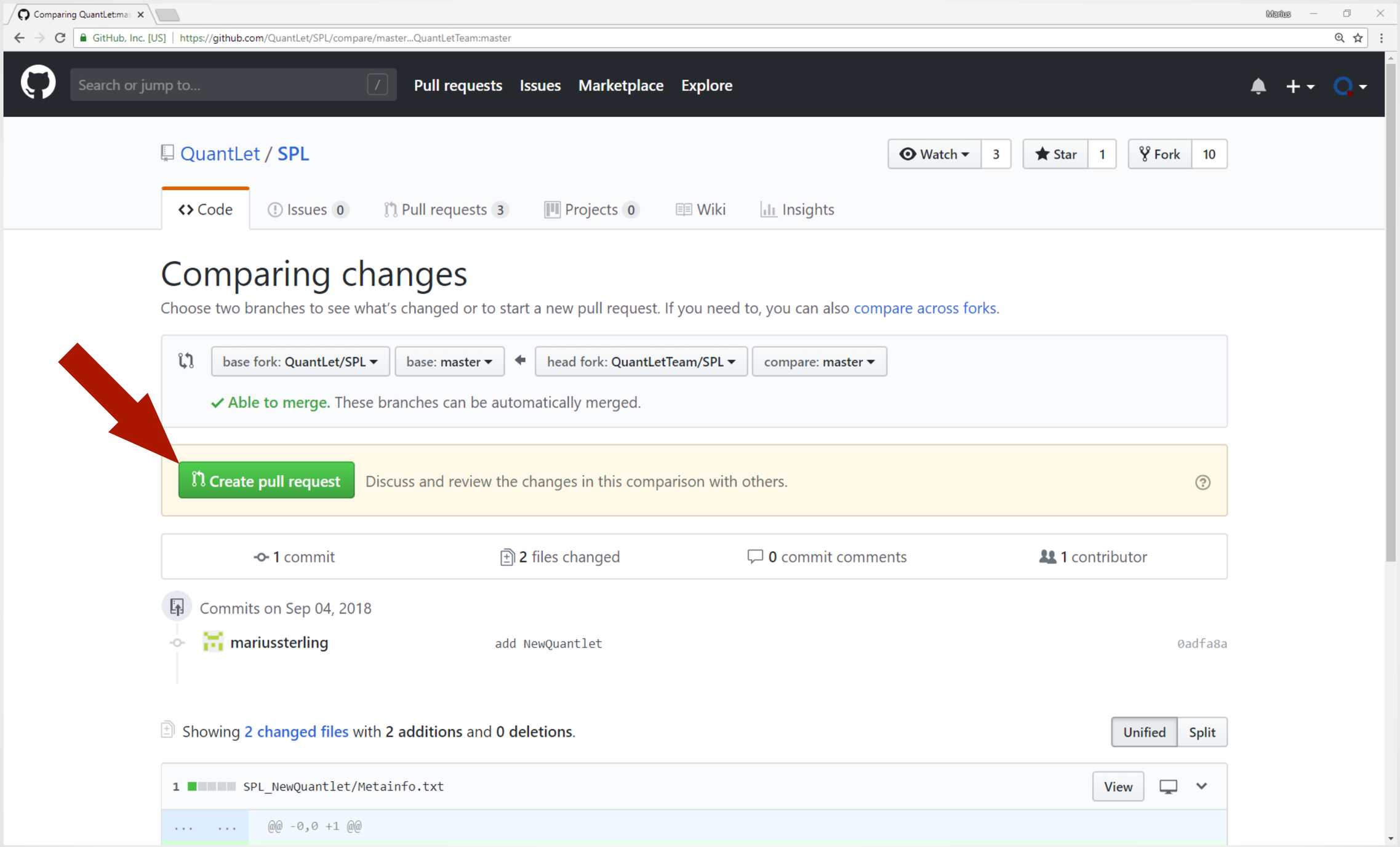
# Step 1: Fork repository



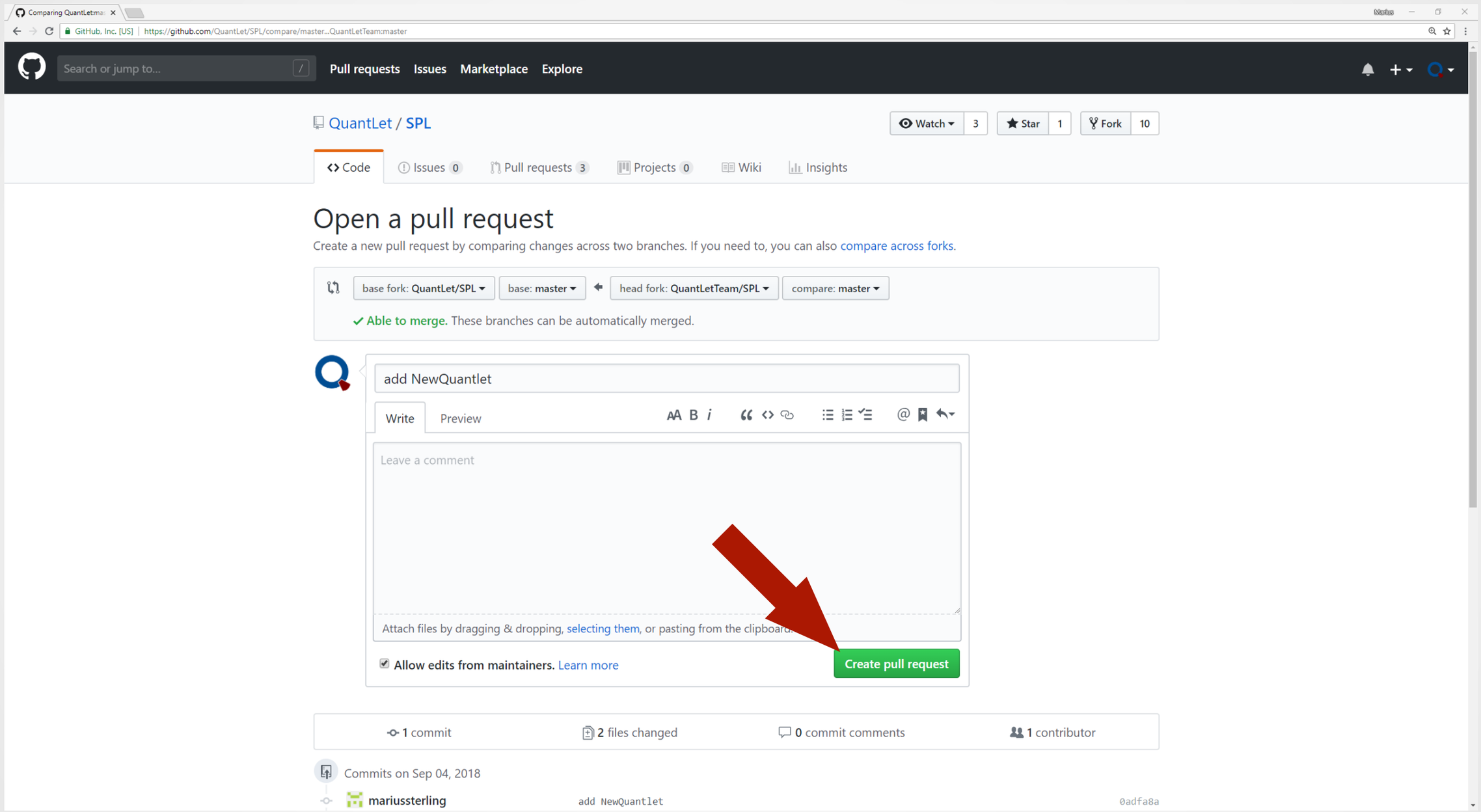
# Step 2: Commit changes



# Step 3: Create pull request 1



# Step 3: Create pull request 2

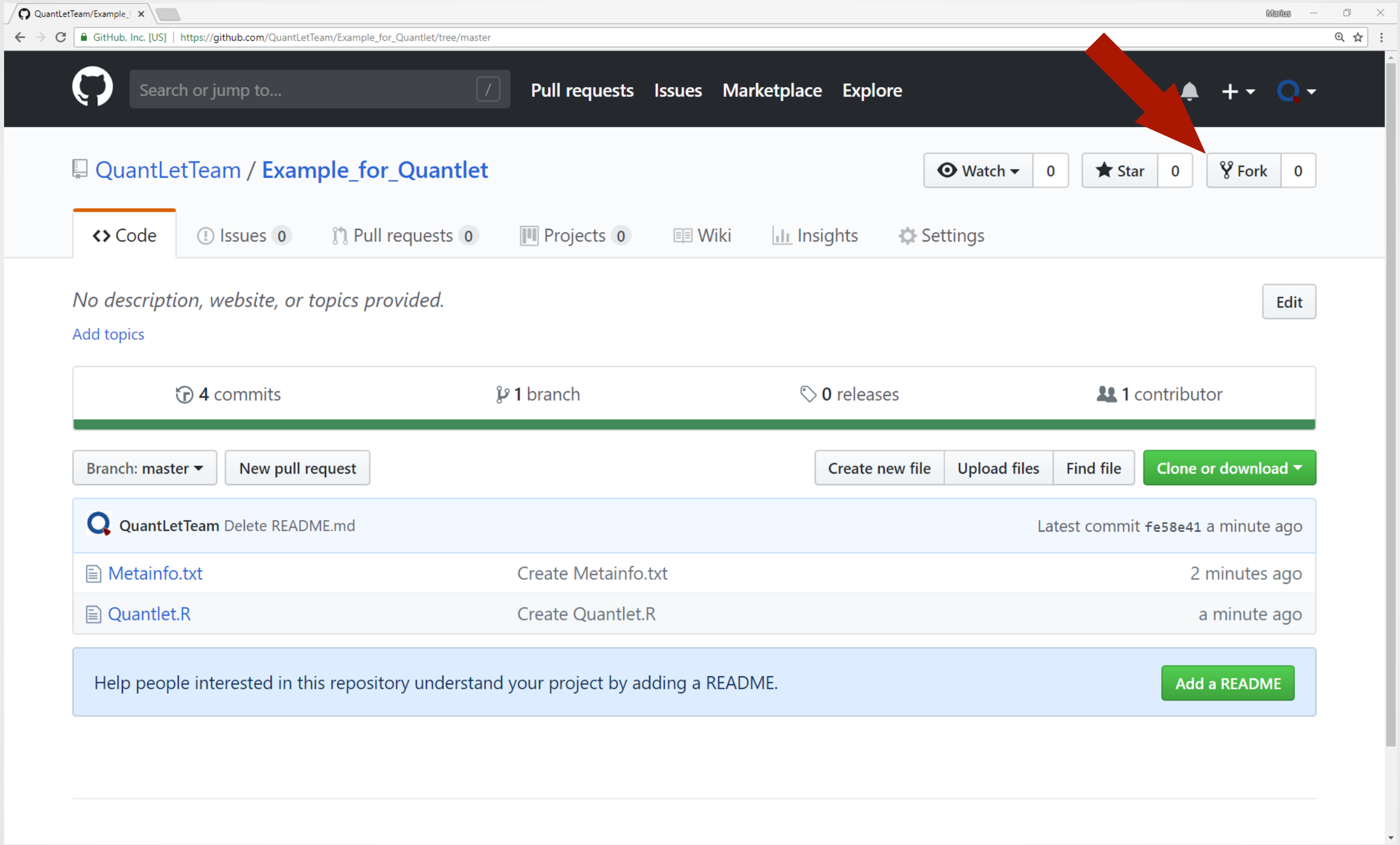


# Step 3: Create pull request 3

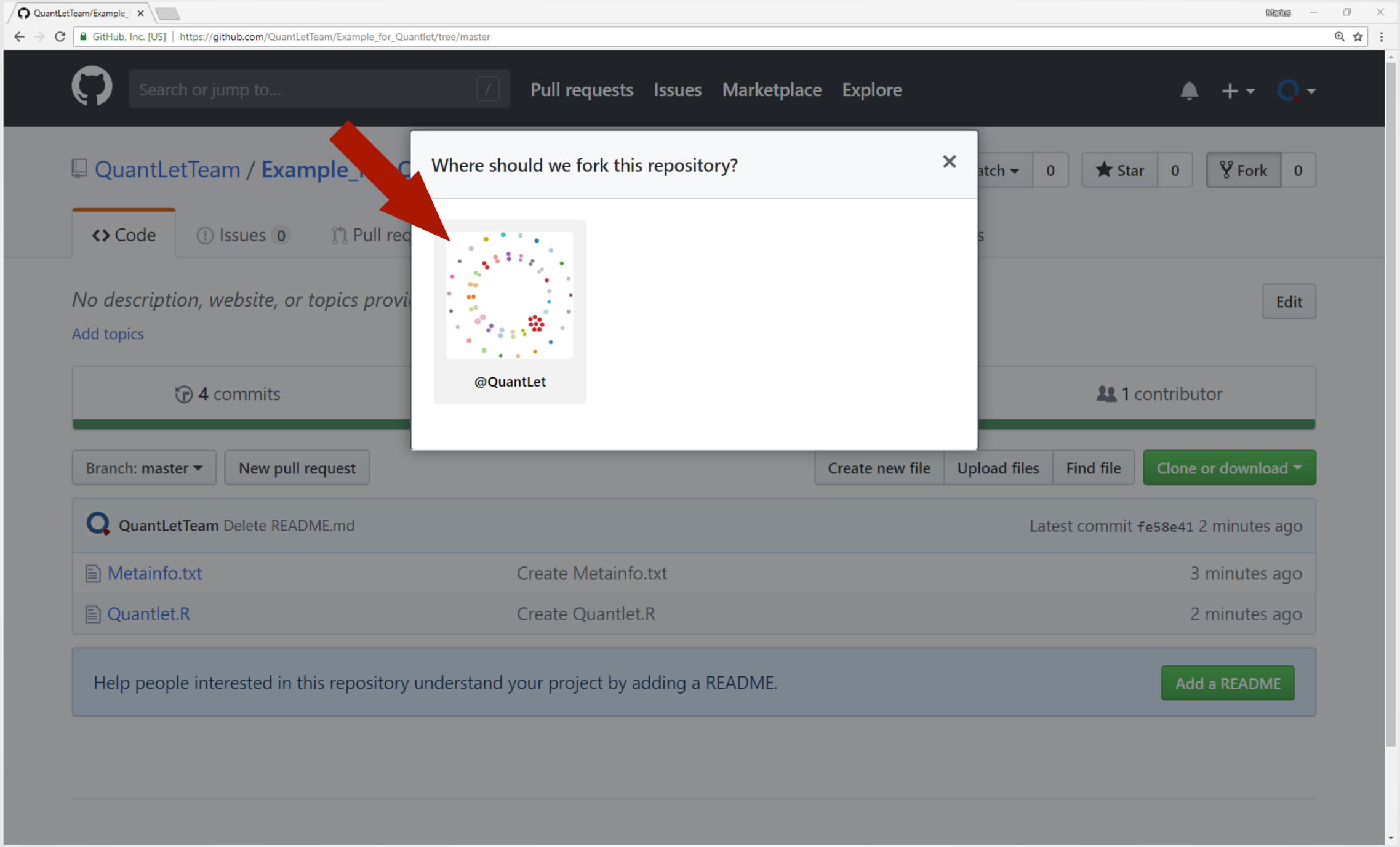
- The pull request is sent to the repository admin who accepts or denies the commit.
- If you want to change something after an accepted pull request, commit new changes to your forked repository and create a new pull request.



# Fork repository from your account to Quantlet 1



# Fork repository from your account to Quantlet 2

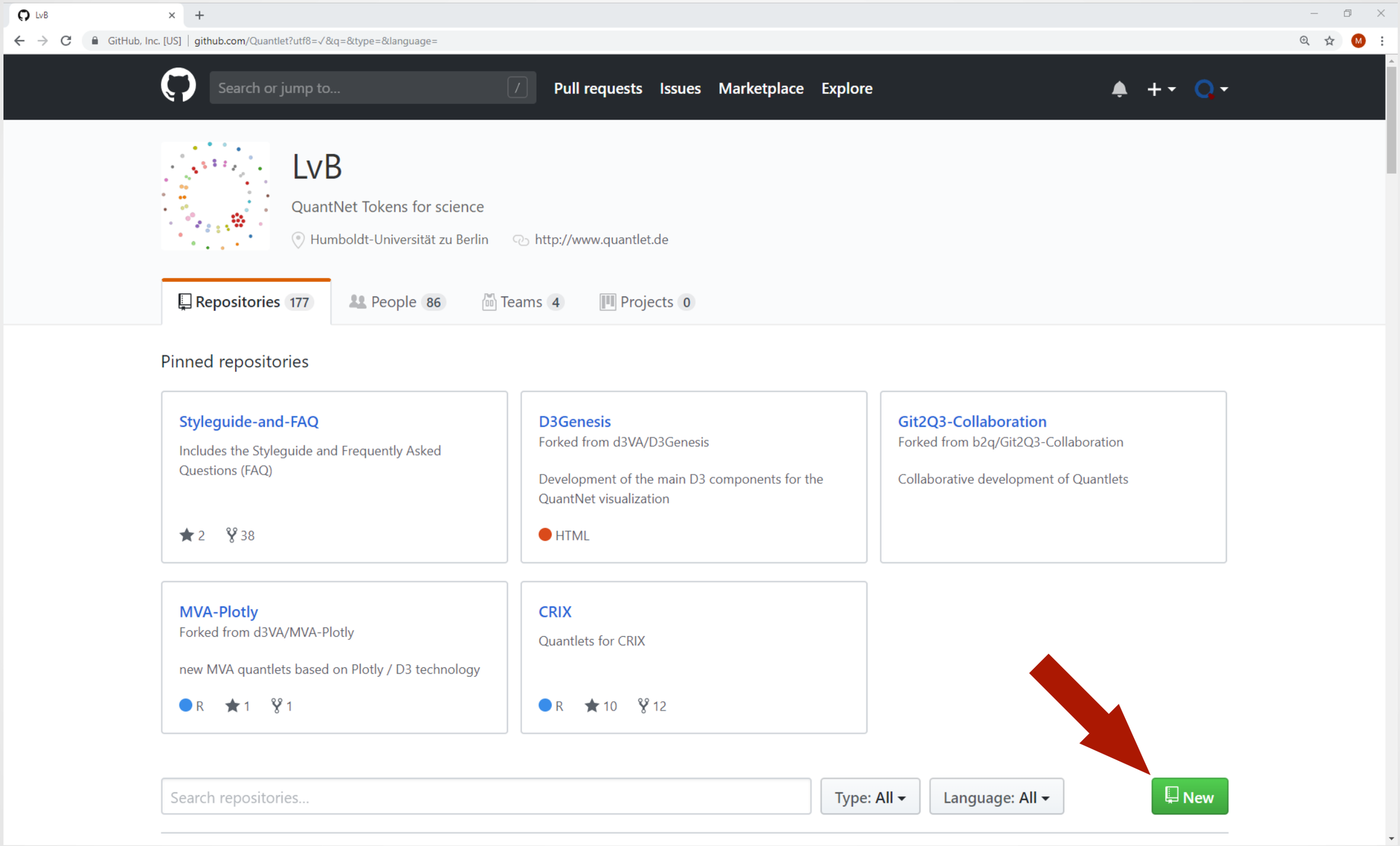


# Responsibilities

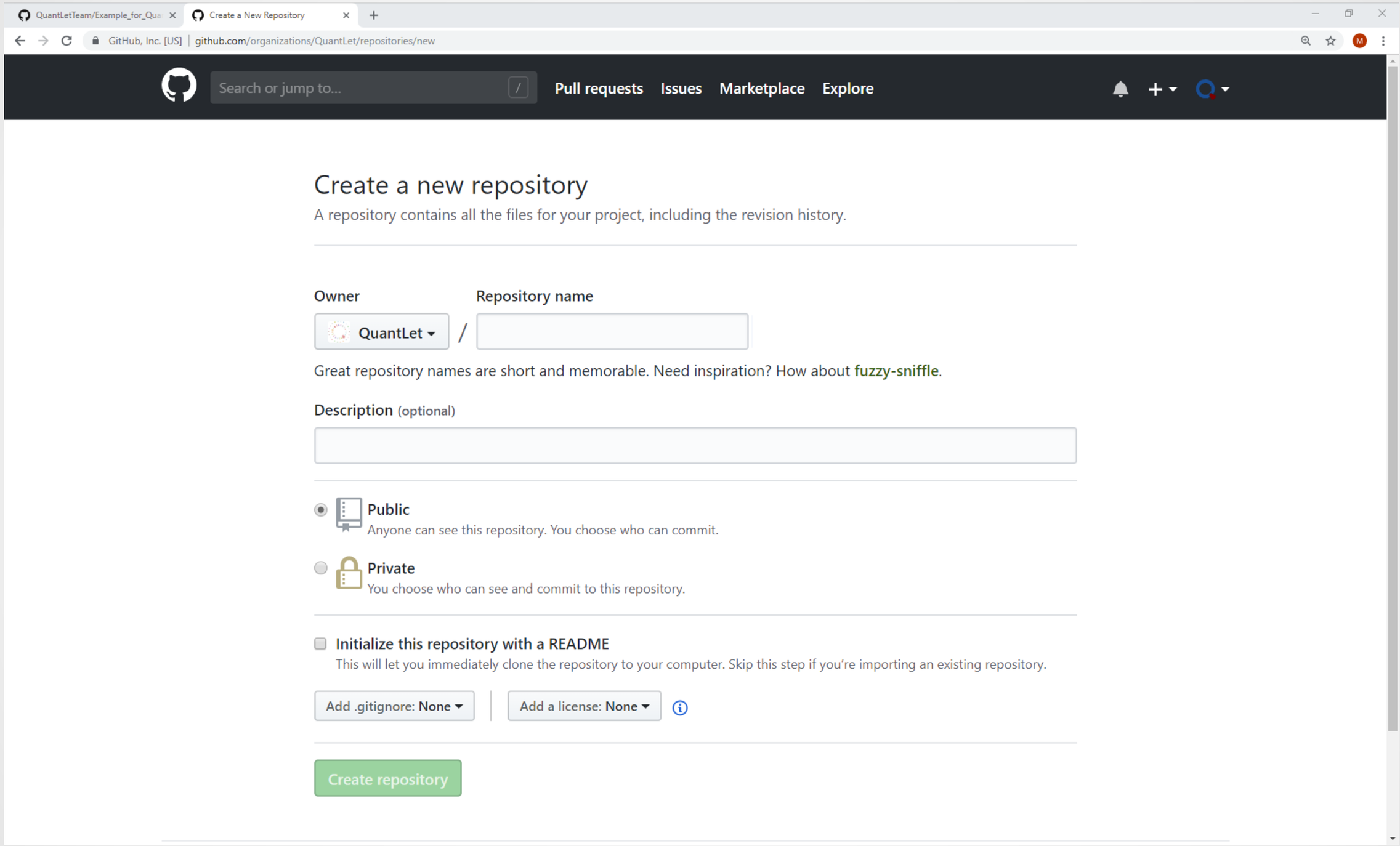
- When you fork a repository into the Quantlet organisation, you are its creator and thus automatically admin!
- This brings some responsibilities:
  - Make sure every submitted Quantlet contains a Metainfo file.
  - Accept / deny pull requests by others.
  - Manage access privileges.



# Create new repository 1



# Create new repository 2



# Responsibilities

- When create a repository in the Quantlet organisation, you are automatically its admin!
- This brings some responsibilities, see responsibilities in page ‘Responsibilities’ in the 2. Fork repositories to Quantlet.



# Outline

1. Styleguide for Quantlets
2. Submission Guide for Members
3. **Submission Guide for Non Members**
    - 3.1. Submission to an existing repository via pull request
    - 3.2. Fork of entire repository by a member of the Quantlet Organization

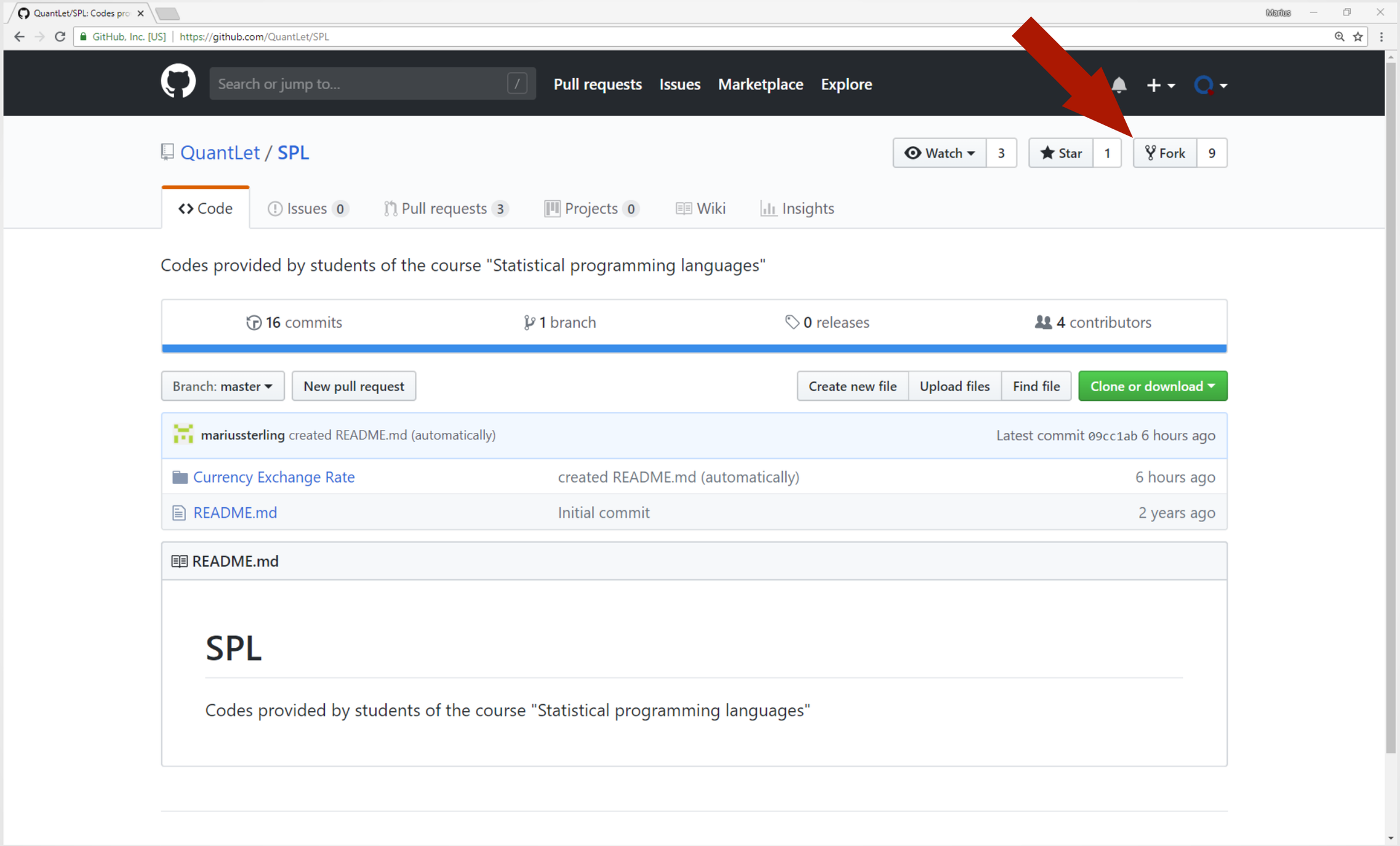


# Submission to existing repository via pull request

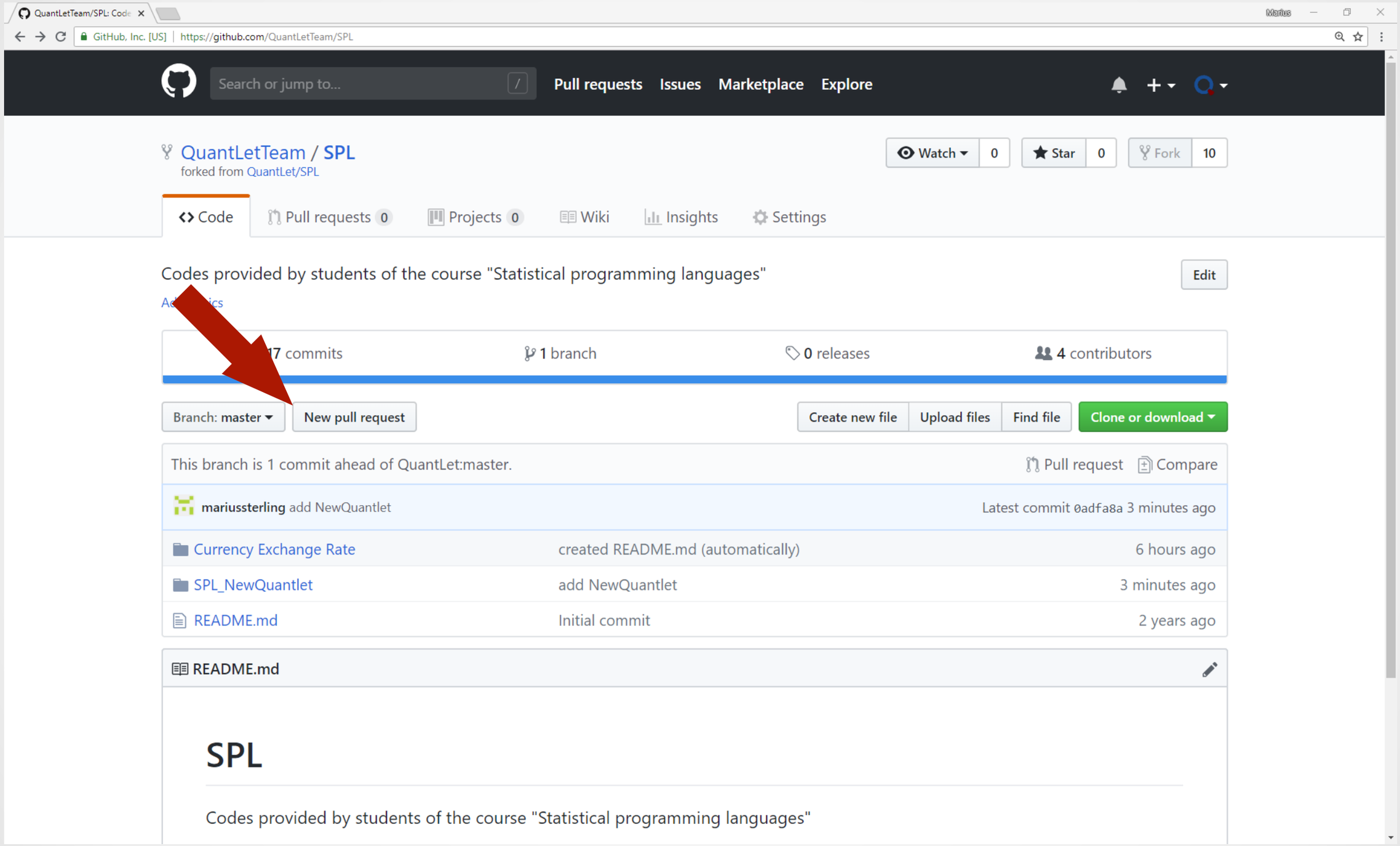
- Submitting a pull request
  - Step 1: Fork the target repository
  - Step 2: Commit changes
  - Step 3: Create a pull request
- The steps are explained on the following pages



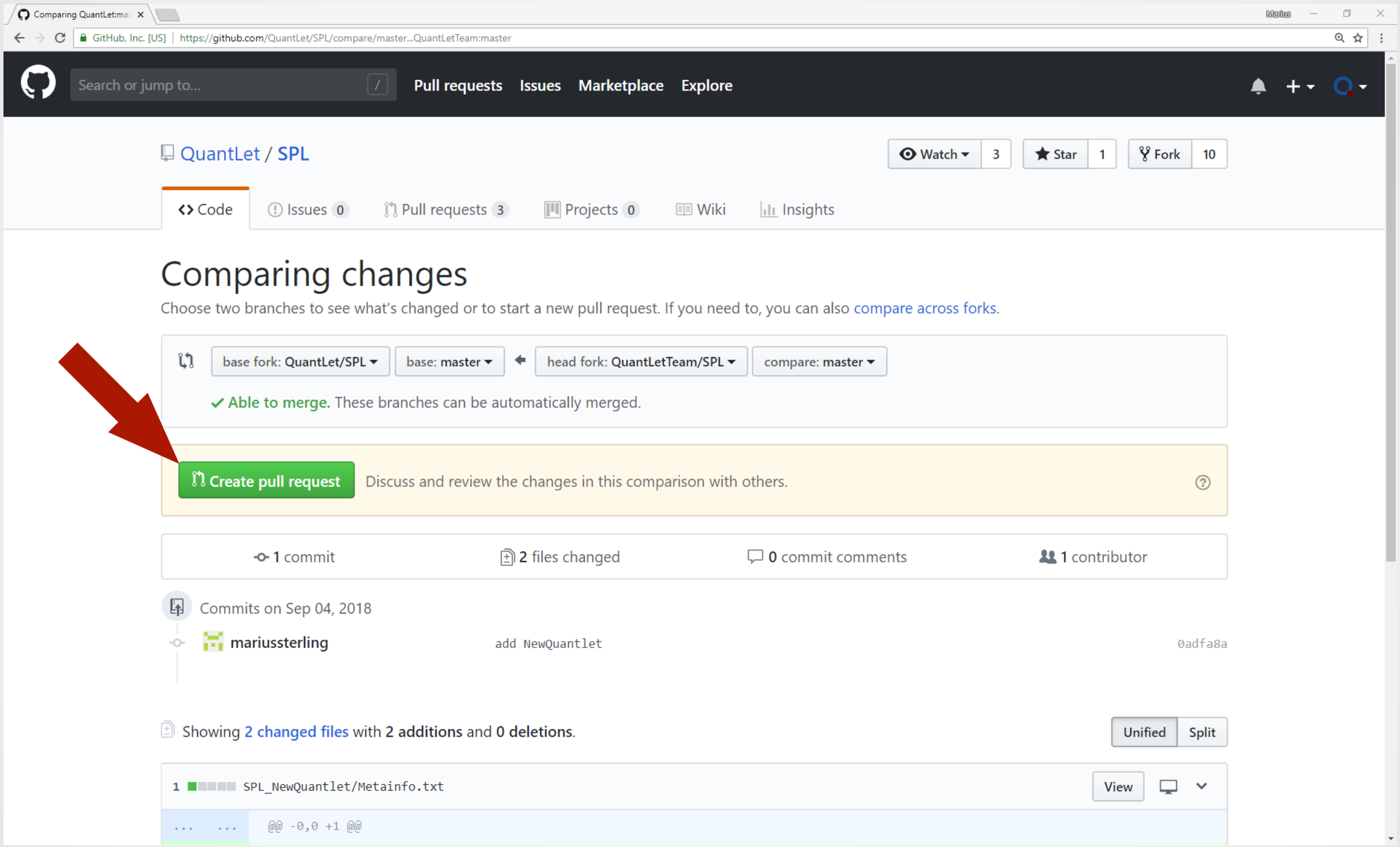
# Step 1: Fork the target repository



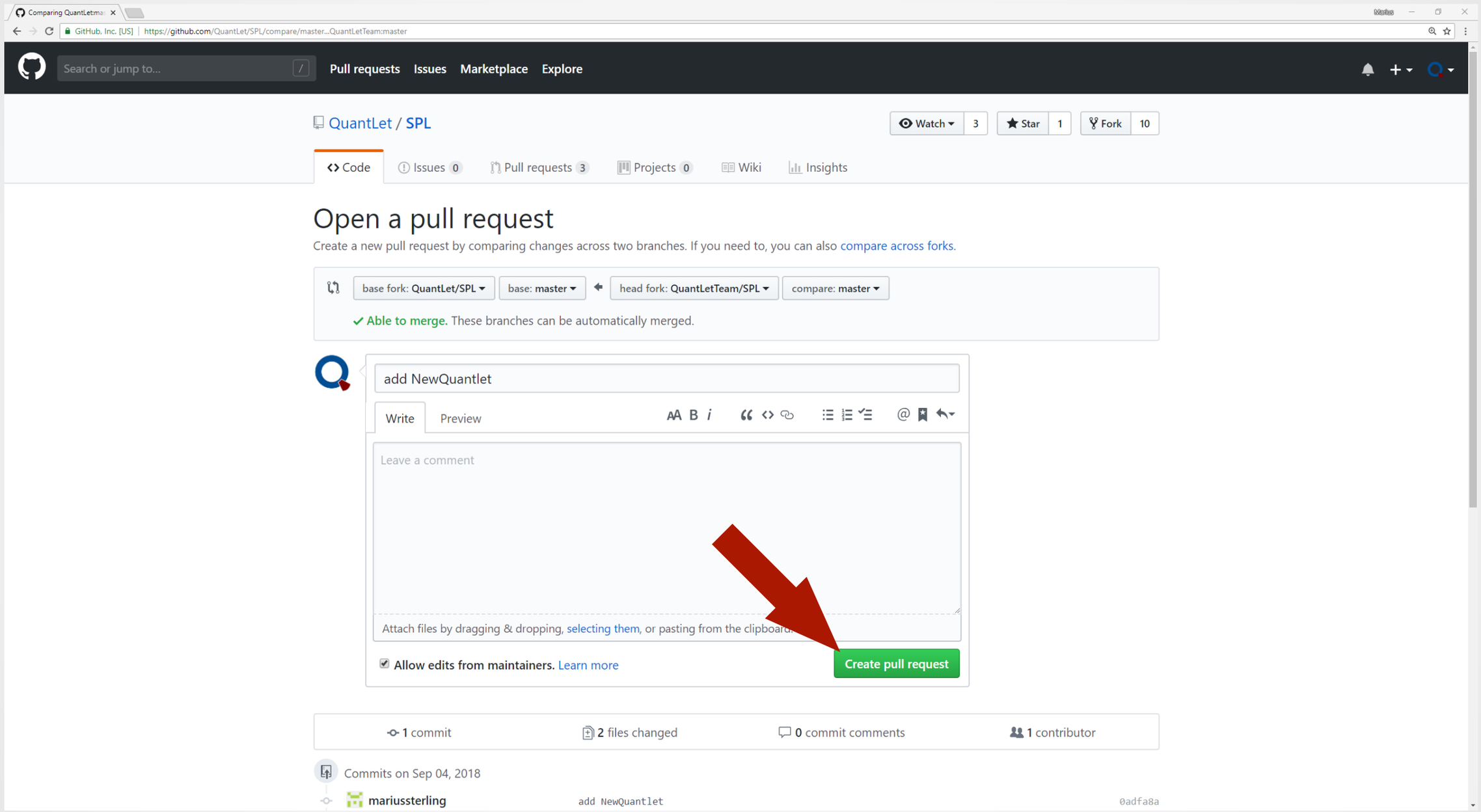
# Step 2: Commit changes



# Step 3: Create pull request 1



# Step 3: Create pull request 2



# Step 3: Create pull request 3

- The pull request is sent to the repository admin who accepts or denies the commit.
- If you want to change something after an accepted pull request, commit new changes to your forked repository and create a new pull request.



# Fork by a member of the Quantlet Organization 1

- Members of the Quantlet organisation can fork public repositories into the Quantlet organisation.
- Since only entire repositories can be forked, it is not possible to fork parts of repositories, e.g. Quantlets.
- If your Quantlet is part of project, lecture, exercise etc. ask your Quantlet contact to create a repository and use Submission to existing repository via pull request method.



# Fork by a member of the Quantlet Organization 2

- Steps:
  - Create in your own Github account a repository and commit your program to it.
  - Make sure that the repository and Quantlet names are not yet used in the Quantlet organisation.
  - Ask your Quantlet contact to fork your repository into it. If you do not have one please contact the Quantlet Team via quantlet.wiwi@hu-berlin.de.
  - All following changes can be submitted by the Submission to existing repository via pull request method.



# References

Borke L, Härdle WK (2018) Q3-D3-LSA, Handbook of Big data Analytics, (Härdle, Lu, Shen eds), Springer Verlag, ISBN 978-3-319-18284-1, DOI: 10.1007/978-3-319-18284-1






Härdle Hu Petukhina Sterling Zinovyeva

The Q Team

International Research Training Group 1792
"High Dimensional Nonstationary Time Series"

School of Business and Economics
Unter den Linden 6
10099 Berlin

E-Mail: quantlet.wiwi@hu-berlin.de