

Maintaining Translations with GitHub Desktop

Updated: 04-Jan-2025

By: Chris Jakeman

Contents

What are <i>Git</i> and <i>GitHub</i> ?	2
Maintenance Procedure	3
Initially:	3
Thereafter:	3
Using <i>GitHub Desktop</i>	4
Install <i>GitHub Desktop</i>	4
Create an account at GitHub.com.....	5
Maintenance Procedure	8
Initially:	8
1. Make your own remote repository (or repo) at GitHub	8
2. Download a copy (called a “clone”) of your remote repo	11
Thereafter:	15
3. Bring your repos up to date with the official repo at GitHub	15
4. Make and test your changes using your translation tools	18
5. Commit your changes to the local repo on your PC.....	19
6. Push your commit with all your changes	21
7. Submit your commit by making a “pull request”	21

What are *Git* and *GitHub*?

Git is a free version control system designed to track digital projects. It was created to support the development of Linux and has become the leading version control system. (Microsoft has switched from their in-house *Team Services* to *Git* and now uses it for Windows development.)

GitHub is the world's largest hosting service for software, both public and private projects. Digital assets such as Open Rails routes benefit just as much as software from this tracking capability. *GitHub* is now owned by Microsoft.

Git is a small program that you can install on your PC to keep each version of your project in a repository (or repo) where it is safe from changes. The free *GitHub Desktop* program includes a version of *Git* and also communicates between your PC and a remote repository in your account at GitHub.com. (Note that *GitHub Desktop* is for 64-bit Windows only.¹)

Git helps a translator to work independently or with partners.

¹ For 32-bit Windows, many other graphical programs are available free (such as SourceTree) and Git can also be used from the command line using Git Bash.

Maintenance Procedure

The procedure to maintain translations is:

Initially:

1. Make your own remote repository (or repo) at GitHub as a special copy (called a “fork”) of the official Open Rails remote repo. Git remembers where your repo was forked from and provides a mechanism to check and adopt your changes back into the official repo.

If you are collaborating with someone else then this is the repo that you share and Git coordinates your changes so none get lost or overwritten.

2. Download a copy (call a “clone”) of your remote repo to make a local repo on your PC. Once again, Git remembers where your local repo was cloned from and makes it easy to push your local changes back to your remote repo.

Thereafter:

3. Bring your repos up to date with the official repo at GitHub. Of course, you won’t need to do this the very first time.

4. Make and test your changes using your translation tools.

5. Commit your changes to the local repo on your PC. This is a permanent record which keeps your work safe and so you can safely and easily reverse or repeat the changes.

6. Push your commit with all your changes back to your remote repo at GitHub.

7. Submit your commit by making a “pull request” to the official repo at GitHub.

They will be checked by including them automatically in a temporary build just in case there is a problem when compiling.

The commit will then be reviewed by another member of the team and adopted by “pulling” the changes from your remote repo to the official repo. This is the reason that a submitted change is called a “pull request”.

Using *GitHub Desktop*

Install *GitHub Desktop*

Browse to <http://desktop.github.com>, click on the *Download* button to download the installer, then run the installer. Choose the free option:

Individuals Teams


Free
\$0 USD

The basics of GitHub for every developer

Choose Free


Pro
\$7 USD
Per month

Pro tools for developers with advanced requirements

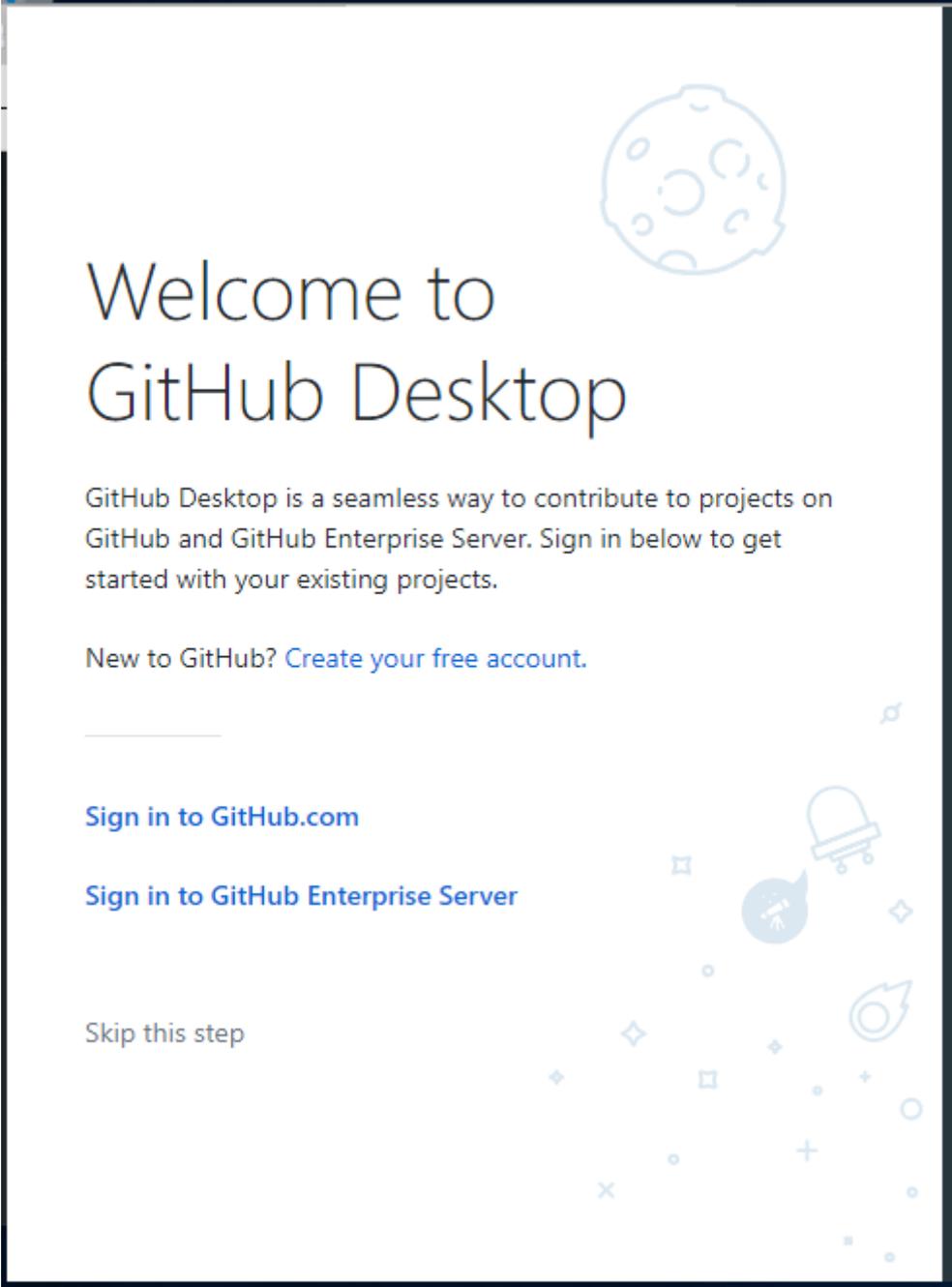
Choose Pro

- ✓ Unlimited public repositories
- ✓ Unlimited private repositories
- ✓ **Limited to 3 collaborators** for private repositories
- ✓ 2,000 total Action minutes/month
See pricing details
- ✓ 500MB of GitHub Packages storage
See pricing details
- ✓ Advanced vulnerability scanning for public repositories
- ✓ Automated security updates
- ✓ GitHub Security Advisories
- ✓ Issues and bug tracking
- ✓ Project management

← Includes everything in Free

- ✓ Unlimited collaborators
- ↑ 3,000 total Action minutes/month
See pricing details
- ↑ 1GB of GitHub Packages storage
See pricing details
- ✓ Private GitHub Pages and Wikis
- ✓ Private protected branches
- ✓ Code owners
- ✓ Repository insights

Create an account at [GitHub.com](https://github.com)



The image shows the GitHub Desktop welcome screen. At the top right is a circular icon with a face and spots, resembling a moon or a planet. Below it, the text reads "Welcome to GitHub Desktop". Underneath is a paragraph: "GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise Server. Sign in below to get started with your existing projects." This is followed by a link: "New to GitHub? [Create your free account.](#)". Below that is a horizontal line. Then there are two sign-in options: "Sign in to [GitHub.com](#)" and "Sign in to [GitHub Enterprise Server](#)". At the bottom left is the text "Skip this step". On the right side of the screen, there is a decorative graphic of various icons including a lightbulb, a rocket, a magnifying glass, and several geometric shapes like diamonds and squares, all in a light blue color.

Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise Server. Sign in below to get started with your existing projects.

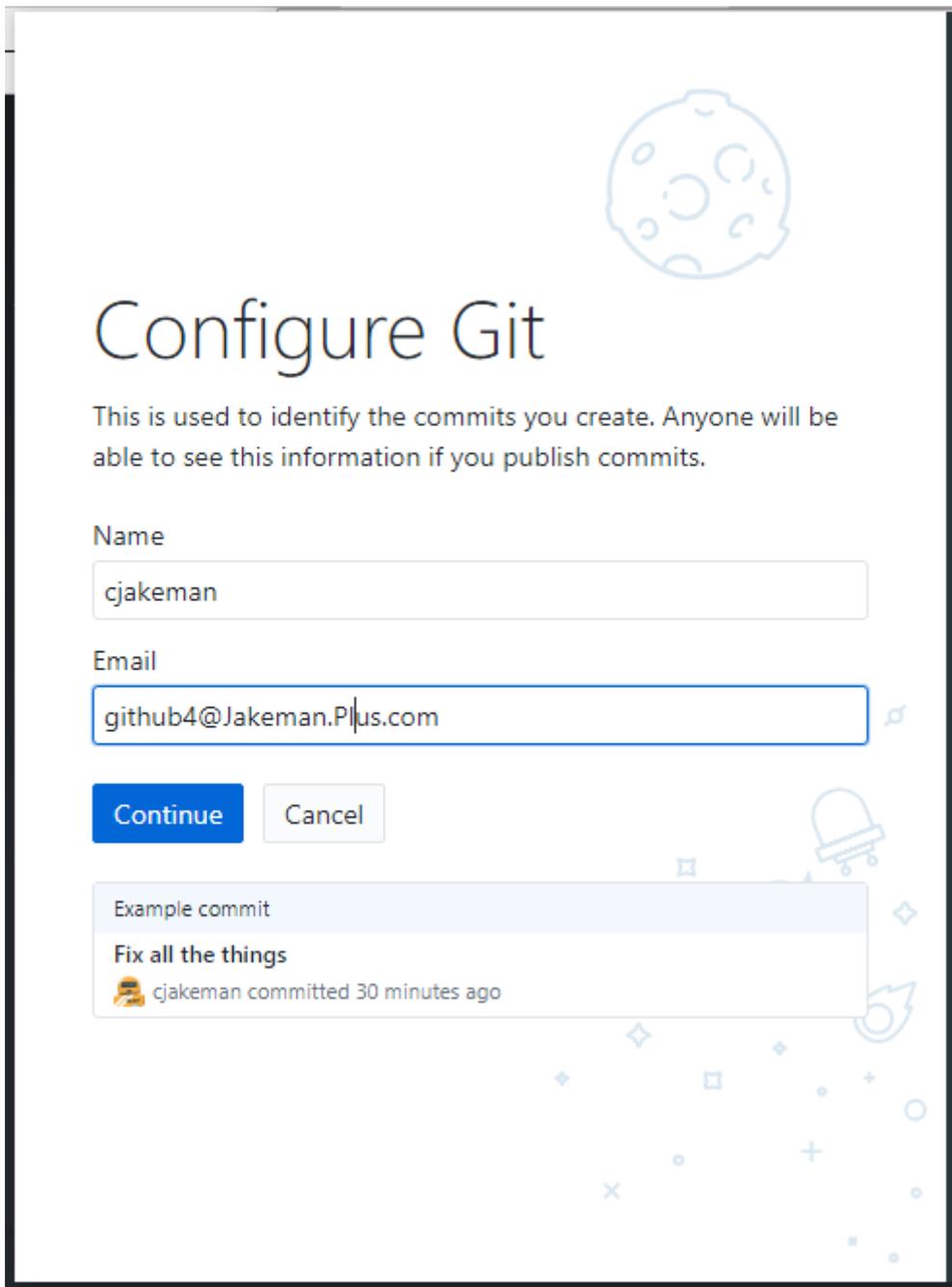
New to GitHub? [Create your free account.](#)

[Sign in to GitHub.com](#)

[Sign in to GitHub Enterprise Server](#)

[Skip this step](#)

Provide details:



The image shows a 'Configure Git' dialog box with a light blue background and a moon icon in the top right. The title 'Configure Git' is in a large, dark font. Below the title is a paragraph explaining that this information is used to identify commits and is visible if published. There are two input fields: 'Name' with the value 'cjakeman' and 'Email' with the value 'github4@Jakeman.Plus.com'. Below the fields are 'Continue' and 'Cancel' buttons. At the bottom, there is a preview of a commit message: 'Example commit', 'Fix all the things', and 'cjakeman committed 30 minutes ago' with a user icon. The background features faint icons of a rocket, a lightbulb, and various geometric shapes.

Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

Name

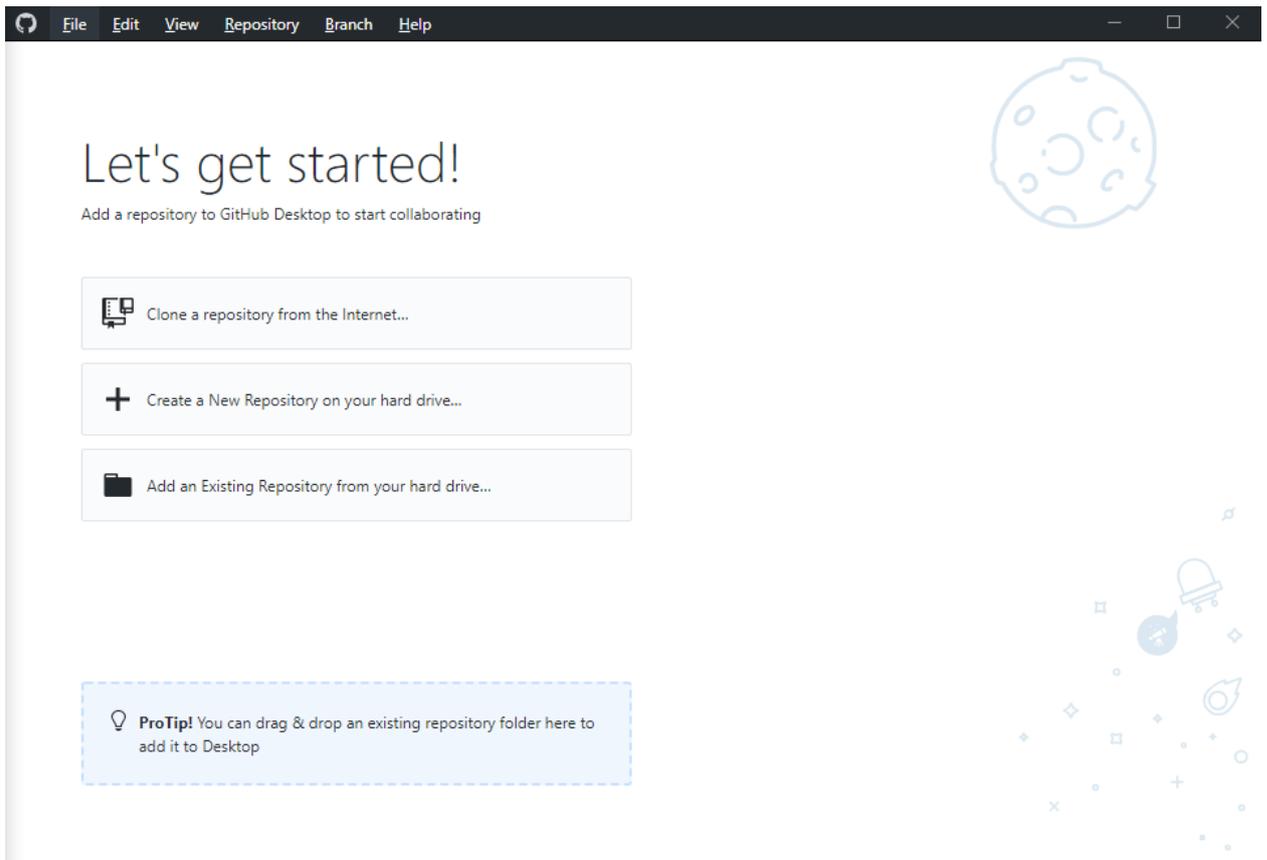
Email

Example commit

Fix all the things

 cjakeman committed 30 minutes ago

Finally, *GitHub Desktop* opens its launch page:



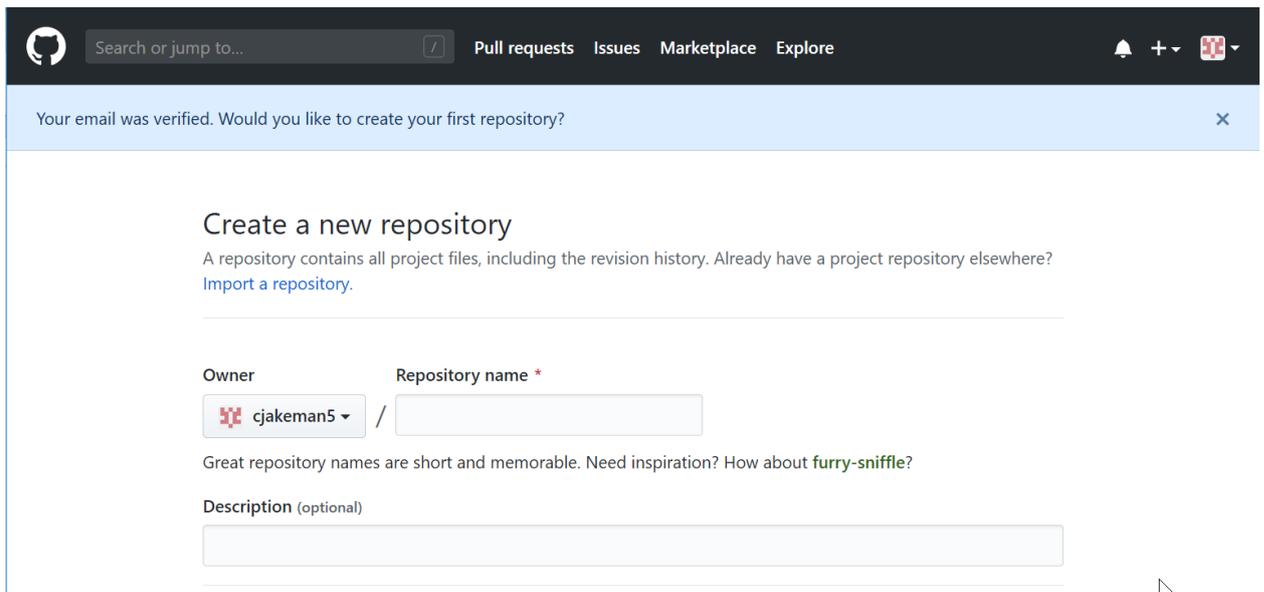
Maintenance Procedure

Initially:

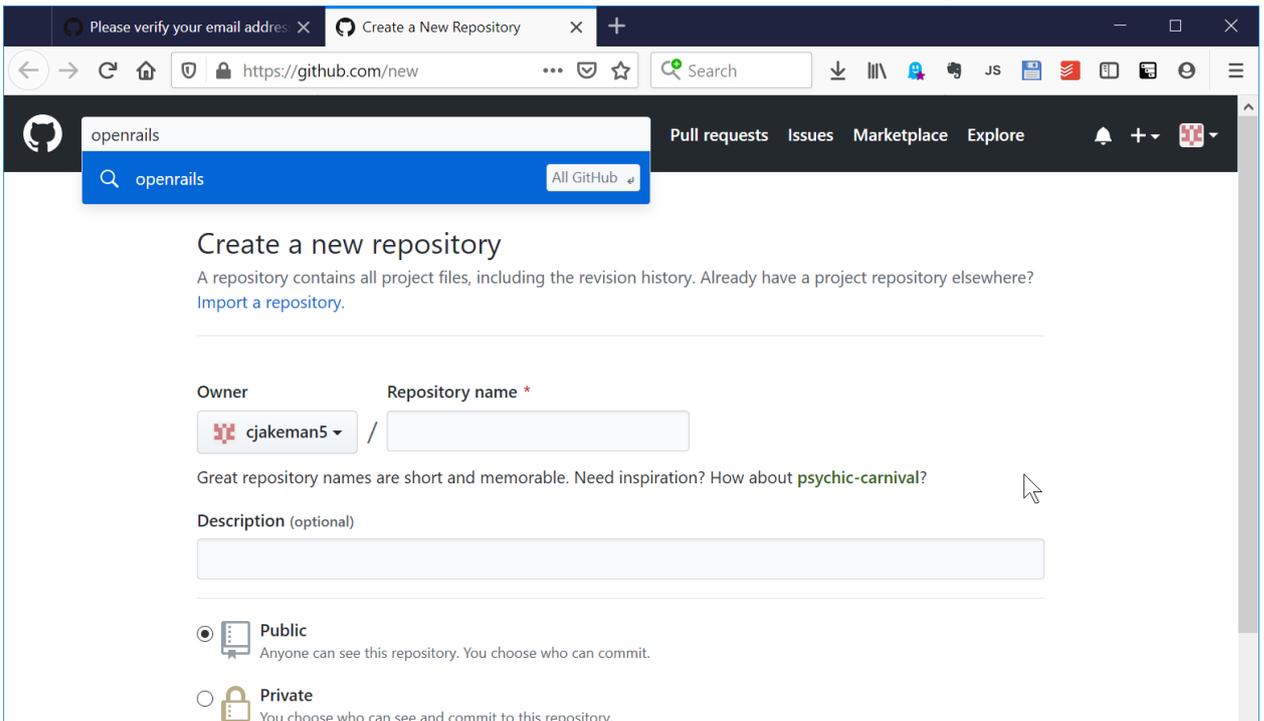
1. Make your own remote repository (or repo) at GitHub

This will be a special copy (called a “fork”) of the official Open Rails remote repo. Git remembers where your repo was forked from and provides a mechanism to check and adopt your changes back into the official repo.

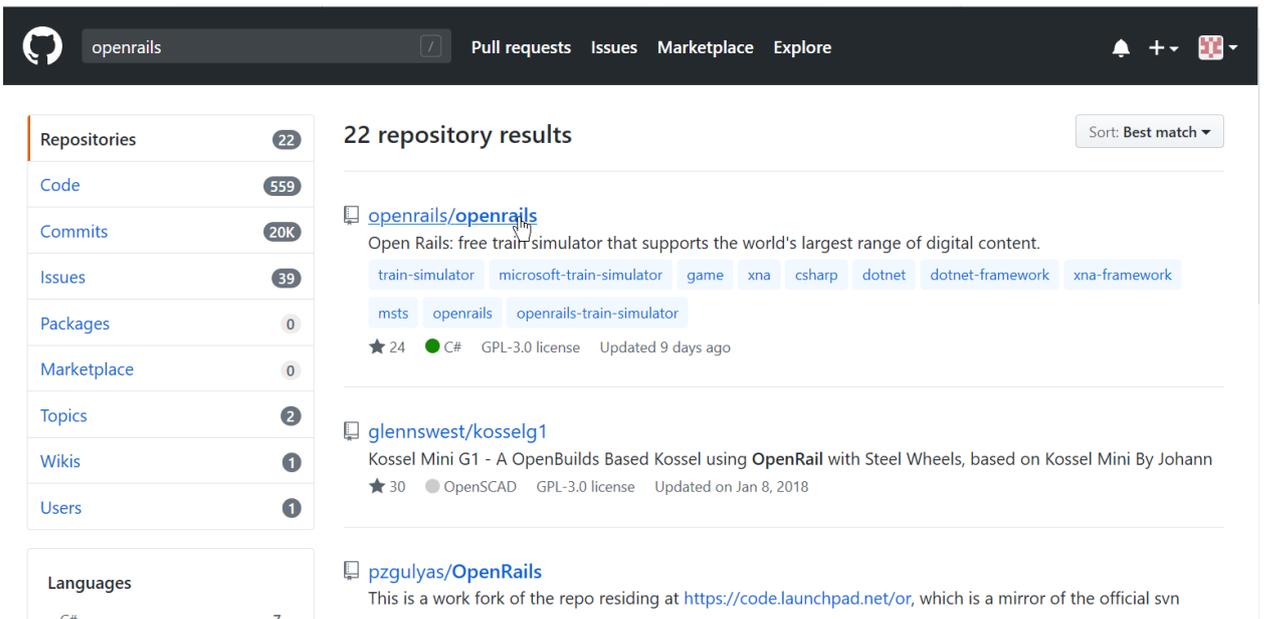
Register at www.GitHub.com and sign in.



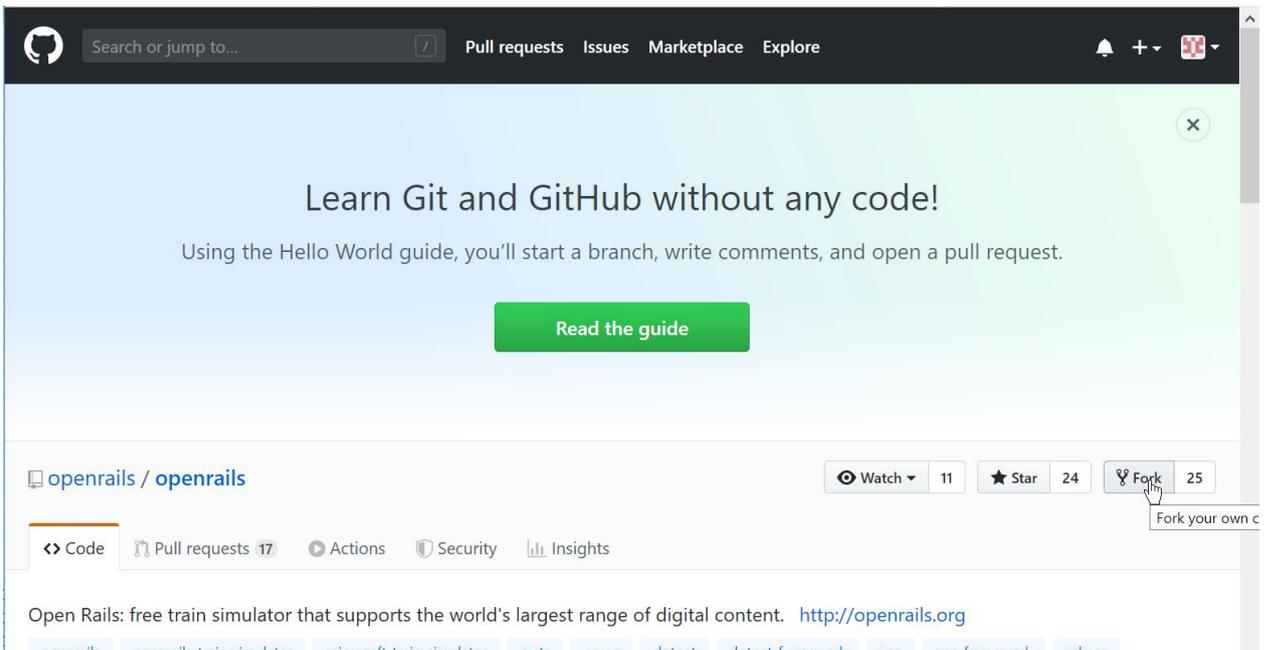
Do not create a new repository. Instead, we want to “fork” the official Open Rails repo, so search for “openrails” or browse directly to <https://github.com/openrails/openrails>:



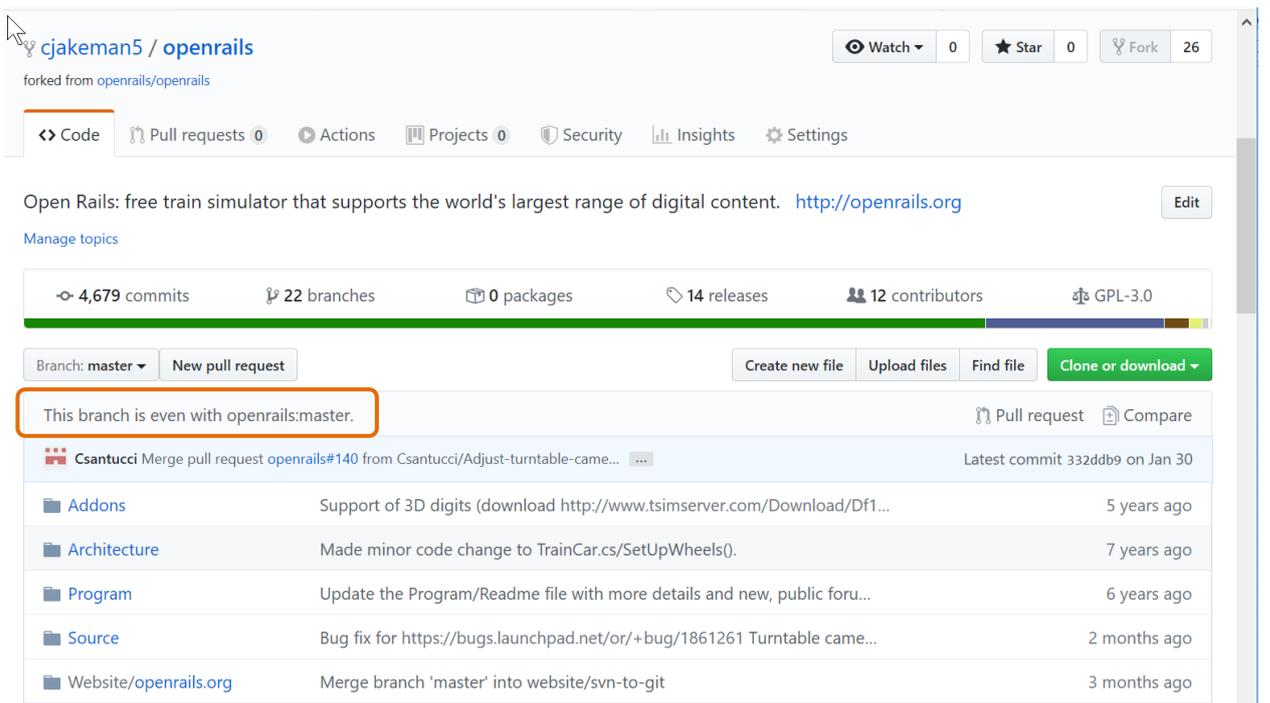
The search will find multiple results for “openrails”, so we click on the official one:



and end up here:



Press the “fork” button and GitHub will copy the official repo to give you a remote repo which remembers from where it was copied.

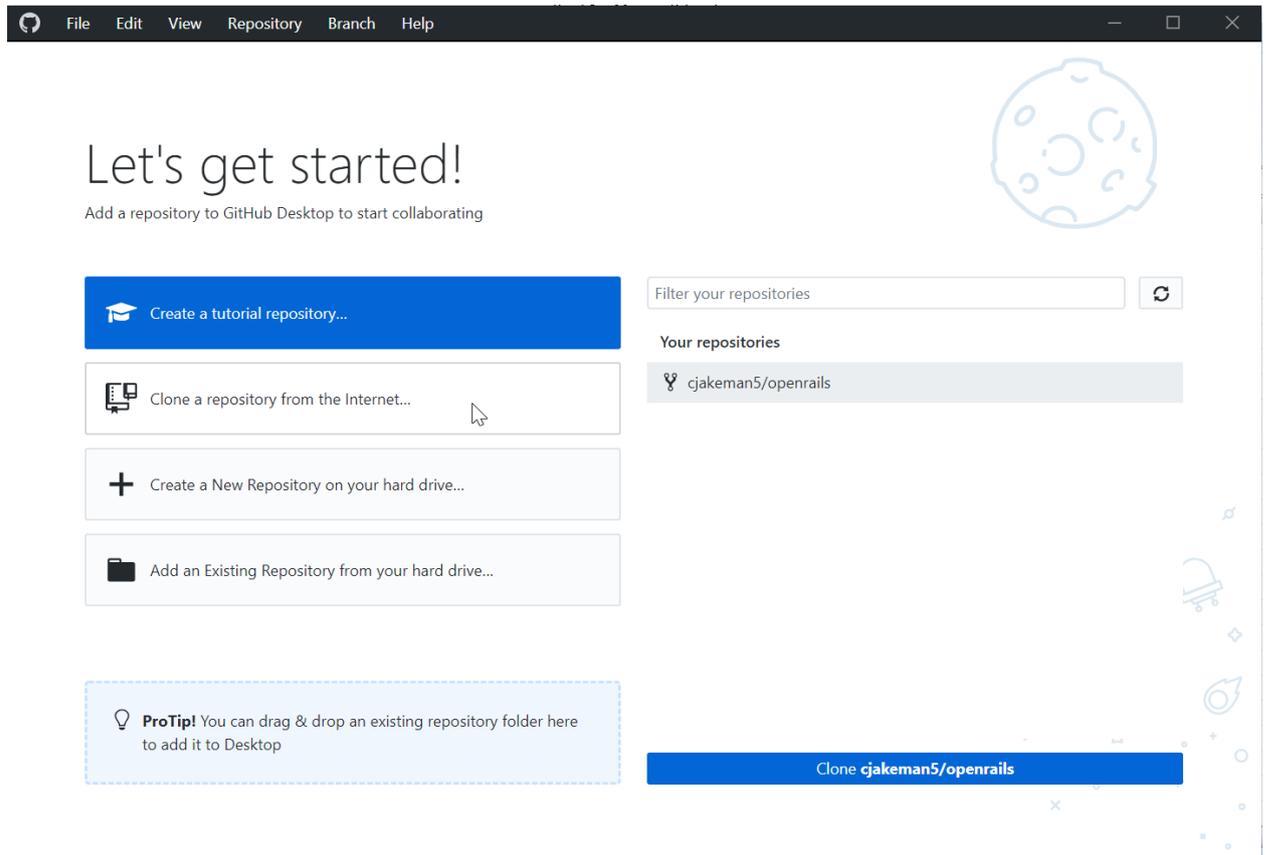


The message “even” shows that your remote repo is up to date.

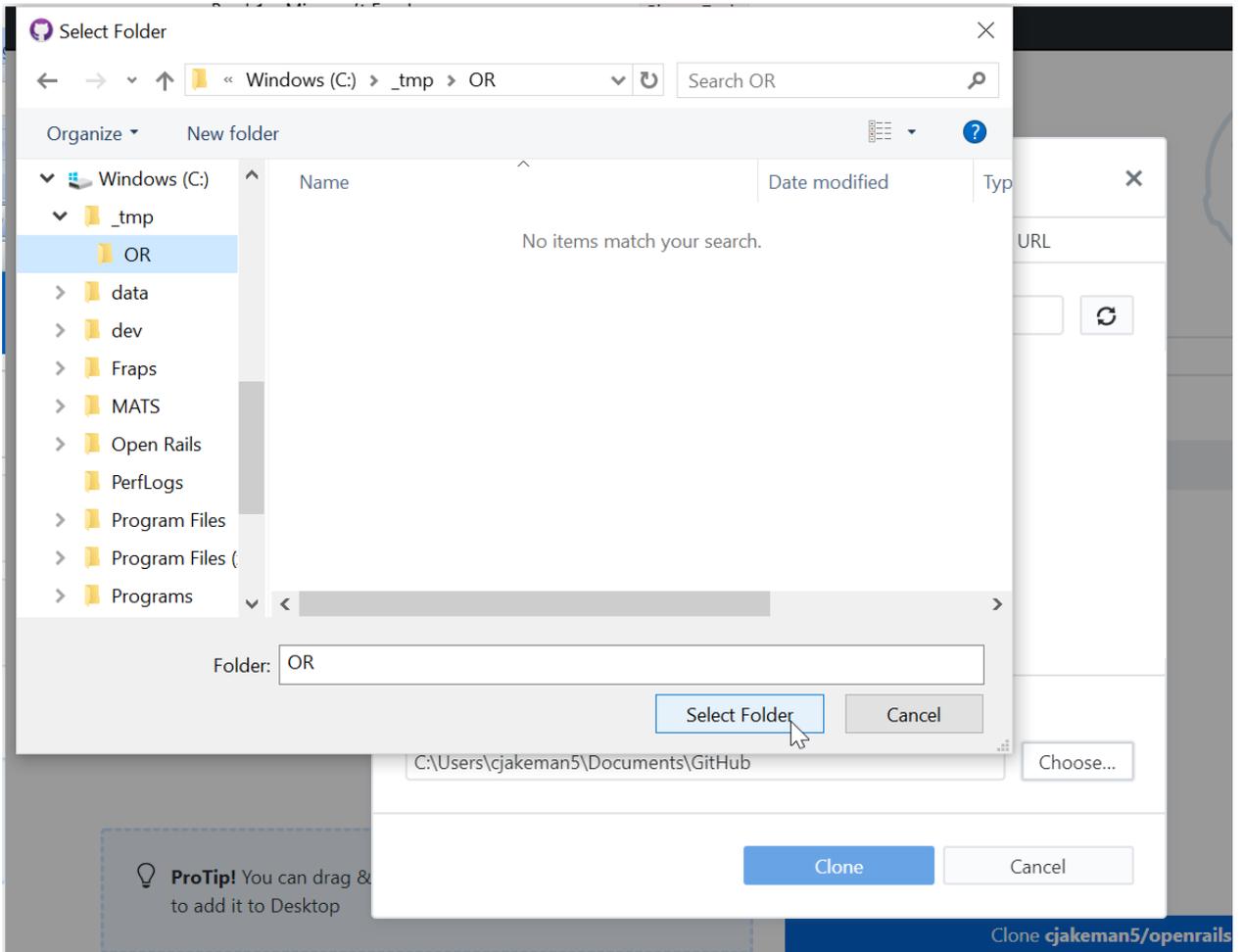
2. Download a copy (called a “clone”) of your remote repo

Now return to *GitHub Desktop* where you can “clone” your remote repo from GitHub.com to a local repo on your PC’s disk.

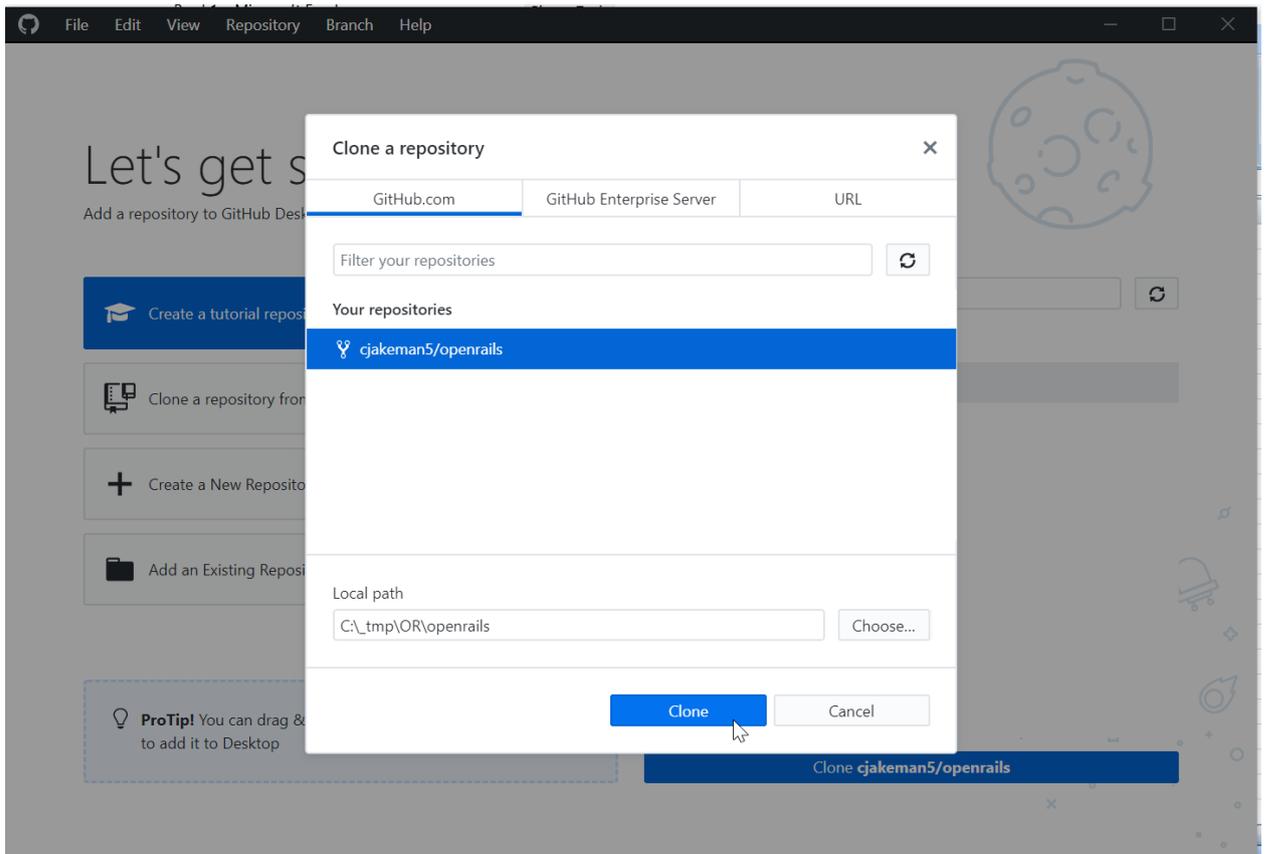
Once again, Git remembers where your local repo was cloned from and makes it easy to push your local changes back to your remote repo.



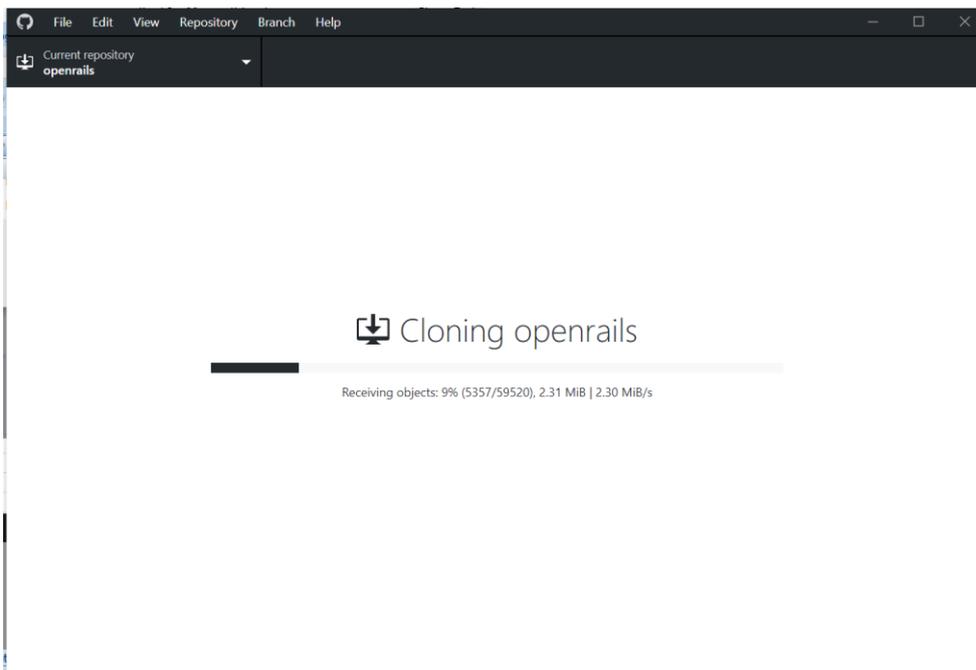
Use *Local path* > Choose to select an empty folder on your PC to hold the local repo:



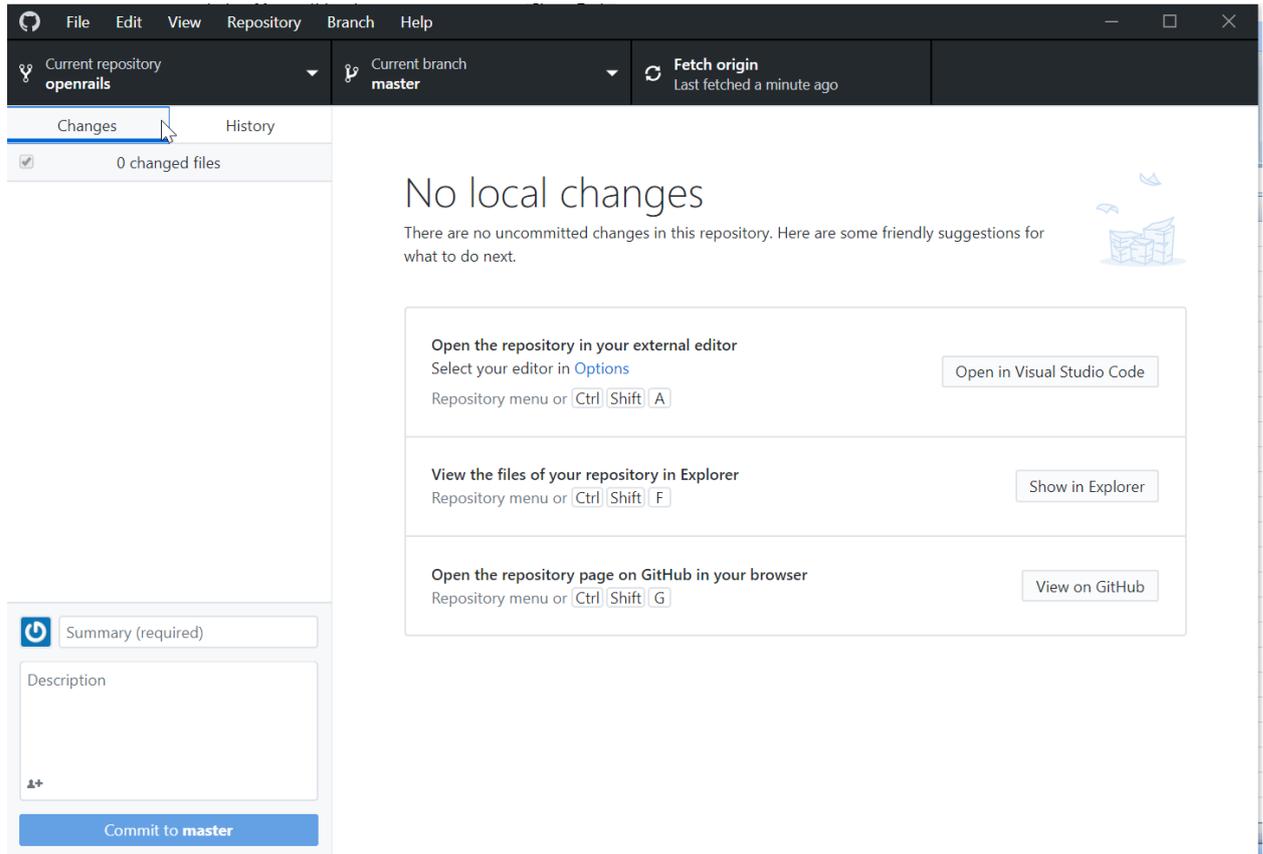
GitHub Desktop will find your remote repo and offer that as “Your repositories”:



Downloading the remote repo may takes some minutes. Subsequent uploads and downloads will be quick as they only transfer the changes.



and finally, the system is complete and ready for translation work:

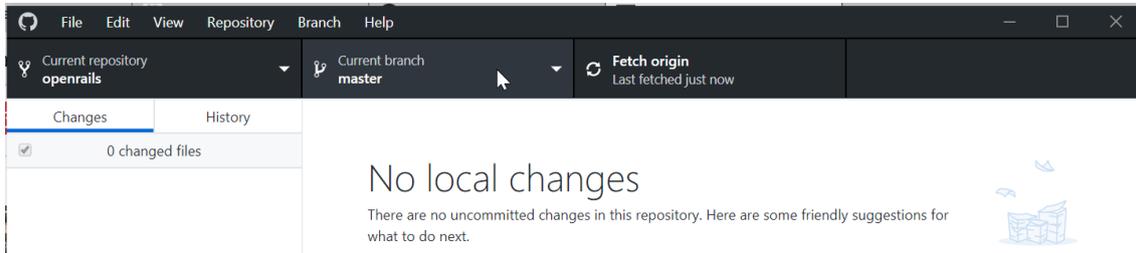


Thereafter:

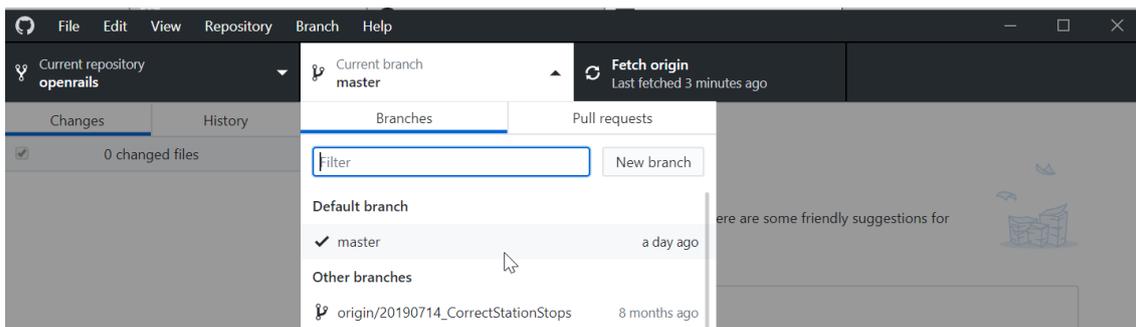
3. Bring your repos up to date with the official repo at GitHub

Of course, you won't need to do this the very first time.

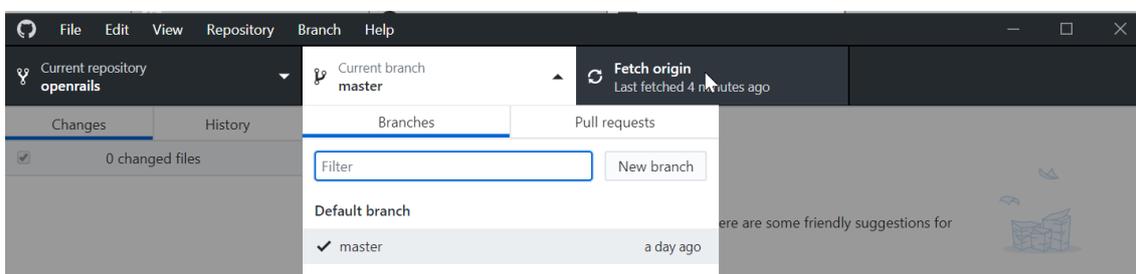
From *GitHub Desktop*, click on the *Current Branch* tab



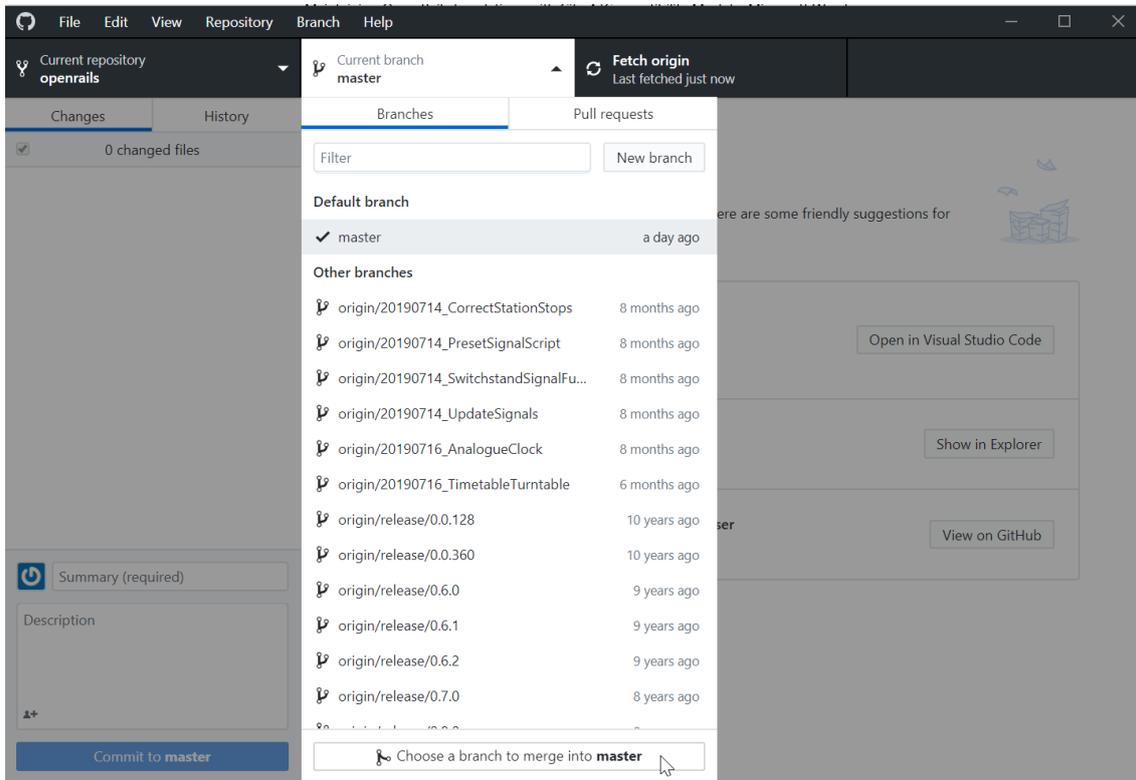
Make sure that the current branch is the “master” branch as we will be updating that from the “master” branch of the official (or “upstream”) repo.



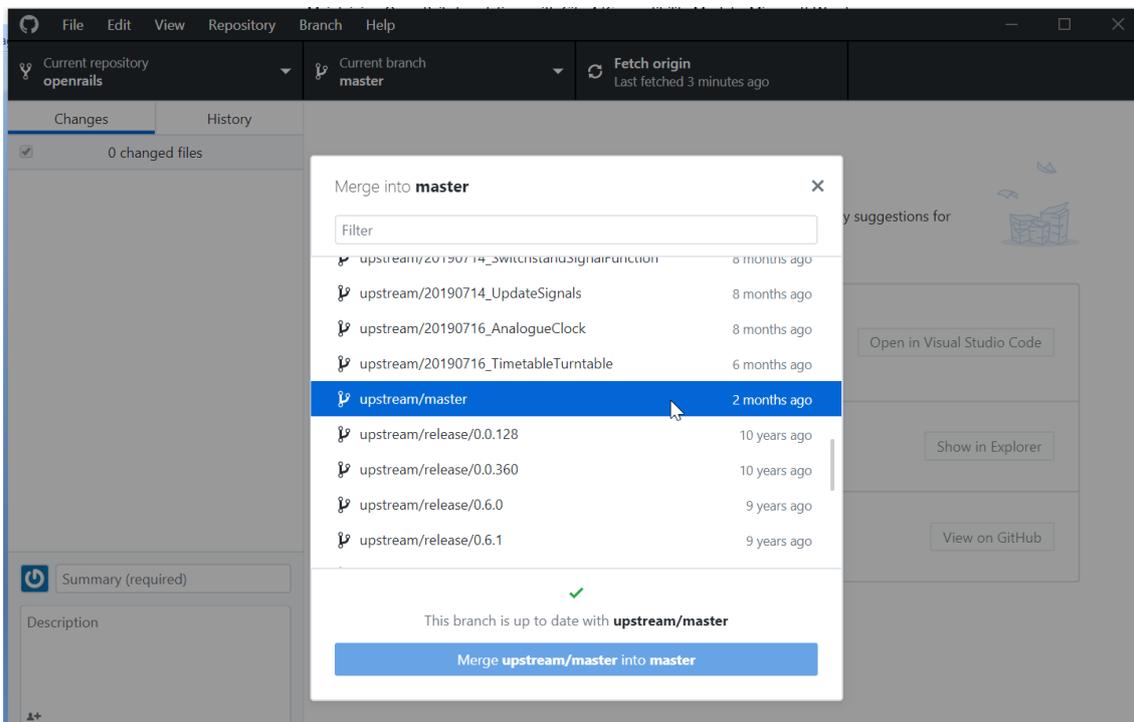
Then click on the *Fetch origin* tab, which downloads any waiting changes from your remote repo (“origin”) and also the official repo (“upstream”).



Then return to the *Current branch* tab and pick the button at the bottom:



Then scroll down the list of remote branches. This will be the branch from which changes will be merged into your local “master” branch. You should pick the branch called “upstream/master”



If the branch is not up-to-date, then:

1. click the blue button at the bottom to adopt the changes
2. You should also copy these changes to your remote repo so, from the *Changes* tab, press the “Push origin” button to transfer the changes to GitHub.com.

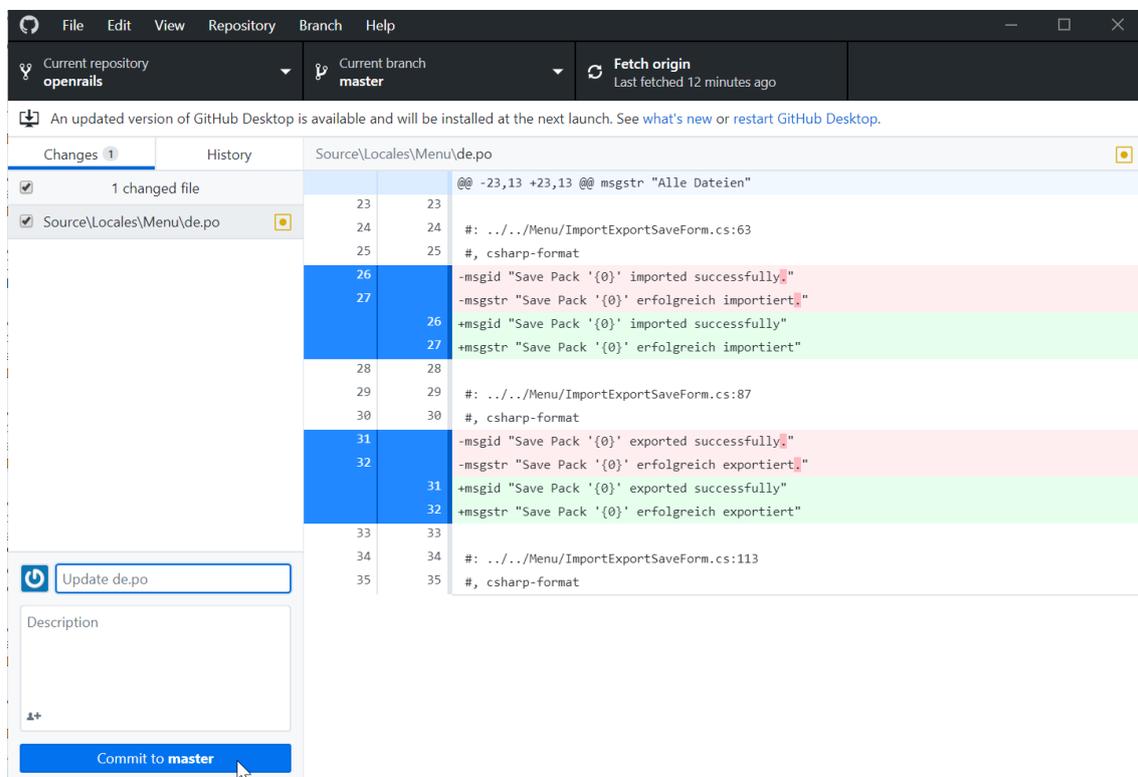
4. Make and test your changes using your translation tools

Follow the instructions in [Translating Open Rails](#)².

The translation work that you do will change files and possibly add new ones.

GitHub Desktop will track every file in your project that has been added or changed and prompt you to store them locally and securely with a “commit” operation. This gives you a version which you can always go back to.

For example, changes to the German file for Open Rails’ Menu program will be automatically detected once the file is saved. *GitHub Desktop* highlights small changes, provides a default description for the commit and the button *Commit to master* becomes enabled.:



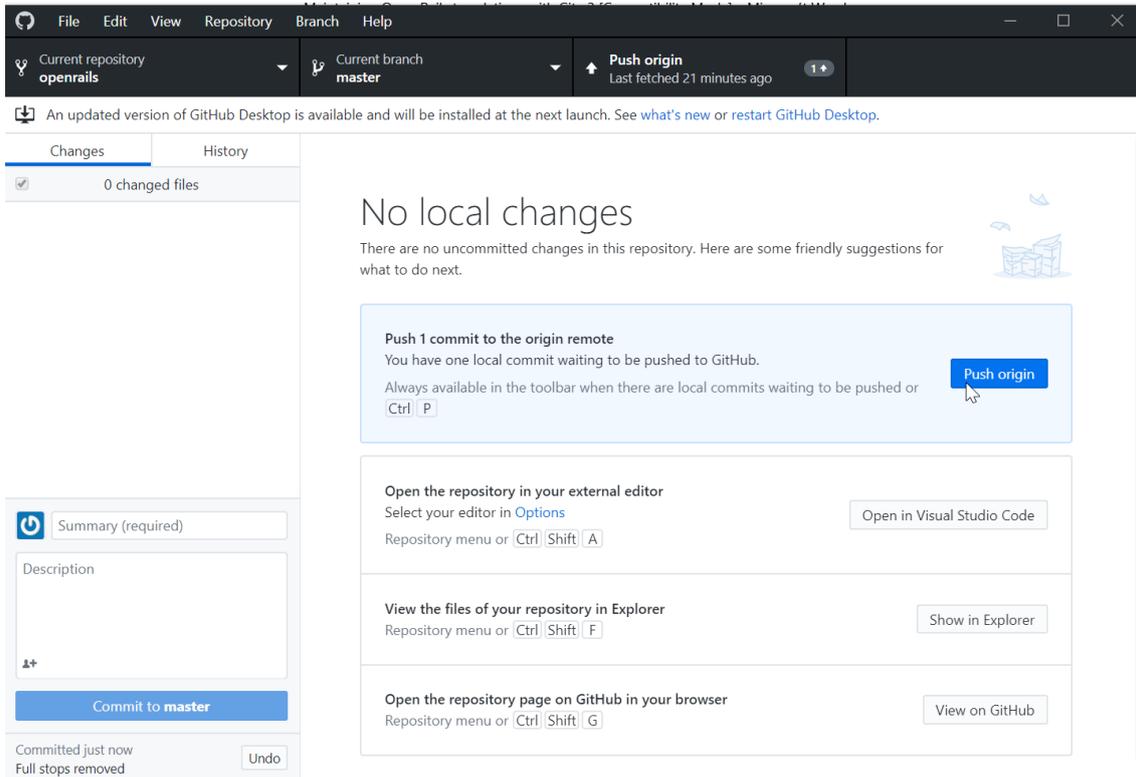
“master” is the branch which is current and to which new commits will be made. You should not need to change this for translation work.

² Download from <https://static.openrails.org/files/Translating-Open-Rails.pdf>

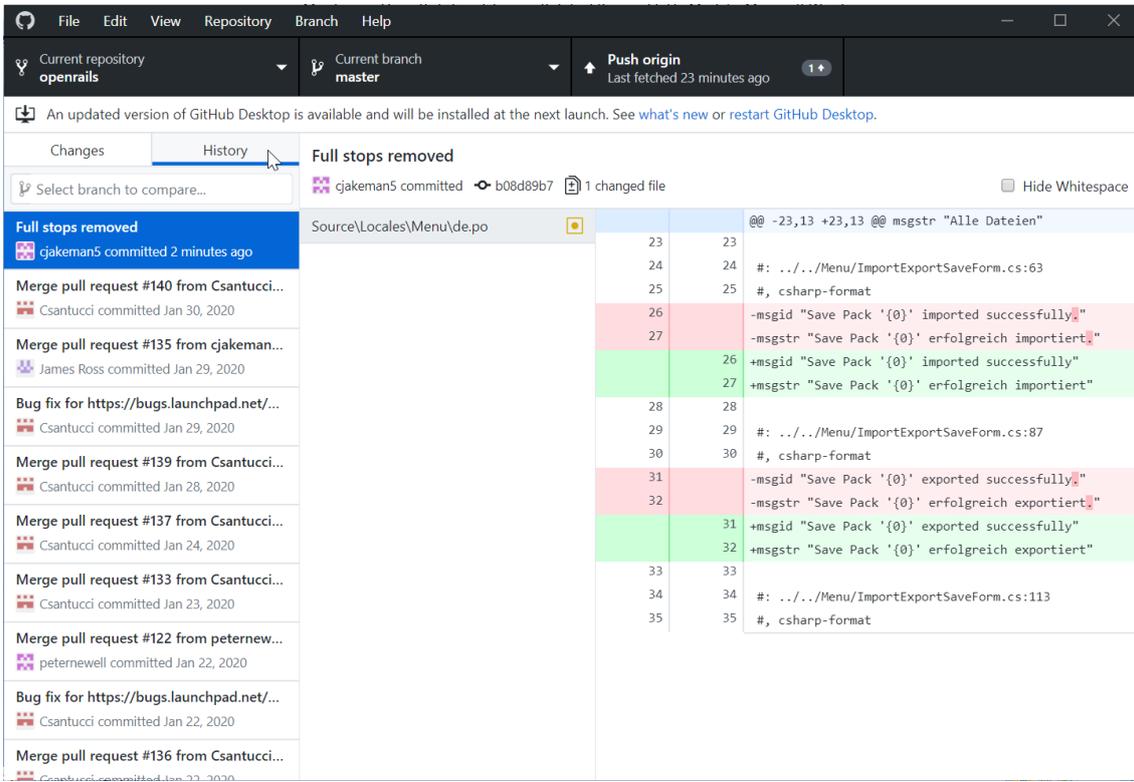
Before committing your changes, change the description from “Update de.po” to something with more meaning.

5. Commit your changes to the local repo on your PC

The “commit” operations stores your changes securely and locally. You can make more changes and more commits if you need to.



By the way, switching from the *Changes* tab to the *History* tab shows all the commits you have made.



6. Push your commit with all your changes back to your remote repo at GitHub

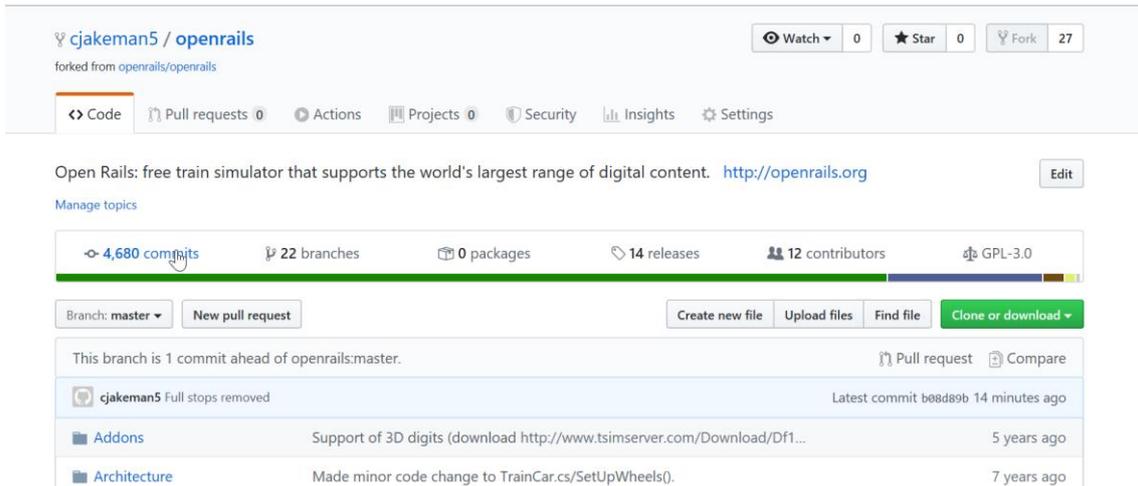
Once all your changes are ready and committed to the local repo, you must “push” them up to your remote repo (which has the default name “origin”).

From the *Changes* tab, press the “Push origin” button to transfer the commits to GitHub.com.

7. Submit your commit by making a “pull request” to the official repo at GitHub

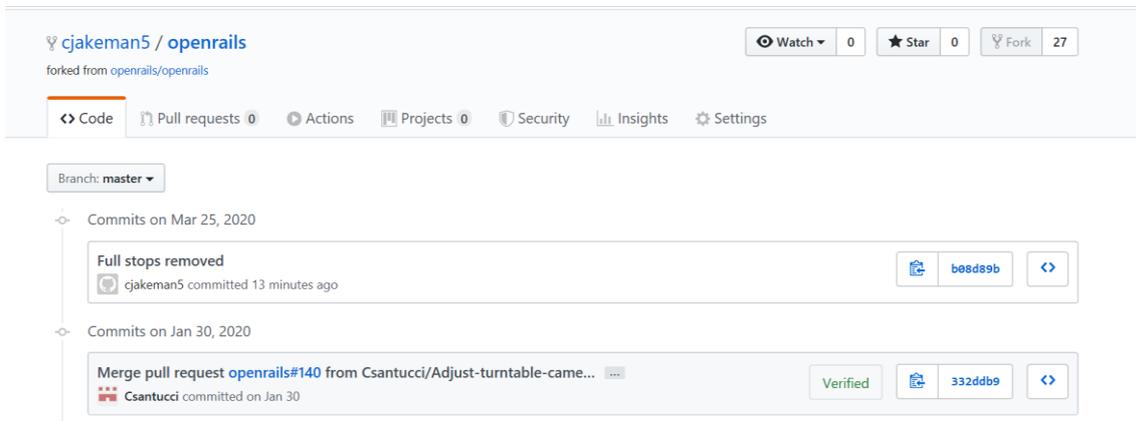
Return to your remote repo at GitHub.com.

You will see that your remote repo is 1 commit ahead of openrails:master (the “master” branch at the official repo).

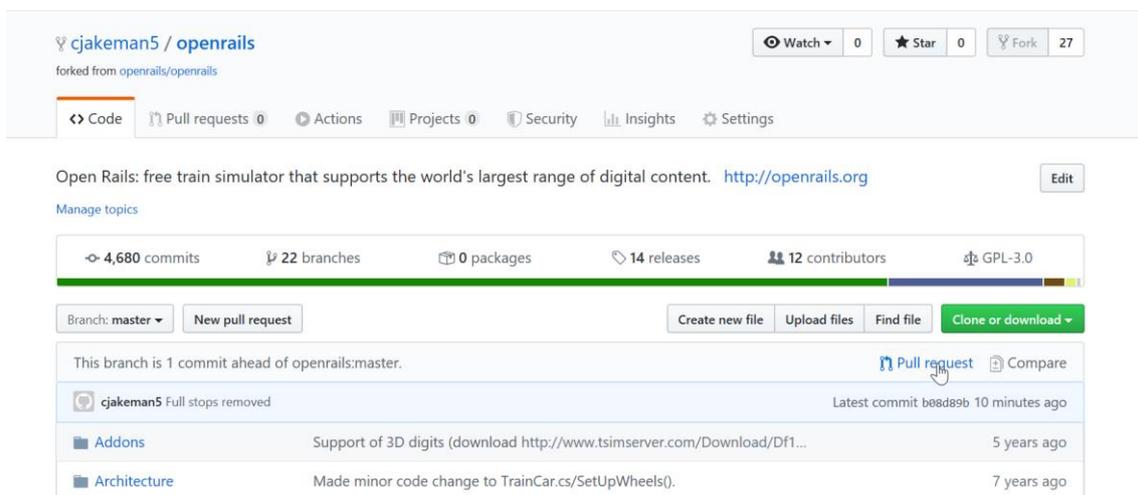


The screenshot shows the GitHub interface for a forked repository. At the top, it displays the repository name 'cjakeman5 / openrails' and indicates it was forked from 'openrails/openrails'. There are buttons for 'Watch', 'Star', and 'Fork'. Below this is a navigation bar with 'Code', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The repository description is 'Open Rails: free train simulator that supports the world's largest range of digital content.' with a link to 'http://openrails.org'. A 'Manage topics' link is also present. A statistics bar shows '4,680 commits', '22 branches', '0 packages', '14 releases', '12 contributors', and 'GPL-3.0'. Below this is a 'Branch: master' dropdown and a 'New pull request' button. A 'Clone or download' button is also visible. The main content area shows a message: 'This branch is 1 commit ahead of openrails:master.' and a list of commits. The latest commit is by 'cjakeman5' with the message 'Full stops removed' and a commit hash 'b08d89b' from 14 minutes ago. Two other commits are listed: 'Addons' (Support of 3D digits) from 5 years ago and 'Architecture' (Made minor code change) from 7 years ago.

If you press the blue commits link, then you get to see that your new commit has arrived:



Then go back to the previous page and press “Pull request” to submit your change to the official repo.



GitHub will show you your files and check that a merge with the official repo is possible without conflicts. When translating, there is unlikely to be a conflict unless two translators are working on the same language at the same time. In that case, it would be better to share the same remote repo.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: openrails/openrails base: master head repository: cjakeman5/openrails compare: master

✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Mar 25, 2020

cjakeman5 Full stops removed b08d89b

Showing 1 changed file with 4 additions and 4 deletions. Unified Split

If you want to provide an explanation of your work to help the reviewer, then use the form to “Leave a comment”.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: openrails/openrails base: master head repository: cjakeman5/openrails compare: master

✓ Able to merge. These branches can be automatically merged.

Full stops removed

Write Preview AA B i “ <> 🔗 ☰ ☷ ☰ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Allow edits from maintainers. [Learn more](#) Create pull request

Finally, press the green button:

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: `openrails/openrails` base: `master` head repository: `cjakeman5/openrails` compare: `master`

✓ Able to merge. These branches can be automatically merged.

Full stops removed

Write Preview AA B i “ <> @

Full stops removed to conform to new style guide.

Attach files by dragging & dropping, selecting or pasting them.

Allow edits from maintainers. [Learn more](#)

Create pull request

Your new Pull Request will be added to the top of the list at the official repo:

openrails / openrails Watch 11 Star 24 Fork 27

Code Pull requests 17 Actions Security Insights

Filters is:pr is:open Labels 12 Milestones 0 New pull request

17 Open 130 Closed Author Label Milestones Reviews Assignee Sort

🐛 Bug fix for <https://bugs.launchpad.net/or/+bug/1868278> AI train authorization to pass red signal not saved bug

#146 opened 5 days ago by Csantucci • Review required

Job done !