

**From:** Alexandre Göttel

**Subject:** Guide to personal web updates with git

**Date:** January 2021

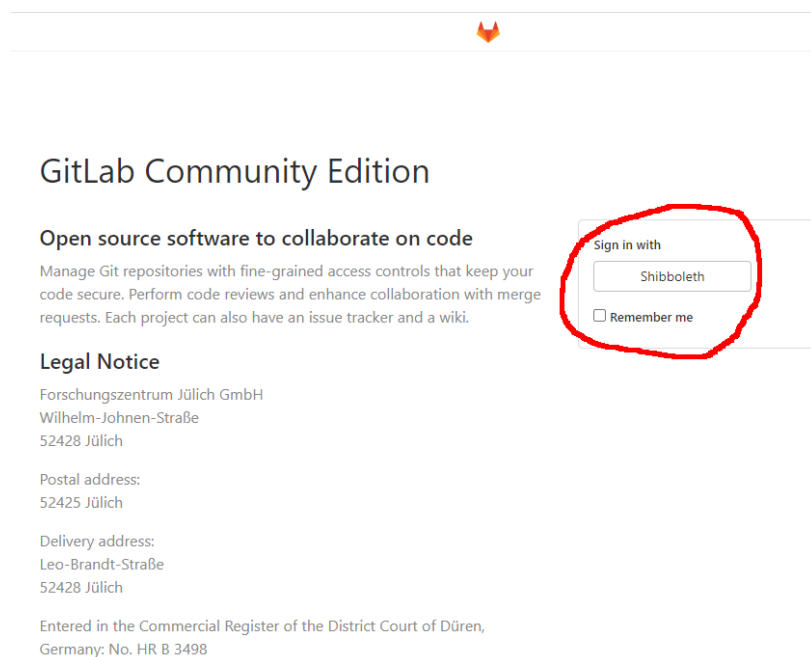
---

## 1 Introduction

This guide explains how to make personal changes to the website using the FzJ GitLab. It is separated in different sections, with first an explanation of how to setup from scratch, and then of how to actually make changes. If you already setup your GitLab account, you can skip to section 3. It also contains a quick guide for experienced users, as well as some tips and tricks.

## 2 Setting up from scratch

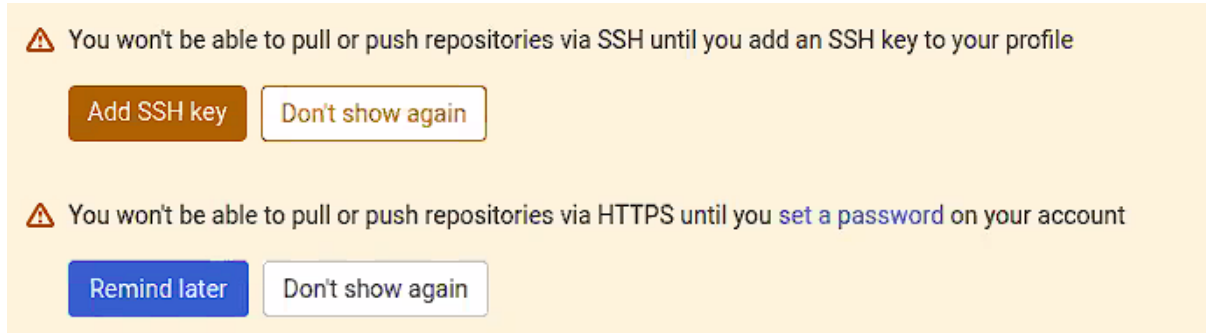
The first step in the chain is to log into the gitlab website of the Forschungszentrum. It should be accessible from any network, so there is no need for a vpn. The adress in [https://jugit.fz-juelich.de/users/sign\\_in](https://jugit.fz-juelich.de/users/sign_in), you should get to the webpage depicted in the figure below:



Click on the *Shibboleth* button on the right (red circle), then log-in using your FzJ central e-mail credentials. Afterwards, you should see a list of your projects. Look for „IKP Website“ and click on it. If you cannot find it, write an e-mail to [l.pelicci@fz-juelich.de](mailto:l.pelicci@fz-juelich.de) to be added to the

project.

Once you are on the project page, if you have not used this website before, you should see warnings at the top of the page, as shown in the figure below:



Before continuing, at least one of those warnings must be resolved. While you can add a Secure Shell for ssh access, it is also sufficient to use a password. To set a password on your account (This is the password you will use to pull/push from the git server, NOT your central e-mail password!), click on your profile picture on the top right, then on „Settings“. Near the center left of the page, you should now see „Password“. Click on it and follow the instructions to set a password for your account.

You are now ready to setup the web-page folder on your computer. Using a terminal, navigate to the folder in which you would like to put the website folder, then type:

```
$ https://jugit.fz-juelich.de/group_ikp_website/ikp-website.git
```

You are now setup! See the following section to learn how to update the website

### 3 Updating your branch

The following section describes in detail what to do when you wish to update the website. If you are already familiar with the steps, you can go directly to section 3.2 to find a short list of steps to take, without the explanation and details. Section 3.4 contains a few example screenshots to help.

#### 3.1 Full guide

Open a terminal and navigate to the git website folder. The first step is to go to the master branch:

```
$ git checkout master
```

During a normal workflow, the administrator deletes the (online) branches that have been merged into master. So, even if you already have a branch / have made changes in the past, you will need to create a new one. One can think of it this way: a branch is a temporary object created to implement a specific change, each time you want to change something new, you create a new branch. However, if the administrator has not processed your past request yet, it could happen that your branch still exists remotely. This is easy to check with the following commands (see Fig. 1):

```
$ git pull
$ git fetch --prune
$ git branch
$ git branch -r
```

The first two commands will update your local repository to be fully up-to-date, making sure that no ambiguous branches exist. The *git branch* command will list all the local branches (on your machine), and *git branch -r* will list all of the remote branches.

At this point, you might find that your old branch still exists locally, but not remotely. Delete the outdated **local** branch by using:

```
$ git branch -D BranchName
```

If you really know what you are doing, you can also switch to your existing branch to continue updating it. However the simpler option is simply delete it and create a new branch. To make things easier for your administrator, include your name in the branch name.

This is how you create the new branch:

```
$ git checkout -b BranchName
```

where „BranchName“ is the name of your branch. You are now ready to make changes to any files in the repository. Once you are done editing the .html, remember to stage the edited files, then to commit and push the branch (See Fig. 3):

```
$ git add group_mem/myFile.html
$ git commit -m "Fixing typo in myFile.html"
$ git push --set-upstream origin BranchName
```

The *add* command stages your changes. This allows you to group different changes together if necessary. In theory, you can even stage different parts of a single file in different commits! Here however, that will probably not be necessary. *commit* then takes your current stage, labels it, and makes it a part of your local repository history. The *-m* option lets you write a comment for this commit. The comment should briefly and broadly summarise that the commit does, for example “Updating myFile.html”. Finally, *push* uploads your branch to the server. If you are working on a branch which already exists online, *git push* is enough. However, since you are probably working on a new local branch which doesn’t yet exist online, you need to tell git to push your local branch to a new online branch with the same name, which is done by using the *-set-upstream* flag.

Congratulations, you are now done and can expect to shortly see your changes on the web-page. Remember to set the monday.com sub-item status to „Done“ (on the *Web Page* board) to let me know you’re ready. If you encountered any problems along the way, or have any feedback, contact your administrator to find a solution. You can also set your monday.com sub-item status to „Stuck“, and I will get a notification.

## 3.2 Quick guide

Open a terminal and navigate to the git website folder, then:

```
$ git checkout master
$ git pull
$ git fetch --prune
```

At this point, you might want to check on the status of your (possibly existing) branches:

```
$ git branch
$ git branch -r
```

If you find an old local branch that was deleted remotely, simply delete it with.

```
$ git branch -D BranchName
```

Then choose a name for a new branch, and proceed:

```
$ git checkout -b BranchName
```

This is where you edit .html files using your favourite text/code editor. Once you are done, stage and push all the files you modified:

```
$ git add group_mem/myFile.html
$ git commit -m "Fixing typo in myFile.html"
$ git push --set-upstream origin BranchName
```

Remember to update your monday.com sub-item status. All done!

## 3.3 Tips and tricks

For those who are unfamiliar with git, here are a few commands to help you along:

### 3.3.1 git status

You can use this command to get some information about the current state of your local repository. It will display the current branch you are on, as well as show which files currently differ from the previous commit. It will also show you which files are already staged.

### 3.3.2 git branch -d branchName

If you accidentally create a local branch which you don't want (e.g. typo), use this command to delete the local branch. However, it will not delete the corresponding remote branch if one exists!

### 3.3.3 git reset --hard

If something goes wrong and the changes you made in the code are very buggy and non-recoverable, use this command to reset the code to the status of the previous commit. Very useful to play around, but be careful or you might lose some work! This is irreversible!

### 3.4 Figures

```
D:\Work\ikp-website>git pull
Already up to date.

D:\Work\ikp-website>git fetch --prune

D:\Work\ikp-website>git branch
* master
  test_branch

D:\Work\ikp-website>git branch -r
  origin/HEAD -> origin/master
  origin/master
  origin/test_branch

D:\Work\ikp-website>
```

Figure 1: What updating a clean repository looks like. In this example, *git pull* outputs: „Already up to date.“, and *git fetch --prune* outputs nothing, since the local repository was already in sync with the remote. So don't worry if nothing is displayed! Afterwards you can see an example of the *git branch* commands. The first command lists the local branches, you can see that this repository is currently in **master**, but that **test\_branch** also exists. The last command lists the remote branches, as one can see, **master** and **test\_branch** also exist remotely. In this case, to update the webpage, you would probably want to create a new branch (not called test.branch!).

```

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (group/Cornelius)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (master)
$ git pull
remote: Enumerating objects: 192, done.
remote: Counting objects: 100% (192/192), done.
remote: Compressing objects: 100% (138/138), done.
remote: Total 140 (delta 53), reused 1 (delta 1), pack-reused 0
Receiving objects: 100% (140/140), 20.23 MiB | 5.16 MiB/s, done.
Resolving deltas: 100% (53/53), completed with 51 local objects.
From https://jugit.fz-juelich.de/a.goettel/ikp-website
 8d82b99..dc07f07  master    -> origin/master
* [new branch]    group/Alex    -> origin/group/Alex
* [new branch]    group/Giulio   -> origin/group/Giulio
* [new branch]    group/apeksha  -> origin/group/apeksha
* [new branch]    group/livia    -> origin/group/livia
* [new branch]    group/mariam   -> origin/group/mariam
* [new branch]    group/ricky    -> origin/group/ricky
Updating 8d82b99..dc07f07
Fast-forward
 activities_2016.html | 3 +
 activities_2017.html | 3 +
 activities_2018.html | 9 +-
 activities_2019.html | 6 +-
 activities_2020.html | 57 +++++
 activities_2021.html | 152 ++++++
 ann_reports.html     | 1 +
 awards.html          | 2 +-
 events.html          | 5 +-
 exp_Borexino.html    | 2 +-
 exp_JUNO.html        | 2 +-
 exp_OSIRIS.html      | 2 +-
 experiments.html     | 2 +-
 group_mem.html       | 2 +-
 group_mem/group_mem_Alexandre.html | 2 +-
 group_mem/group_mem_apeksha.html    | 2 +-
 group_mem/group_mem_christoph.html   | 2 +-
 group_mem/group_mem_cornelius.html   | 2 +-
 group_mem/group_mem_dong.html         | 2 +-
 group_mem/group_mem_giulio.html       | 2 +-
 group_mem/group_mem_lars.html          | 2 +-
 group_mem/group_mem_livia.html         | 2 +-
 group_mem/group_mem_luca.html          | 2 +-
 group_mem/group_mem_lukas.html         | 2 +-
 group_mem/group_mem_mariam.html        | 2 +-
 group_mem/group_mem_mariia.html        | 2 +-
 group_mem/group_mem_oemer.html         | 2 +-
 group_mem/group_mem_philipp.html       | 2 +-
 group_mem/group_mem_ricky.html          | 2 +-
 group_mem/group_mem_rikhav.html        | 2 +-
 group_mem/group_mem_sale.html           | 2 +-
 group_mem/group_mem_sindhujha.html     | 2 +-
 group_mem/group_mem_yaping.html         | 2 +-
 group_mem/group_mem_yu.html             | 2 +-
 group_mem/group_mem_yuhang.html         | 2 +-
 group_mem/group_mem_zara.html           | 2 +-
 images/.directory          | 4 +
 .../2021-Jan-GroupPhoto_EuropeanJUNOSWorkshop.jpg | Bin 0 -> 405113 bytes
 images/Events/Christoph_prize.jpeg | Bin 0 -> 89074 bytes
 images/Events/Christoph_thesis.jpeg | Bin 0 -> 57345 bytes
 images/Events/Cultural.jpg | Bin 0 -> 555671 bytes
 images/Events/Cultural_oemer.jpg | Bin 0 -> 173698 bytes
 images/Events/DFG_juelich.jpg | Bin 0 -> 1110015 bytes
 images/Events/DSC_0247.JPG | Bin 0 -> 280857 bytes
 images/Events/FerraraAlsdorf.jpg | Bin 0 -> 488136 bytes
 images/Events/Mariia_thesis.jpeg | Bin 0 -> 78194 bytes
 images/Events/Neutrino_VR.jpg | Bin 0 -> 104687 bytes
 images/Events/Sindhu_taup2019_prize.jpeg | Bin 0 -> 59350 bytes
 images/Events/TagDerNeugier.jpeg | Bin 0 -> 462639 bytes
 images/Events/Thiery_Aachen.jpg | Bin 0 -> 762976 bytes
 images/Events/Winter_Seminar_2018.jpg | Bin 0 -> 188310 bytes
 images/Events/christmas2020.jpg | Bin 0 -> 588843 bytes
 images/Events/gnaikp2.jpg | Bin 0 -> 267495 bytes
 images/Events/gnaikp2018_group_meeting.jpg | Bin 0 -> 78078 bytes

```

Figure 2: A typical example *git pull* command. You can see the list of all changes that were made since the last time you used the *pull* command!

```

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (master)
$ git checkout -b group/cornelius
Switched to a new branch 'group/cornelius'

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (group/cornelius)
$ git add group_mem/group_mem_cornelius.html

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (group/cornelius)
$ git commit
[group/cornelius 65f66bc] Added phone numbers for Jülich and Aachen
1 file changed, 2 insertions(+), 1 deletion(-)

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (group/cornelius)
$ git push
fatal: The current branch group/cornelius has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin group/cornelius

Cornelius@DESKTOP-1K6MFOG MINGW64 ~/sciebo/Uni/Promotion/Website_Juelich/ikp-website (group/cornelius)
$ git push --set-upstream origin group/cornelius
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 24 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 470 bytes | 470.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for group/cornelius, visit:
remote:   https://jugit.fz-juelich.de/a.goettel/ikp-website/-/merge_requests/new?merge_request%5Bsource_branch%5D=group%2Fcornelius
remote:
To https://jugit.fz-juelich.de/a.goettel/ikp-website.git
 * [new branch]      group/cornelius -> group/cornelius
Branch 'group/cornelius' set up to track remote branch 'group/cornelius' from 'origin'.

```

Figure 3: A typical example for your web update. The first command creates a new branch. **Attention:** in between the first and second command (checkout and add), the changes are made to the files using an external text/code editor. Then, the stage is committed and the user attempts to push. This fails since the corresponding remote branch does not exist yet. Finally, the last command pushes the local branch using the `--set-upstream` flag to create the corresponding remote branch. (Thanks to Cornelius for testing).