

HowTo use a WPIA document repository



2018-11-16

---

# Contents

<b>1</b>	<b>HowTo use a WPIA document repository</b>	<b>1</b>
1.1	Create fork . . . . .	1
1.2	Edit File And Commit Change . . . . .	2
1.3	Create Merge Request . . . . .	3
1.4	Share Repository With Others . . . . .	4
1.5	Style Guide . . . . .	5
1.5.1	Headings . . . . .	5
1.5.2	Links . . . . .	5
1.5.3	Pictures . . . . .	5
1.5.4	Lists . . . . .	5
1.5.5	Tables . . . . .	5
1.5.6	Comments . . . . .	6
1.5.7	Highlight Code . . . . .	6
1.5.8	Highlight Text . . . . .	6
1.6	Output Via Pandoc . . . . .	6

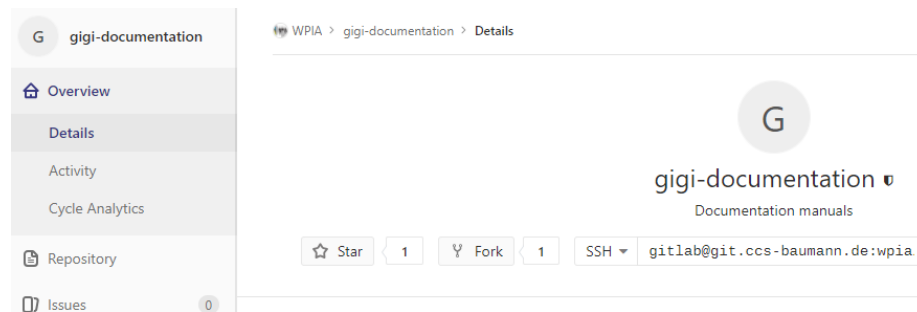
# 1 HowTo use a WPIA document repository

Currently WPIA uses 2 document repositories placed on a non-public GitLab repository server. \* wpia/gigi-documentation \* wpia/policies

You should work not in the master branch and the original repository. To start you should fork the repository.

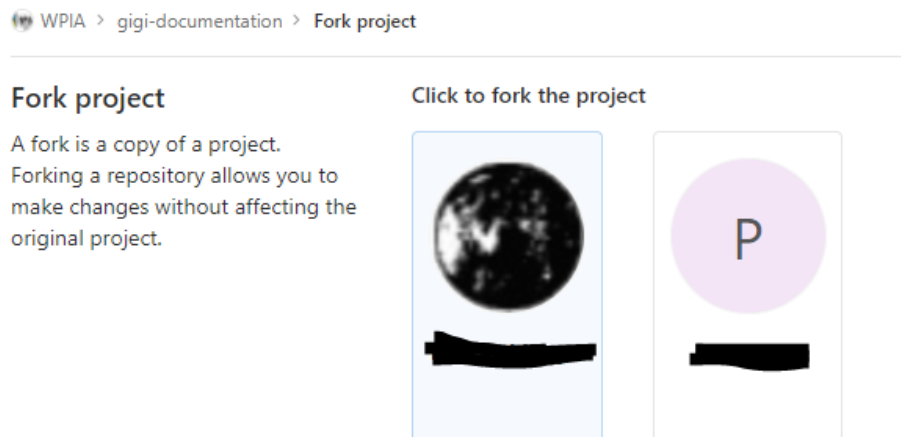
## 1.1 Create fork

Open the desired repository and navigate in the left navigation area to Overview - Details

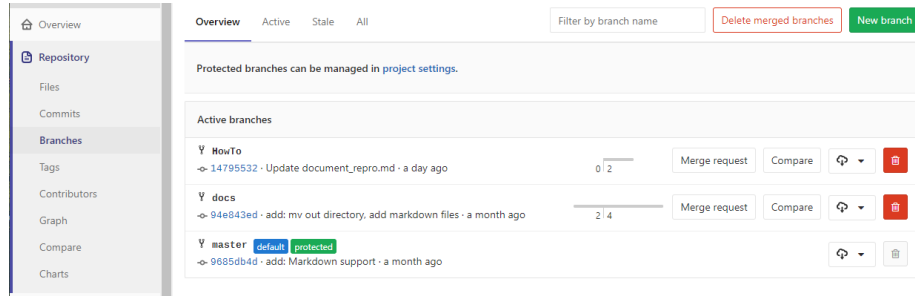


Use the fork button to create a fork of the repository.

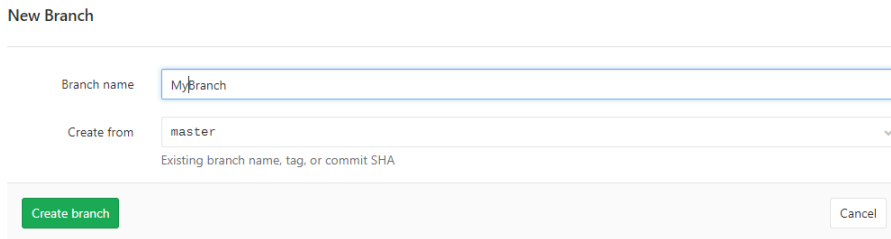
In the next step choose where the new fork should be placed.



## Add new branch In the navigation bar choose Repository - Branches



Use the button on the right “New Branch”



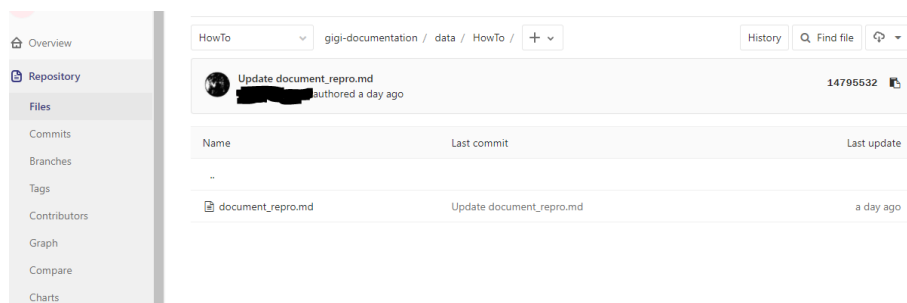
Add a branch name, choose anything that is specific for you. No spaces are allowed.

Normally you should create the new branch from master.

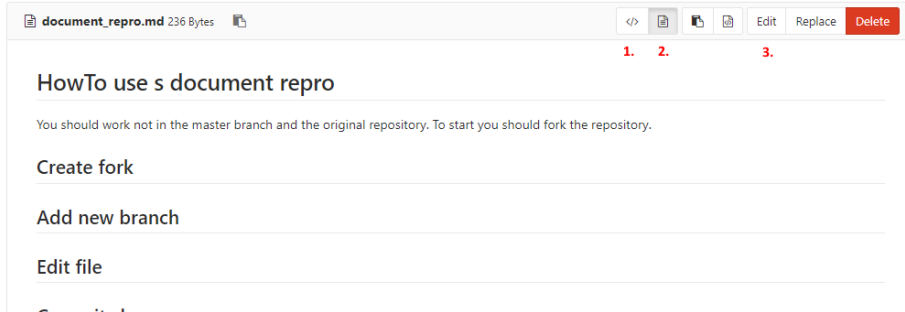
Finish the creation with the button Create branch

## 1.2 Edit File And Commit Change

To edit a file navigate to the file first with the navigation bar Repository - Files. Then select the branch and by choosing the appropriate folders.



Click on to the file to open the preview of the file.

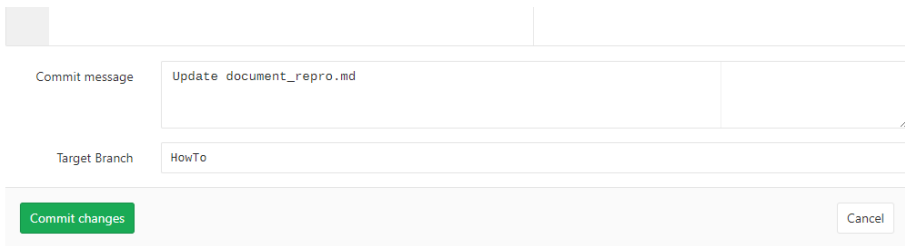


Use the button to change the view between preview (2.) and source view (1.) or to edit the file (3.)



While editing the text choose between editing the text (Write button) or preview of your work (Preview button) on the top left.

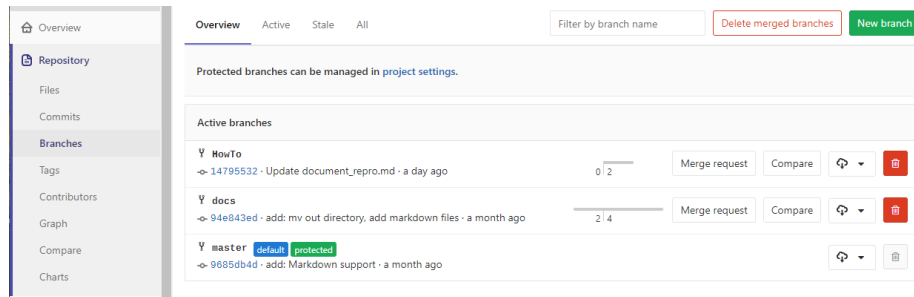
The preview is useful especially if you work on format and layout of the text.



Once you finished your changes add a commit message at the bottom of the screen. Make sure you work in the correct branch and finish with Commit changes.

### 1.3 Create Merge Request

Once you finished all your changes on different files you create a merge request to show others that you have something new to add to the repository.



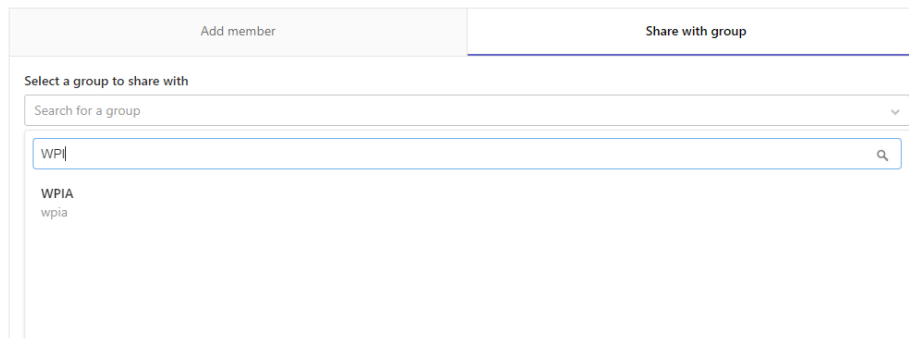
Navigate to Repository - Branches and use the button Merge request to create your merge request.

## 1.4 Share Repository With Others

To share the repository with others navigate to Settings - members in the navigation bar.

### Project members

You can add a new member to **gigi-documentation** or share it with another group.

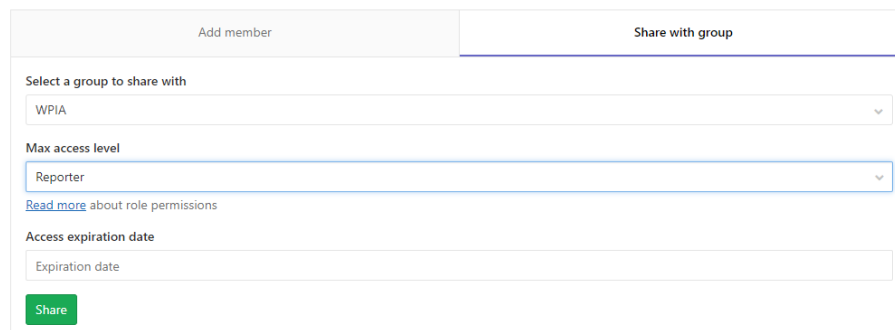


### Existing members and groups

Switch to Share with group and select the group WPIA.

### Project members

You can add a new member to **gigi-documentation** or share it with another group.



Choose the role permission. Suggestion is reporter and use “Add to project” to apply the new permission to the repository.

## 1.5 Style Guide

### 1.5.1 Headings

Headings are marked with a # for each level

### 1.5.2 Links

A link can be added as inline link with `[inline link](/url)`. No space in between the closing and opening brackets.

A link can used as reference link

Here is `[my link] [FOO]`

`[Foo]: /bar/baz`

### 1.5.3 Pictures

A link immediately preceded by a ! will be treated as an image. The link text will be used as the image' alt text.

Place the pictures in a subfolder Pictures and link the picture file with `\![image](Pictures/foo.jpg){#id .class width=30 height=20px}\`

An image with nonempty alt text, occurring by itself in a paragraph, will be rendered as a figure with a caption. The image's alt text will be used as the caption. If you just want a regular inline image, just make sure it is not the only thing in the paragraph. One way to do this is to insert a nonbreaking space after the image.

### 1.5.4 Lists

To indent use 4 spaces for each level.

**1.5.4.1 Unordered List** Use asterisks (\*), plus (+) or minus(-) for an unordered list

**1.5.4.2 Ordered List** Use the numbering at the beginning of the line (1., 2., ...)

### 1.5.5 Tables

If a table is needed the following structure is used

```
| Heading    | Heading    |
|-----|-----|
| some text | more text  |
```

### 1.5.6 Comments

Comments are added with `<!-- here the comment -->` followed by a blank line.

### 1.5.7 Highlight Code

To highlight code in line use a single backtick ``` at beginn and end. ``here some code``

To highlight a code block use triple backsticks ````` at beginn and end.

```
```  
some  
code  
```
```

### 1.5.8 Highlight Text

Italics are added with single *astericks* oder *underscores*.

Italics are added with single `*astericks*` oder `_underscores_`.

Bold is added with double **astericks** oder **underscores**.

Bold is added with double `**astericks**` oder `__underscores__`

Strikethrough is added with double tilde ~~Strikethrough~~.

Strikethrough is added with double tilde `~~Strikethrough~~`

More syntax features are available at <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet> and <https://pandoc.org/MANUAL.html>

## 1.6 Output Via Pandoc

All markdown files can be used to create a html or pdf output via pandoc.

For the different repositories there are different template files available which need some information for consistent styling.

Place the following data at the beginning of the document.

```
---  
---  
title: 'Title of document'  
  
date: yyyy-mm-dd  
tags: [tag1, tag2, tag3, ...]
```



```
subject: 'subject'  
  
oid: '47934.x.x.x.x.xx'  
version: 'x'  
status: 'WIP'  
  
language: de-DE  
...  
  
\pagenumbering{gobble}  
\pagebreak  
\tableofcontents  
\pagebreak  
\pagenumbering{arabic}  
...
```

The part with oid, version and status is only necessary for controlled documents.

The language information is only needed for documents in German.