

**How to work with EBRD’s GitHub: Repositories creation**

July 2020

Document characteristics

|  |  |
| --- | --- |
| **Property** | **Value Proposition** |
| Release date | *03/07/2020* |
| Status: | *Pending approval* |
| Version: | *V1* |
| Authors: | *everis* |
| Reviewed by: | *EBRD OL, Eliza Niewiadomska* |
| Approved by: | *EBRD OL, Eliza Niewiadomska* |

Document history

|  |  |  |
| --- | --- | --- |
| **Version** | **Description** | **Date** |
| 1.0 | How to work with EBRD’s GitHub: Repositories creation | *03.07.2020* |
|  |  |  |
|  |  |  |

Table of Contents

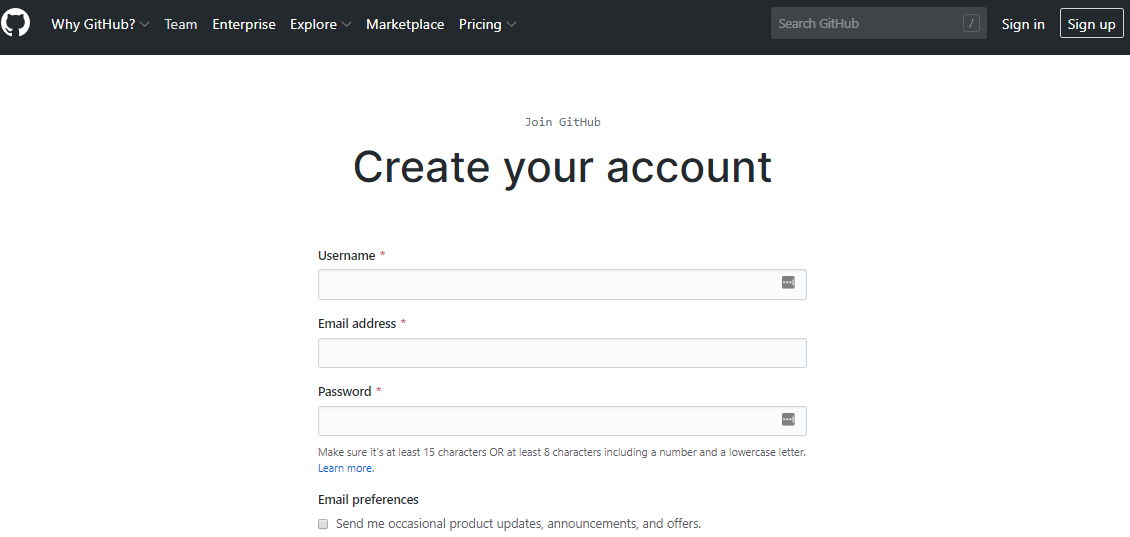
[1 Create a GitHub account 4](#_Toc44676669)

[2 Step 2 – Create a GitHub repository 5](#_Toc44676670)

[3 Step 3 – Manage content in the Code tab 7](#_Toc44676671)

[4 Step 4 – Manage content in the Wiki tab 8](#_Toc44676672)

# Create a GitHub account

First thing must be done is to create a [GitHub account](https://github.com/join?source=header-home) (if the collaborator already has one, this step can be avoided). An email account and to generate a username will be needed (username will be used in the next step).

Then, the collaborator must send an e-mail to ask the manager ([everis@ppi-ebrd-uncitral.com](mailto:everis@ppi-ebrd-uncitral.com)) to create a new Organization. The following information must be provided in that e-mail:

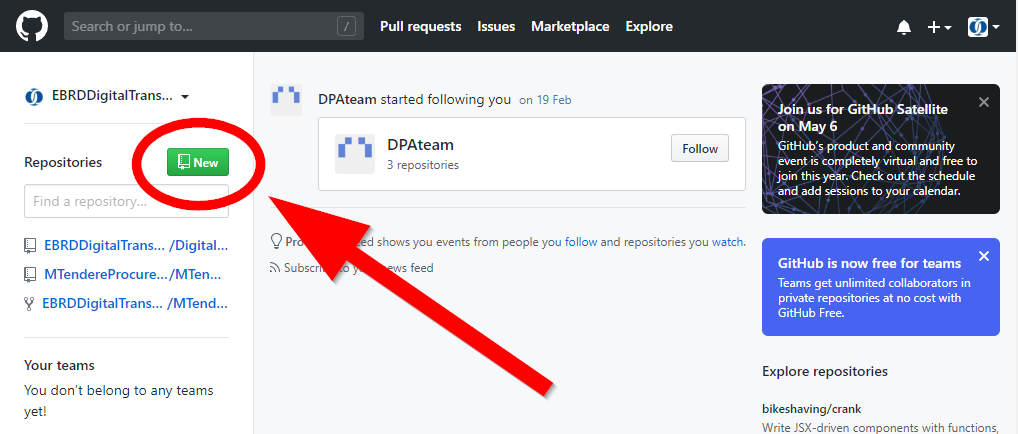
* GitHub Account Username of the collaborator
* Collaborator’s Organization name
* Project Name

The project name should follow this format: EBRD-projectname

Once the space created for that collaborator is created, the manager will use the collaborator's GitHub account in order to add it as member of the Organization. Once this is done, the collaborator will receive an e-mail in the same account he used to create the GitHub account, notifying the success on the operation.

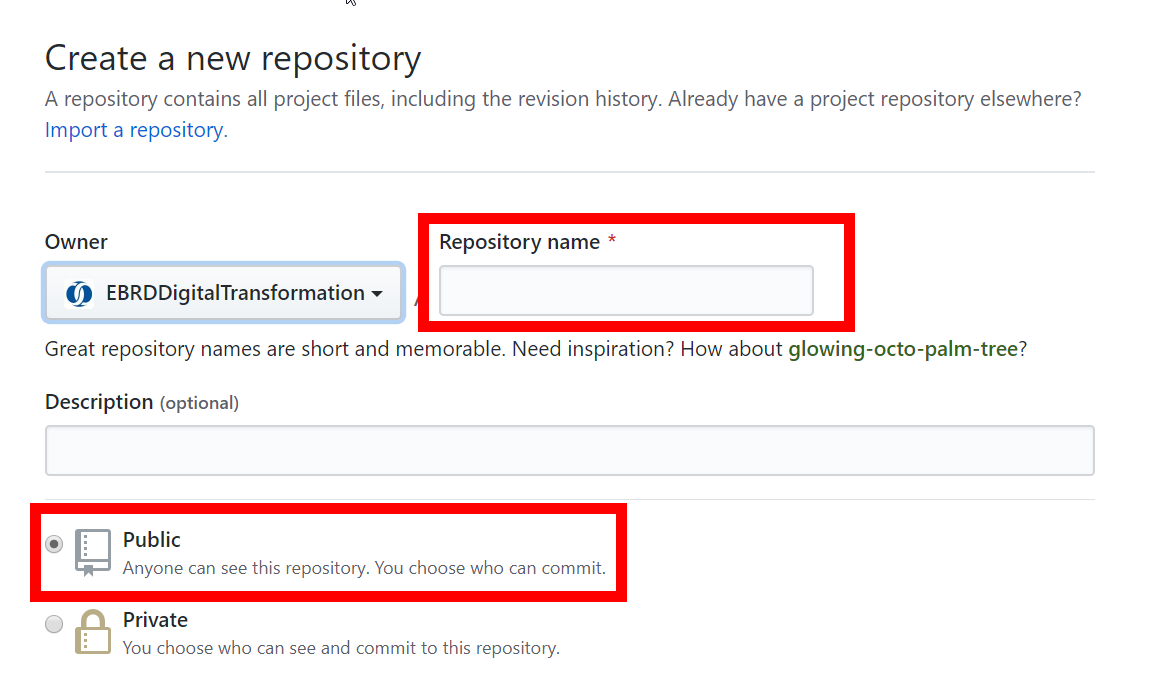
# Step 2 – Create a GitHub repository

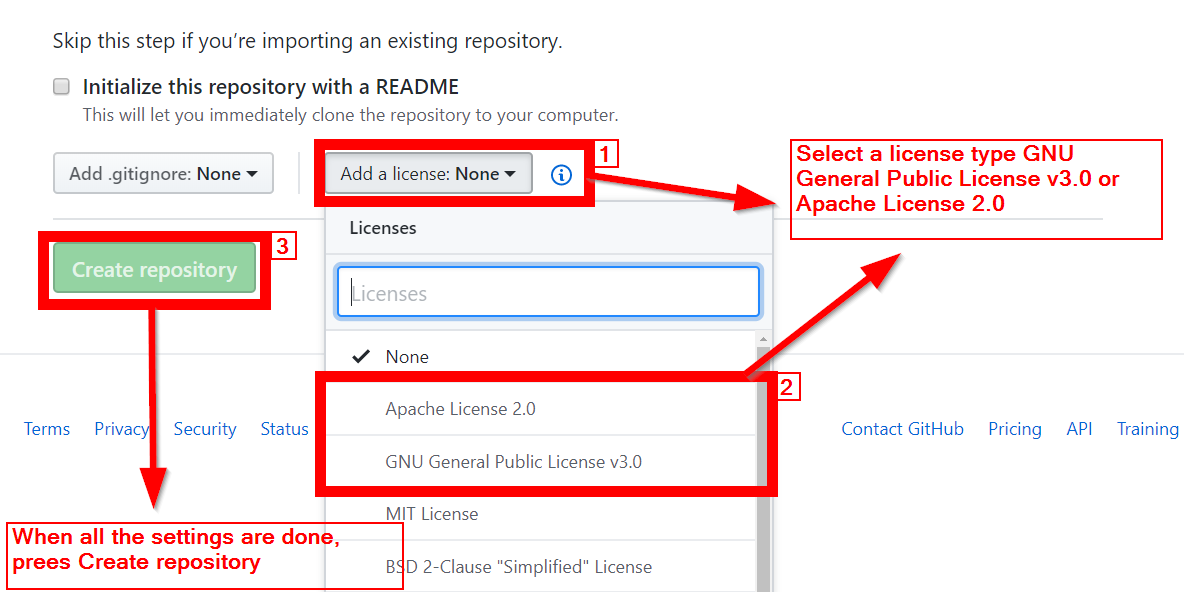
Once the collaborator is signed in, the process to [setup](https://github.com/new) the repositories can start. To create the first repository collaborator must click in the “New” button that is placed in the left panel of the [main page](https://github.com/).



In the new project window, these are the steps to be followed:

* Provide the repository name
* Setup the repository as Public so that it can be accessible for everyone
* Select the “*Initialize this repository with a README”* option
* Add a license type between GNU General Public License v3.0 and Apache License 2.0





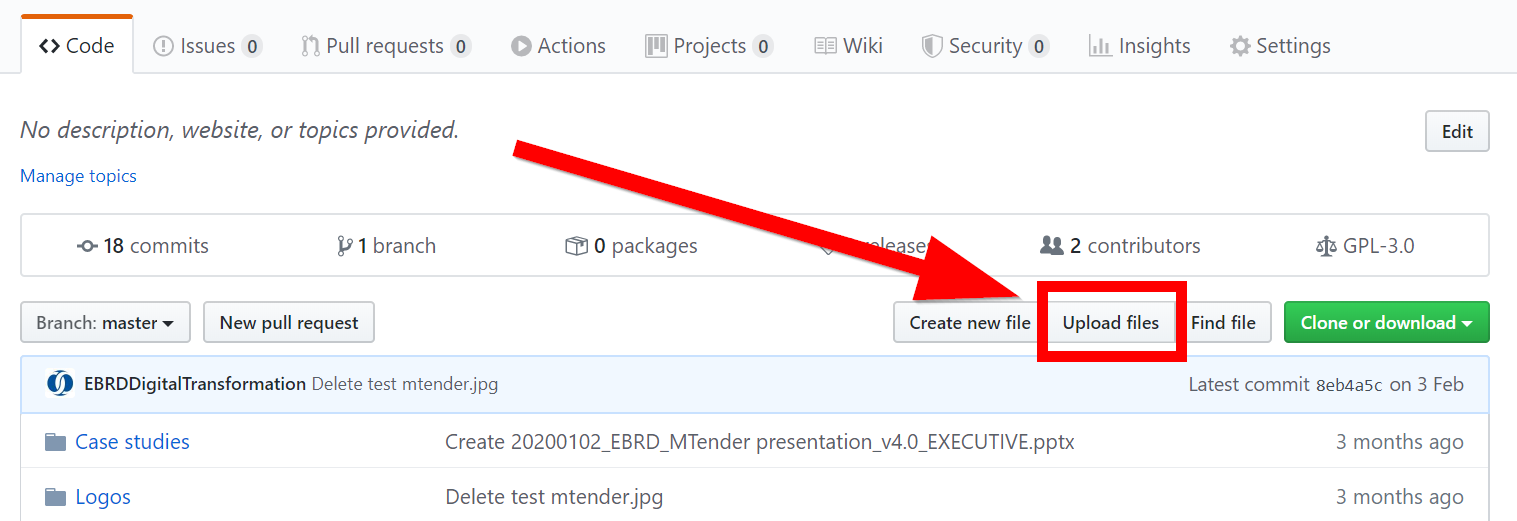
# Step 3 – Manage content in the Code tab

When the repository is created, the GitHub basic tab structure will appear. We will focus on the Code and Wiki tabs.

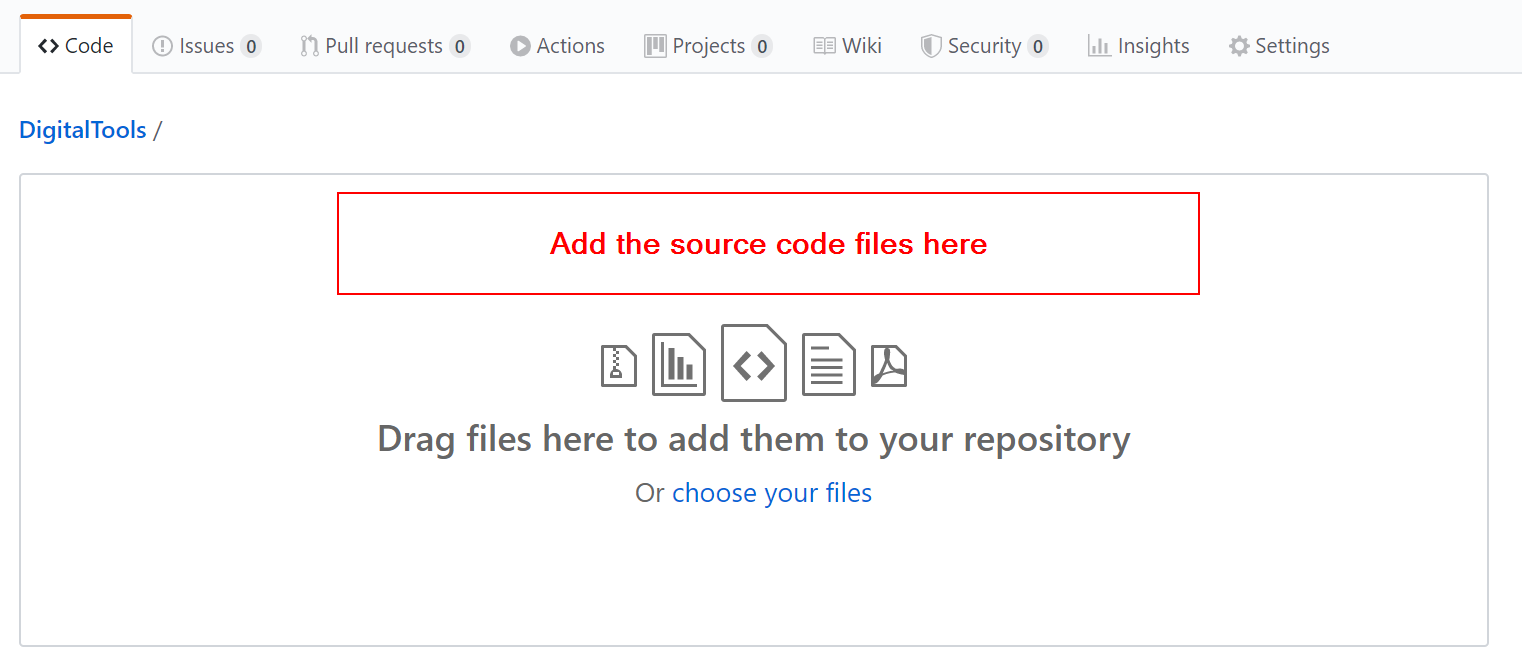


The first tab, ***Code***, will contain the following minimum contents:

1. ***README.md*** file with a brief description of the initial commit / new features of the version. It is mandatory to keep updated this file, referencing the version number.
2. **LICENSE.** This file will contain the type of license information selected while creating the repository. This does not need interaction.
3. **Source Code**. From the starting tab Code, at least one branch (master) with the source code must be uploaded.
4. **Documentation.** All the mandatory documentation uploaded in doc or pdf format in the same war that documentation is uploaded in huddle. (Functional, technical, deployment and development environment documentation)

Click on the Upload files button. 

There, the files can be dragged into the box or choose the files.



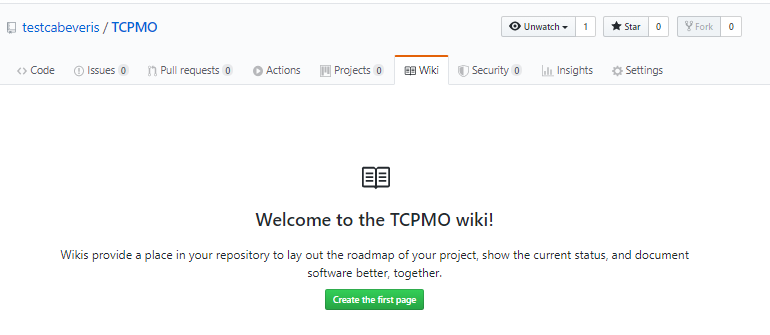
# Step 4 – Manage content in the Wiki tab

The second tab to be used in the projects, ***Wiki***, will contain diverse information and documentation related to the project. The wiki is made up of pages, which will need to be created and formatted following a certain structure:

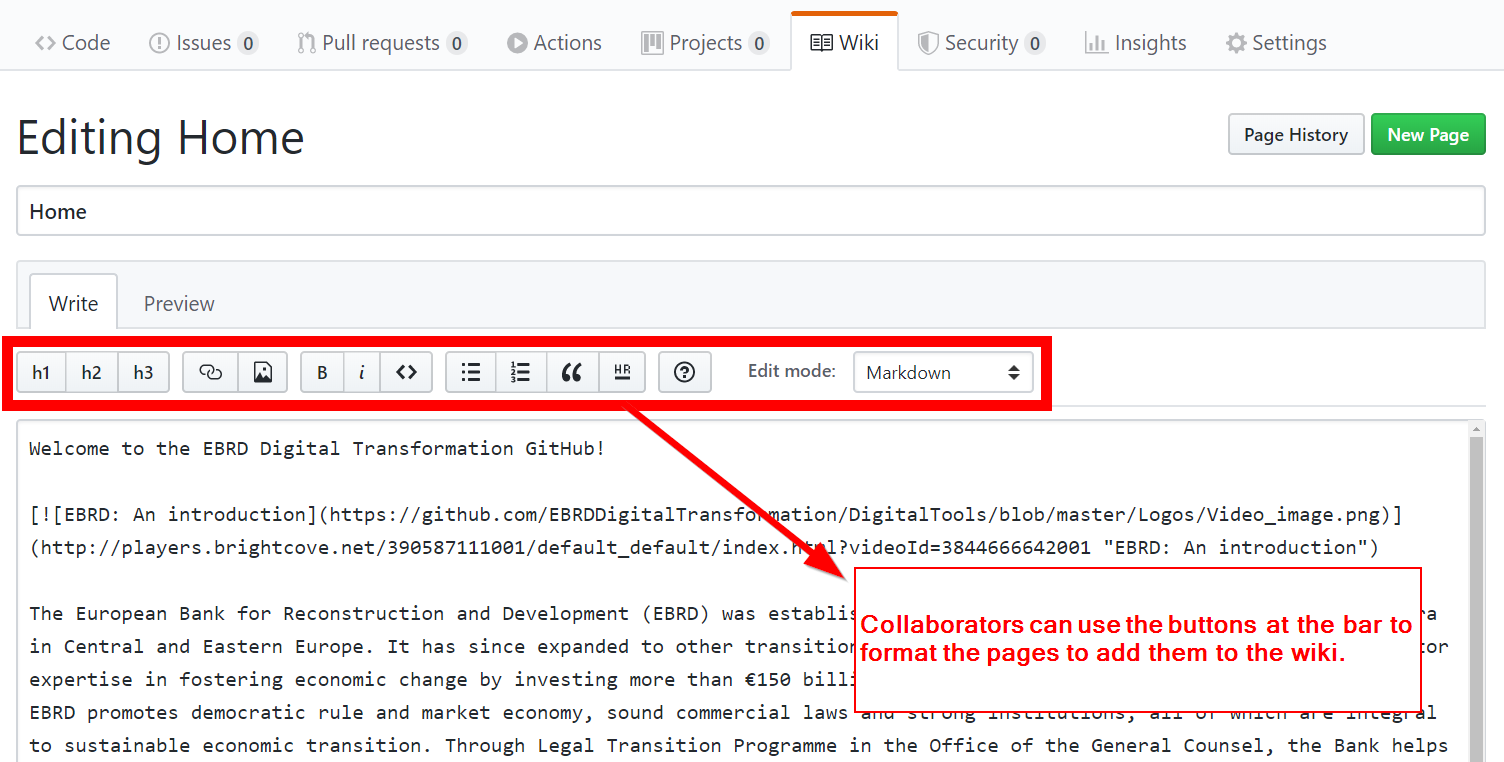
1. Front page: Summary Case Study [Template](https://github.com/EBRDDigitalTransformation/DigitalTools/wiki/Template-for-summary-case-study)
2. Detailed Case Study [Template](https://github.com/EBRDDigitalTransformation/DigitalTools/blob/master/TEMPLATES%20for%20Case%20studies/HUDDLE%20TEMPLATE_DETAILED_CASE%20STUDY_FOR%20GITHUB.docx)
3. Functional Documentation
4. Technical Documentation
5. Deployment documentation
6. Development environment documentation

All this contents will be uploaded to the repository and added to the wiki as links in the sidebar so anyone can have an easy access to those files while reading the project wiki.

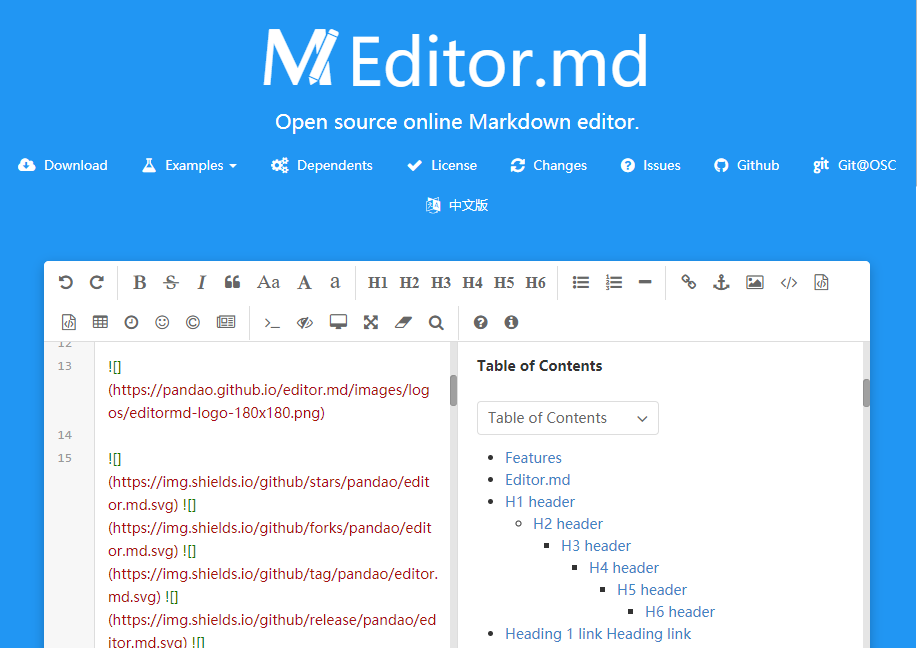
The way to create new pages in GitHub is explained below.



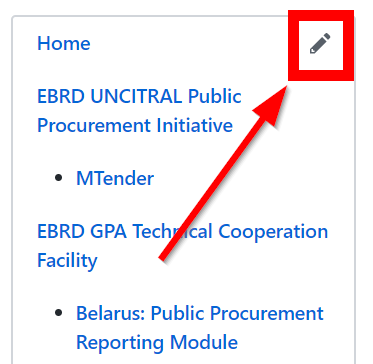
GitHub Wiki uses markdown code to include and give format to text. The text can be formatted by using the buttons on the top of the editing box.



To give format to the content code in an easier way, an Open source online Markdown editor like [this one](https://pandao.github.io/editor.md/en.html) can be used. This editor offers more options and also, shows simultaneously the markdown code and the final shape of the text.



Then, collaborator can just copy the markdown code and paste it in the wiki page.

**Adding contents to the sidebar.** To edit the sidebar and add links to the contents indicated at the head of this section, the user must click in the pencil icon on the right of the screen.

There the link can be added with the usual markdown format by clicking in the link button on the edition toolbar and typing the information in the pop-up window.

