ConsenSys-Komgo

Authors Help Guide

|  |  |
| --- | --- |
| Issued to: | Mark Ugbomah, Guillaume Dechaux, Mano Arunachalasamy – ConsenSys |
| Prepared by: | Barbara Kujawska & George Lewis – 3di |
| Issue date: | 5th December 2018 |

Table of Contents

[1 Introduction 3](#_Toc531796533)

[1.1 Markdown 3](#_Toc531796534)

[1.2 GitBook Editor 3](#_Toc531796535)

[1.3 Stackedit.io 3](#_Toc531796536)

[2 Setting up your content authoring environment 4](#_Toc531796537)

[2.1 Content source 4](#_Toc531796538)

[2.2 Setting up your GitBook 4](#_Toc531796539)

[3 How to edit the content 5](#_Toc531796540)

[3.1 How to update an article 5](#_Toc531796541)

[3.2 How to create a new article 6](#_Toc531796542)

[3.3 How to resolve a conflict 7](#_Toc531796543)

[4 How to review the content 8](#_Toc531796544)

[4.1 Connecting to the GitLab repository 8](#_Toc531796545)

[4.2 Providing feedback using Stackedit.io 8](#_Toc531796546)

[5 How to publish the content 9](#_Toc531796547)

[6 Managing the style and look of the Help Portal 10](#_Toc531796548)

[7 Resources 10](#_Toc531796549)

# Introduction

This Authors Help Guide is intended to provide basic instructions on how to manage and edit the content of the Help Portal for the ConsenSys-Komgo platform.

This guide covers content creation, approval, management, and publication.

The source content is created using Markdown. We use GitLab to manage the content source, GitBook Editor (legacy) to edit the content, and suggest Stakedit.io to provide review feedback to draft content.

GitLab enables collaboration between multiple authors on the same set of files. The version control logic is easy to follow and enables authors to maintain accurate and always up-to-date content.

## Markdown

Markdown is a lightweight markup language that you can use to add formatting elements to plain text documents. You can use any text editor to edit Markdown but we recommend GitBook Editor (legacy) as an offline and easy-to-use tool to manage TOC structure.

For more advanced usage of the GitLab, GitBook or Markdown, check Resources at the end of the guide.

## GitBook Editor

GitBook is a simple desktop application that allows you to create structured documents using Markdown.

This is done by creating individual topics, and then organising them using a table of contents.

You can download the editor from the following page: <https://legacy.gitbook.com/editor>

## Stackedit.io

Stackedit.io is a web-based Markdown editor that provides the capability to comment on content written in Markdown.

This tool makes it convenient for content to be reviewed prior to publication. All changes made in Stackedit.io are saved to the GitLab repository.

However, comments are only visible when files are viewed in Stackedit.io.

You can access Stackedit.io at the following link: [https://stackedit.io/app#](https://stackedit.io/app)

# Setting up your content authoring environment

## Content source

The source content is stored in the following GitLab repository:

* GitLab repository: [Add URL]

If you don’t have a GitLab account yet, request one. You will need to have a GIT client installed to be able to synchronise a copy of the files on your PC with the repository.

## Setting up your GitBook

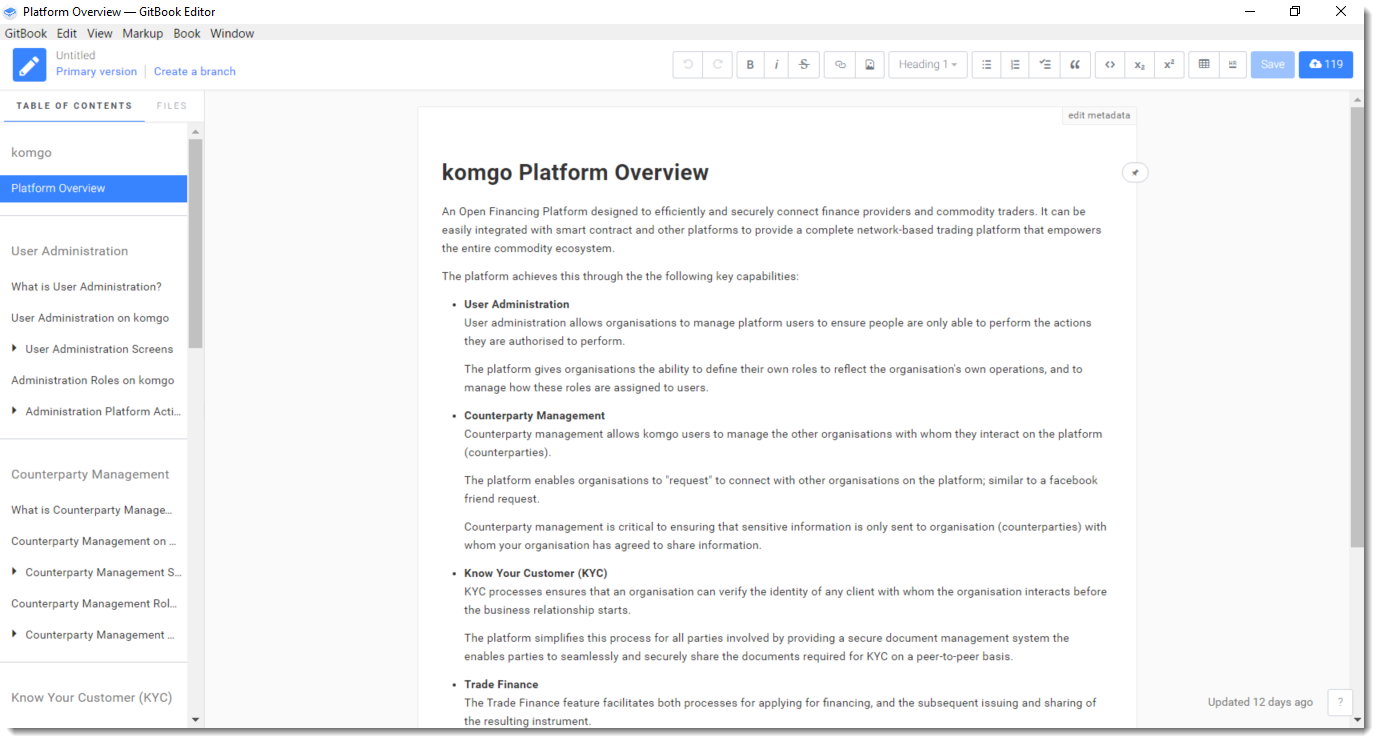
Before editing the content, you need to download and install GitBook.

1. Download and launch GitBook from the following page:   
   <https://legacy.gitbook.com/editor>
2. Open GitBook.

You don’t have to log in to GitBook to use it.

1. From the GitBook’s top menu, select ***GitBook Editor*** > ***Open*** and navigate to your local repository folder.
2. Click ***Select folder***.

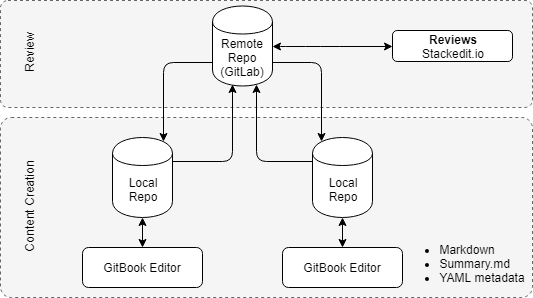
The repository files load into GitBook.



1. From the GitBook’s top menu, select ***GitBook*** > ***Preferences***.
2. In the ***Editor Settings*** window, select the ***Git*** tab.
3. Click the ***Automatically generate commit message from changes*** toggle switch to turn it off.
4. Click the ***Close*** button.

# How to edit the content

GitLab version control system enables collaboration between multiple authors on the same set of files. Each of the authors can make changes to their local repository and commit them to a remote repository where they are then available for other authors.



## How to update an article

In order to avoid conflicts, that is differences between your local files and the files stored in the remote repository, pull (fetch) the most up-to-date version of the remote repository onto your computer each time you start working with the content.

1. Launch GitBook.
2. From the GitBook’s top menu, select ***GitBook Editor*** > ***Open*** and navigate to the repository folder.
3. Click ***Select folder***.

The repository files load into GitBook.

1. From the GitBook’s top menu, select ***GitBook Editor*** > ***Open*** and navigate to the repository folder.
2. Go to the GitBook’s top menu and select ***Book*** > ***Pull***.

Your local repository is in sync with the remote repository.

1. Optional: If there are conflicts to resolve, follow the How to resolve a conflict instructions.
2. From the ***Table of Contents*** in GitBook’s left pane, select the article you want to edit.

The article opens in the right-hand pane.

1. Optional: Click the question mark  > ***Edit Markdown*** to toggle the article view.
2. Update the article.
3. Click the ***Save*** button to save changes in your local repository.
4. In the ***Commit changes*** window, edit the ***Message*** field to briefly describe what changes were made.
5. Click the ***Commit*** button.
6. Click the Publish & Sync button  to push your local changes to the remote repository.

## How to create a new article

1. Launch GitBook.
2. From the GitBook’s top menu, select ***GitBook Editor*** > ***Open*** and navigate to the repository folder.
3. Click ***Select folder***.

The repository files load into GitBook.

1. From the GitBook’s top menu, select ***GitBook Editor*** > ***Open*** and navigate to the repository folder.
2. Go to the GitBook’s top menu and select ***Book*** > ***Pull***.

Your local repository is in sync with the remote repository.

1. Optional: If there are conflicts to resolve, follow the How to resolve a conflict instructions.
2. In the left pane, right-click anywhere in the ***Table of Contents*** view and select ***Add article***.
3. In the ***New Article*** window, edit the ***Title*** field.
4. Click the ***Add*** button.

The new article is added to the ***Table of Contents*** view.

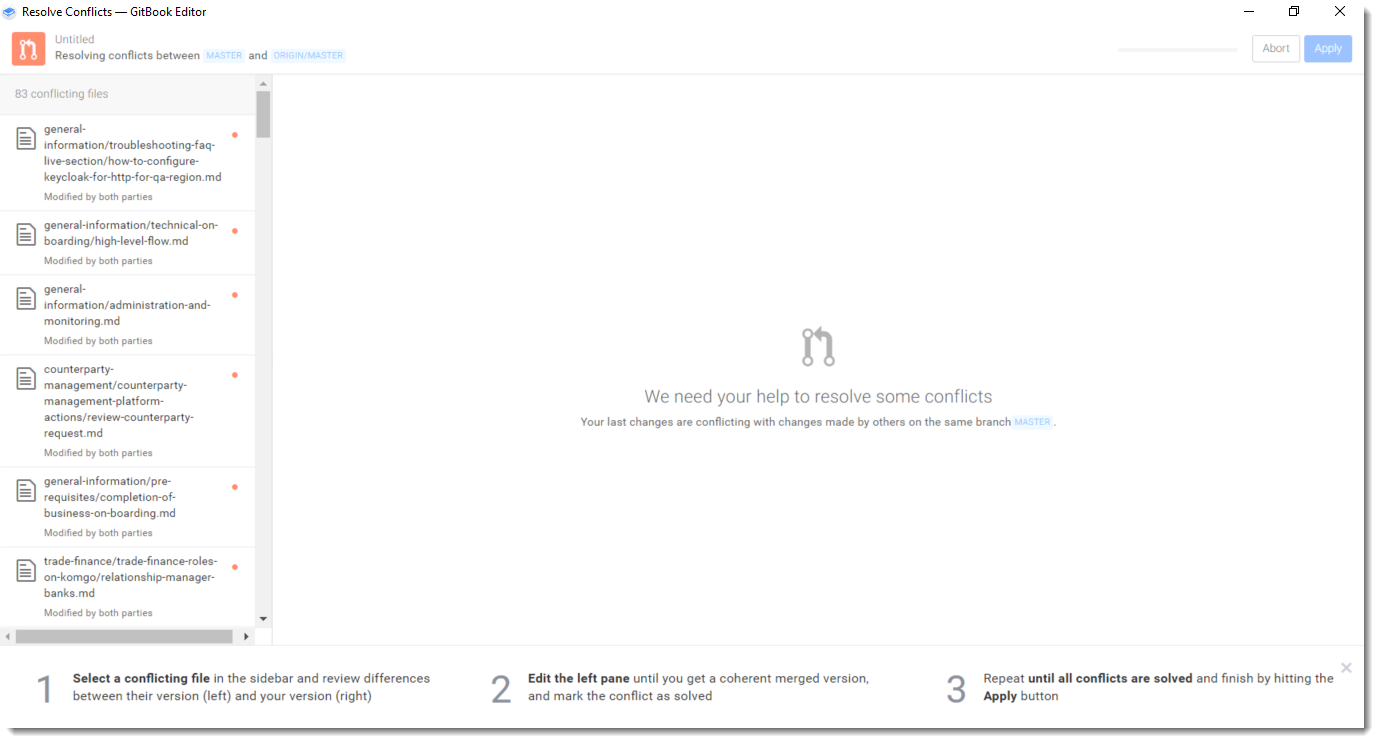
1. In the ***Table of Contents*** view, click on the new article.

The article opens in GitBook.

1. Optional: Click the question mark  > ***Edit Markdown*** to toggle the article view.
2. Create content of the new article.
3. In the ***Table of Contents*** view, drag and drop the new article to change its position.
4. Click the ***Save*** button to save changes in your local repository.
5. In the ***Commit changes*** window, edit the ***Message*** field to briefly describe what changes were made.
6. Click the ***Commit*** button.
7. Click the Publish & Sync button  to push your local changes to the remote repository.

## How to resolve a conflict

A conflict is when there are differences between your local files and the files stored in the remote repository. Conflicts occur when two people work on the same piece of content at the same time. Conflicts need to be resolved to enable the Help Portal to be built correctly.



1. In the ***Resolve Conflicts*** view, select a conflicting file in the sidebar and review differences between the remote repository and your local repository.
2. Edit the left pane until you get a coherent merged version.
3. Optional: To reset your local changes to the changes in the remote repository, click the ***Reset*** button in the left pane.
4. Optional: To choose your local version over the remote repository, click the ***Pick*** button in the right pane.
5. Mark the conflict as solved by clicking the ***Mark as Solved*** button.
6. Repeat until all conflicts are solved.
7. Click the ***Apply*** button.
8. In the ***Confirm conflicts resolution*** window, click the ***Confirm*** button.

# How to review the content

You can provide review feedback to content authors using Stackedit.io: https://stackedit.io/app#

## Connecting to the GitLab repository

To use Stackedit.io, you must first connect to the GitLab source repository.

1. Navigate to [https://stackedit.io/app#](https://stackedit.io/app)
2. Click the Toggle side bar icon  ­to open ***Menu***.
3. In the ***Menu***, select ***Workspaces*** > ***Add a GitLab backed workspace.***
4. In the ***Link your GitLab account to StackEdit*** window, enter the URL of the repository in the ***GitLab URL*** field and your ID in the ***Application ID*** field.
5. Click ***OK***.

Stackedit.io opens a new tab showing the files in the source repository.

## Providing feedback using Stackedit.io

1. Navigate to [https://stackedit.io/app#](https://stackedit.io/app)
2. Open the Help Portal workspace.
3. In the left-hand navigation, click the Toggle explorer icon  and navigate to the files you want to review.

The Summary.md file shows the table of contents. This file provides the path to the individual files.

1. Click a file to open it.

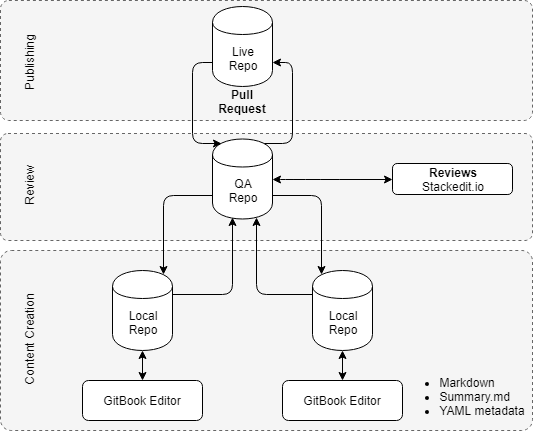
The file opens in the right-hand pane. ­

1. Highlight the text you want to add your feedback to.
2. Click the comment icon  to add a comment. ­
3. Click the Synchronise now icon  to save your changes to the repository.

# How to publish the content

For the MVP version of the Komgo Help Portal, content is automatically published when it is committed (pushed) to the remote GitLab repository. The Help Portal is then distributed to the Komgo nodes.

For the production version, an additional step will be added to enable updates to be coordinated with product updates.



# Managing the style and look of the Help Portal

The portal is built from the Markdown files using the Jekyll processor: https://jekyllrb.com/.

The scripts and resource files required for this are saved in the following repository:

* [Add URL]

# Resources

* GitBook Help Center   
  <https://help.gitbook.com/>
* GitLab Help   
  <https://gitlab.com/help>
* Stackedit.io  
  https://stackedit.io/app#
* Markdown Guide   
  https://www.markdownguide.org/
* Markdown Cheat Sheet   
  <https://guides.github.com/pdfs/markdown-cheatsheet-online.pdf>