

JaJuMa

git 4 Page Builder

06/2023



Table of Contents

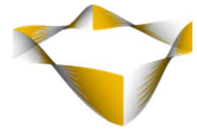
1.	Introduction.....	- 3 -
2.	Installation	- 4 -
2.1.	Extension Installation via composer.....	- 4 -
2.2.	Manual Extension Installation via FTP.....	- 4 -
3.	Extension Scope	- 5 -
4.	Configuration.....	- 6 -
4.1.	General Configuration.....	- 7 -
4.2.	git Setup	- 9 -
4.2.1.	git Directory	- 10 -
4.2.2.	Use Remote git Repository.....	- 10 -
4.2.2.1.	Auto git Pull	- 11 -
4.2.2.2.	Auto git Push.....	- 11 -
4.2.3.	Auto git Commit	- 12 -
4.2.3.1.	git Commit Message.....	- 12 -
4.3.	Hyvä Themes Configs	- 13 -
4.3.1.	Deploy with TailwindCSS	- 13 -
4.3.1.1.	Select current Hyvä Theme	- 13 -
5.	git 4 Page Builder User Interface.....	- 14 -
5.1.	git 4 Page Builder Toolbar.....	- 14 -
5.2.	Template List.....	- 17 -
5.3.	Template History.....	- 18 -
6.	git 4 Page Builder CLI.....	- 19 -
6.1.	Import Page Builder Content Templates from CLI.....	- 19 -
6.2.	Export Page Builder Templates from CLI.....	- 19 -
7.	Using git 4 Page Builder	- 20 -
7.1.	Map Templates To Page Builder Input.....	- 20 -
7.2.	Export Templates / Content Updates.....	- 21 -
7.3.	Import Templates	- 21 -
7.4.	Clone Templates	- 22 -
7.5.	Restore Previous Version.....	- 23 -
7.6.	Deploy Page Builder Content	- 24 -
8.	Using git 4 Page Builder With Hyvä Themes.....	- 25 -
8.1.	Include Static Content in TailwindCSS Purge.....	- 25 -
8.2.	Hyvä Tailwind CMS JIT Integration	- 25 -
9.	Support.....	- 26 -



1. Introduction

[JaJuMa git 4 Page Builder](#) for Magento 2 enables Magento stores to use git for managing static content.

Please see below for more details and configuration options provided by this extension.



2. Installation

For installing the extension, follow installation process as with any Magento Extension.

2.1. Extension Installation via composer

For installing the extension via composer, follow installation process as with any Magento Extension from Magento Marketplace.

See: <https://devdocs.magento.com/extensions/install/>

2.2. Manual Extension Installation via FTP

For manual installation by FTP, please follow these steps:

Before Installing

1. We recommend you to duplicate your live store on a staging/test site and try installation on your staging/test site before deploying to your live store
2. Backup Magento files and the store database

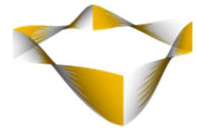
Please Note: It's very important to backup all themes and extensions in Magento before installation, especially when you are working on a live server.

We strongly recommend you to do not skip this step.

Upload the Extension

1. Log into your hosting space via a FTP client (e. g. FileZilla, WinSCP, cuteFtp)
2. Create Folder: <magentoroot>/app/code/Jajuma/Git4PageBuilder
3. Unzip extension package and upload files into:
<magentoroot>/app/code/Jajuma/Git4PageBuilder
4. Enter and run the following commands at the command line:

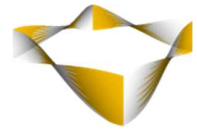
```
php bin/magento setup:upgrade  
php bin/magento setup:static-content:deploy
```



3. Extension Scope

[JaJuMa git 4 Page Builder](#) for Magento 2 provides following improvements for Page Builder content management workflows:

- Support for local and remote git repositories to support different use cases
- Content versioning/history/backup
- Easy content deployments
- Simple UI In backend for content creators
- CLI (Command Line Interface) flexibility for developers
- TailwindCSS support for Hyvä Themes (Hyvä CMS JIT & regular purge)



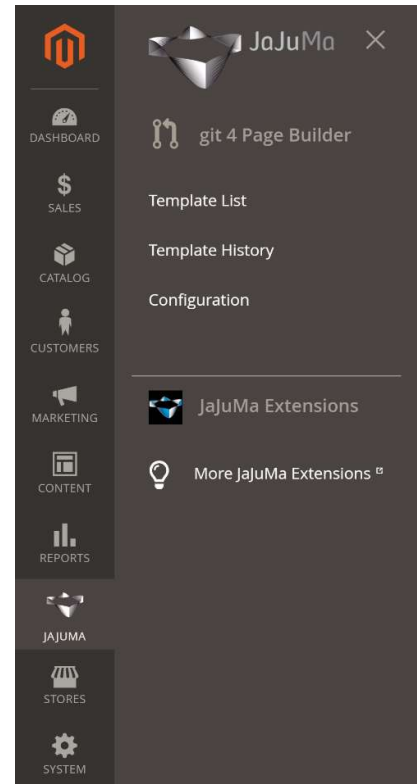
4. Configuration

In Magento Backend see

JaJuMa -> git 4 Page Builder -> Configuration

For JaJuMa git 4 Page Builder configuration with 3 sections:

General Configuration	⌵
Git Setup	⌵
Hyvä Themes Configs	⌵



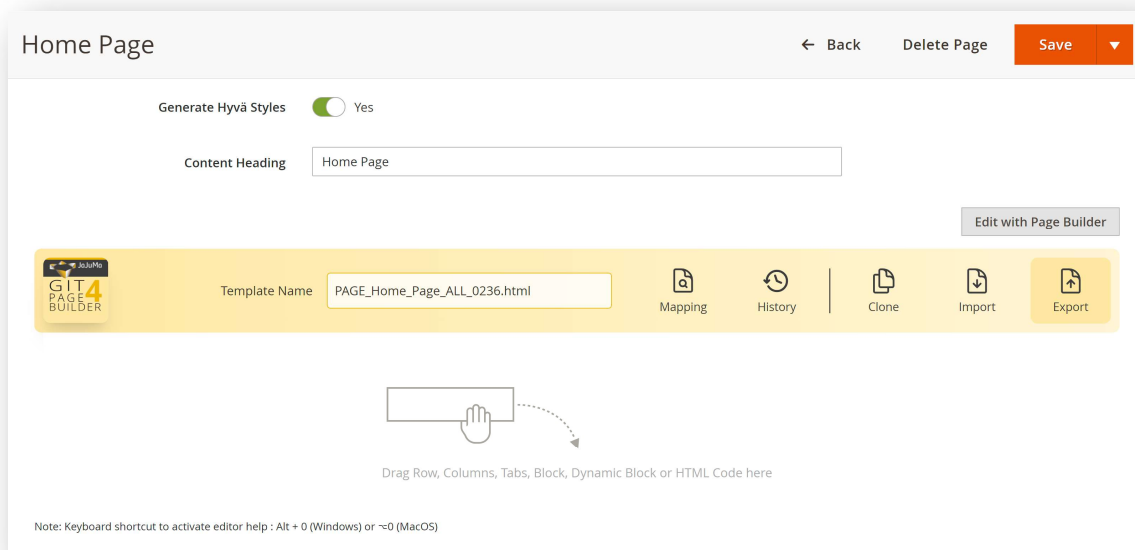
To manage content managed via git 4 Page Builder see

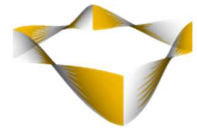
JaJuMa -> git 4 Page Builder -> Template List

To see a global history for content managed via git 4 Page Builder see

JaJuMa -> git 4 Page Builder -> Template History

For adding content to git and use the main features, please see the **git 4 Page Builder Toolbar** added to any Page Builder input in your backend:





4.1. General Configuration

The general configurations in

JaJuMa -> git 4 Page Builder -> Configuration

provide the options to

- Enable/Disable the extension &
- Configure the Lookup ID

General Configuration ⌵

Enable [global] Use system value

Lookup ID [global] Use system value

Default: *id,page_id,block_id*
Note: No need to change this config when using Page Builder with Magento default content types only.
 It may also work with certain 3rd party integrations using Page Builder such as e. g. [JaJuMa Category Grid Callouts for Magento 2](#) or some Blog extensions.
 For using this extension with custom content types that are utilizing Page Builder, please see the manual for further instructions on how to find the correct Lookup ID.

Enable:

To enable git 4 Page Builder, select from Drop Down:

- Yes → git 4 Page Builder is enabled
- No → git 4 Page Builder is disabled

Lookup ID:

When using Page Builder only for Magento default content types (CMS Pages & Blocks, Category Descriptions, Product Descriptions) this config does not need to be changed.

These content types are covered with the comma separated list of Lookup IDs:

id, page_id, block_id

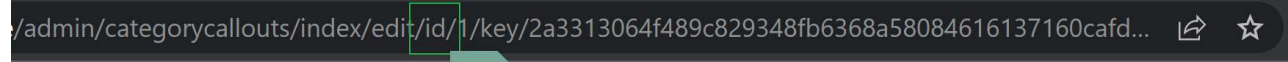
For using git 4 Page Builder for custom content types, please navigate to your custom Page Builder input in your backend and check the URL and find the ID qualifier as explained below:

**Example 1 – JaJuMa Category Grid Callouts:**

For example for [JaJuMa Category Grid Callouts](#) the URL will look as follows.

With ID qualifier = “id”:

```
/admin/categorycallouts/index/edit/id/1/key/2a3313064f489c829348fb6368a58084616137160cafd...
```

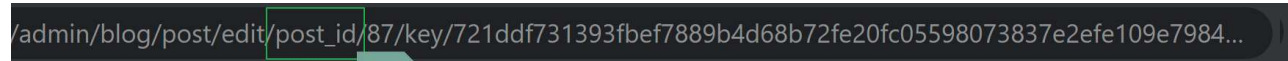


→ git 4 Page Builder will work with [Category Grid Callouts](#) automatically as “id” is already included in list of default Lookup IDs.

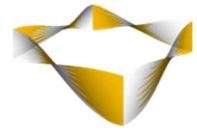
Example 2 – Blog Extension:

If the URL looks e. g. for your blog extension as follows, with ID qualifier = “post_id”:

```
/admin/blog/post/edit/post_id/87/key/721ddf731393fbef7889b4d68b72fe20fc05598073837e2efe109e7984...
```



→ Please add “post_id” to the comma separated list of Lookup IDs to use git 4 Page Builder with your blog extension.



4.2. git Setup

For using this extension you need to setup a git repository.
This can be either a local only or a remote git repository.

NOTE: For instructions and guidance on how to setup and configure git, please refer to generally available documentations.

Notes:

- With local only repository, git 4 Page Builder can be used for versioning, backup and restores. Content deployments via backend will not be available.
- For remote repository you can use any service such as Bitbucket, GitHub or Gitlab.
- A remote repository needs to be setup with push permissions for your web user
- Technically you could use the same repo and branch as for your Magento source code.
However, it is recommended to setup a separate repo or separate branch for your static content managed with git 4 Page Builder extension.

Please see below configuration options and explanations with points to consider when setting up the git repo for your static content.

git Setup ⊞

git Directory [global] Use system value

Please input the path/folder name for your git repository.
Config is relative to PARENT directory of your Magento root folder.
E. g. with default value `git4pagebuilder`, your folder structure would look like this:

```

/public_html/
├─ [magento_root]/
├─ git4pagebuilder/
└─ .git

```

Use Remote git Repository [global] Use system value

If Yes (recommended): The extension will run in remote repository mode and you may configure auto push/pull below for quick and easy content deployments.
Note: Please make sure your web user has push permission for your remote repository.
If No: The extension will run in local repository mode only with limited features.

Auto git Pull [global] Use system value

If Yes (recommended): A git pull will be executed whenever a page in backend with a Page Builder input is visited in backend.
If No: git pull is disabled and will not be run automatically by the extension. In this case, please use CLI for content deployments.

Auto git Push [global] Use system value

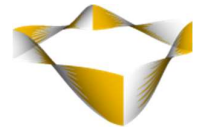
If Yes (recommended): A git push will be executed whenever a content update is exported in backend in a Page Builder input.
Note: Please make sure your web user has push permission for your remote repository.
If No: git push is disabled and will not be run automatically by the extension. In this case, please use CLI for content deployments.

Auto git Commit [global] Use system value

If Yes (recommended): The extension will ask for a commit message when click "Export" button and when confirm the message, the Page Builder content will be exported and committed to git.
If No: git commit is disabled and will not be run automatically by the extension. In this case, please use CLI for commits.

git Commit Message [global] Use system value

With the git Commit Message config you may define a default commit message.
With every commit via git 4 Page Builder toolbar you still have the option to change the default message.



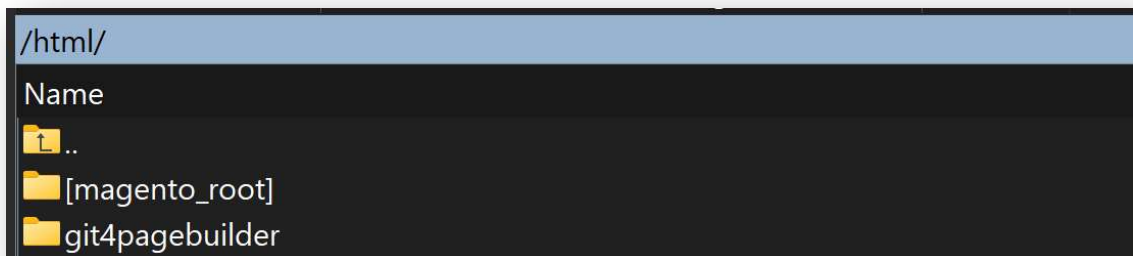
4.2.1. git Directory

The **git Directory** config is for defining the directory in which you setup the git repository for your static content.

The config is a text input for path/folder name, relative to PARENT directory of your Magento root folder.

Note: Please make sure your web user has write permission for your git directory.

E. g. when configure “*git4pagebuilder*” as git directory, the folder structure on your server would look like this:



4.2.2. Use Remote git Repository

With the “**Use Remote git Repository**” you can select if a local or a remote repository should be used.

If Yes:

When setting up your git with a remote repository, the extension can automatically pull/push your content to your remote repo. This allows easy deployments of content between different environments.

Please also see additional configurations below.

Note: Please make sure your web user has push permission for your remote repository.

If No:

When setting up your git as local repository you can use this extension for content versioning/backup and easy restores of previous status.



4.2.2.1. Auto git Pull

The **Auto git Pull** config is for enable/disable auto git pull function by git 4 Page Builder.

If Yes:

If enabled, a git pull will be executed whenever a page in backend with a Page Builder input is visited in backend.

This option allows simple and easy content deployments between different environments from backend.

If No:

git pull is disabled and will not be run automatically by the extension.

In this case, please use CLI for content deployments.

4.2.2.2. Auto git Push

The **Auto git Push** config is for enable/disable auto git push function by git 4 Page Builder.

If Yes:

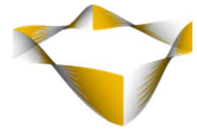
If enabled, a git push will be executed whenever a content update is exported in backend in a Page Builder input.

This option allows simple and easy content deployments between different environments from backend.

If No:

git push is disabled and will not be run automatically by the extension.

In this case, please use CLI for content deployments.



4.2.3. Auto git Commit

The **Auto git Commit** config is for enable/disable auto git commit function by git 4 Page Builder.

If Yes:

If enabled, the extension will ask for a commit message when click “Export” button and when confirm the message, the Page Builder content will be exported and committed to git.

If No:

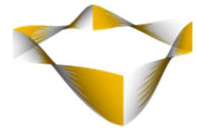
git commit is disabled and will not be run automatically by the extension.

In this case, please use CLI for commits.

4.2.3.1. git Commit Message

With the **git Commit Message** config you may define a default commit message.

With every commit via git 4 Page Builder toolbar you still have the option to change the default message.



4.3. Hyvä Themes Configs

This extension was built to work with Hyvä Themes and supporting TailwindCSS.

Hyvä Themes Configs ⌵

Deploy with Tailwind CSS [global] Use system value

If Yes: TailwindCSS styles will be generated for the theme as configured below and saved to database same as with Hyvä Tailwind CMS JIT module when importing content from CLI.
If No: TailwindCSS styles will not be generated when importing content via CLI.

Select current Hyvä theme [global]

Please select the Hyvä Theme to be used for generating TailwindCSS styles.

4.3.1. Deploy with TailwindCSS

The **Deploy with TailwindCSS** config is for enable/disable the generation of Tailwind styles on import from CLI.

If Yes:

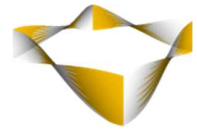
If enabled, the TailwindCSS styles will be generated for the theme as configured below and saved to database same as with Hyvä Tailwind CMS JIT module when importing content from CLI.

If No:

TailwindCSS styles will not be generated when importing content via CLI.

4.3.1.1. Select current Hyvä Theme

Please select the Hyvä Theme to be used for generating TailwindCSS styles.



5. git 4 Page Builder User Interface

git 4 Page Builder enables you to manage your content via git right from your Magento backend. Please see below description for each of the UI elements provided:

(Note: The git 4 Page Builder UI will not show when create e. g. a new page, but only show after the content has been saved once).

5.1. git 4 Page Builder Toolbar

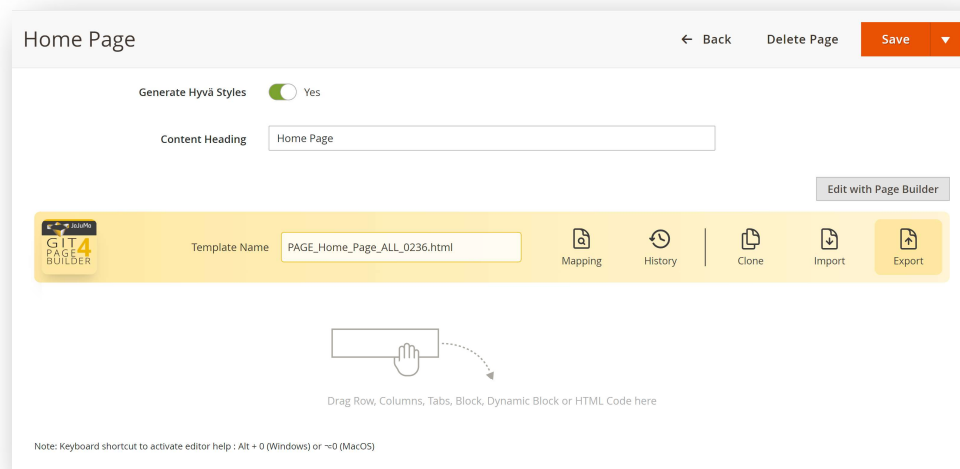


Figure 1 - git 4 Page Builder Toolbar

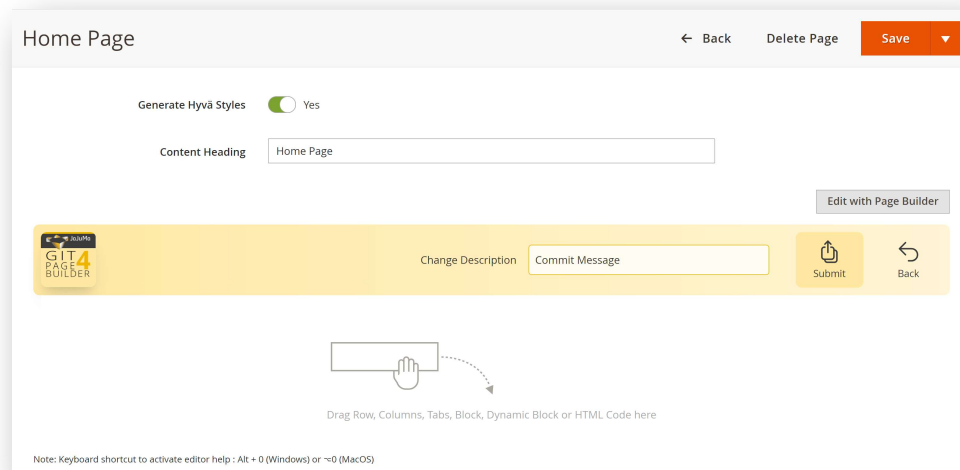


Figure 2 - Export

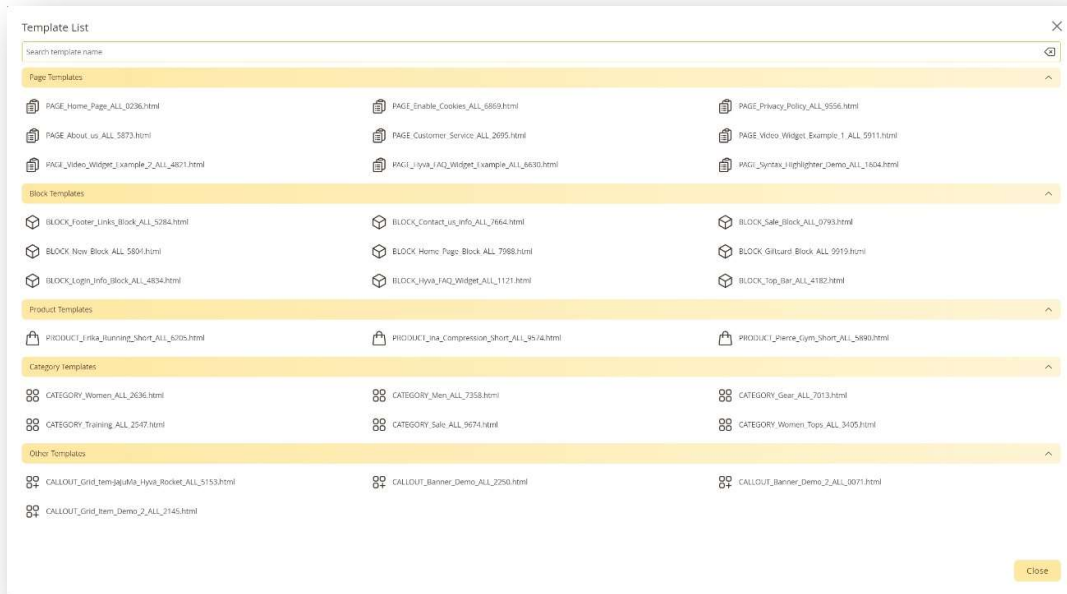


Figure 3 - Template List

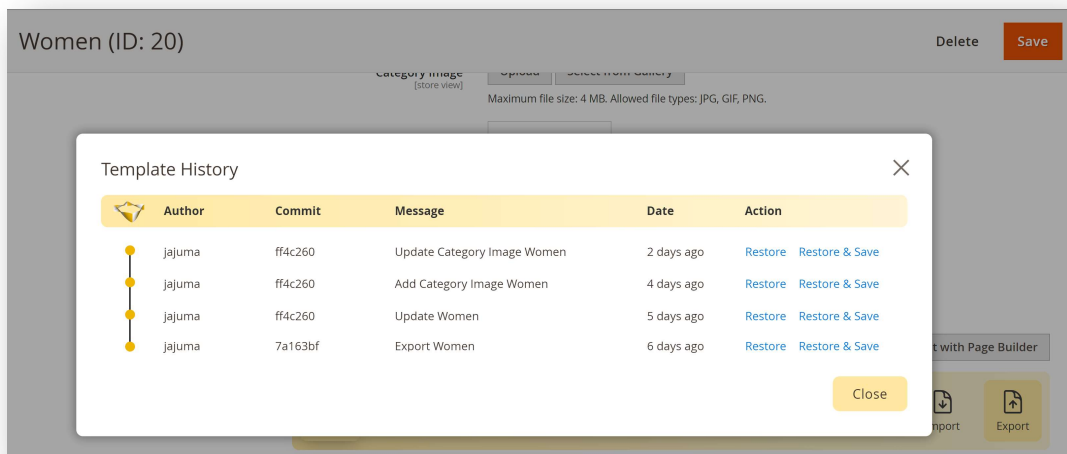


Figure 4 - Template History

**Template Name:**

Input the name for the Page Builder content to be managed via git.

The extension will automatically add post fixes for Storeview (ALL or Store Code) and a numeric ID to enforce unique template names.

Mapping:

When click “Mapping”, a popup will open showing all templates currently available in your content git repository. From here, templates can be selected for import or clone.

(See Figure 3 - Template List)

History:

When click “History”, a popup will open showing the history with all previous versions for the currently mapped template. For each version, there are options to restore or fully restore. (See Figure 4 - Template History)

Clone:

The “Clone” button allows cloning the content from a template that is already mapped to another Page Builder input into current Page Builder.

After cloning a template you may edit and export as new template mapped to current Page Builder input.

Import:

When click “Import”, latest version from git will be imported into Page Builder input.

After import you may save the page with the updated content.

Export:

When click “Export”, current content will be saved to your git repository folder.

If Auto Commit is enabled, the extension will ask for a commit message and after confirm by click on “Commit”, commit the update as new version to git.

Change Description:

The “Change Description” (a.k.a. commit message) describes the change or version submitted. Each submitted version will show with this message in content history.

Submit:

When click “Submit”, current content will be committed as new version to git.

Back:

The “Back” cancels the export process and allows to do further editing before commit.



5.2. Template List

The Template List overview in

JaJuMa -> git 4 Page Builder -> Template List

provides a list of all content templates currently managed via git.

The Action column allows to

- Delete templates via **Select -> Delete**
- View the history for each template via **Select -> History**

Template List

49 records found

20 per page 1 of 3

ID	Full Action Name	Template Name	Type	Local Template ID	Store View	Date Created	Date Updated	Action
12	cms_page_edit	PAGE_404_Not_Found_ALL_8270.html	content	1	All Store Views	May 28, 2023 9:26:00 AM	May 28, 2023 9:26:00 AM	Select
13	cms_page_edit	PAGE_1_home_Page_ALL_0236.html	content	2	All Store Views	May 28, 2023 9:26:46 AM	May 28, 2023 9:26:46 AM	Select
14	cms_page_edit	PAGE_Enable_Cookies_ALL_6869.html	content	3	All Store Views	May 28, 2023 9:27:25 AM	May 28, 2023 9:27:25 AM	Select
15	cms_page_edit	PAGE_Privacy_Policy_ALL_9556.html	content	4	All Store Views	May 28, 2023 9:27:58 AM	May 28, 2023 9:27:58 AM	Select
16	cms_page_edit	PAGE_About_us_ALL_5873.html	content	5	All Store Views	May 28, 2023 9:28:29 AM	May 28, 2023 9:28:29 AM	Select
17	cms_page_edit	PAGE_Customer_Service_ALL_2695.html	content	6	All Store Views	May 28, 2023 9:29:00 AM	May 28, 2023 9:29:00 AM	Select
18	cms_page_edit	PAGE_Video_Widget_Example_1_ALL_5911.html	content	7	All Store Views	May 28, 2023 9:29:40 AM	May 28, 2023 9:29:40 AM	Select
19	cms_page_edit	PAGE_Video_Widget_Example_2_ALL_4821.html	content	8	All Store Views	May 28, 2023 9:30:10 AM	May 28, 2023 9:30:10 AM	Select
20	cms_page_edit	PAGE_Hyva_FAQ_Widget_Example_ALL_6630.html	content	9	All Store Views	May 28, 2023 9:30:49 AM	May 28, 2023 9:30:49 AM	Select
21	cms_page_edit	PAGE_Syntax_Highlighter_Demo_ALL_1604.html	content	10	All Store Views	May 28, 2023 9:31:25 AM	May 28, 2023 9:31:25 AM	Select
22	cms_block_edit	BLOCK_Footer_Links_Block_ALL_5284.html	content	1	All Store Views	May 28, 2023 9:33:30 AM	May 28, 2023 9:33:30 AM	Select
23	cms_block_edit	BLOCK_Contact_us_Info_ALL_7664.html	content	2	All Store Views	May 28, 2023 9:34:00 AM	May 28, 2023 9:34:00 AM	Select
24	cms_block_edit	BLOCK_Sale_Left_Menu_Block_ALL_0732.html	content	3	All Store Views	May 28, 2023 9:34:34 AM	May 28, 2023 9:34:34 AM	Select
25	cms_block_edit	BLOCK_Gear_Left_Menu_Block_ALL_8686.html	content	4	All Store Views	May 28, 2023 9:35:08 AM	May 28, 2023 9:35:08 AM	Select
26	cms_block_edit	BLOCK_Men_Left_Menu_Block_ALL_3855.html	content	5	All Store Views	May 28, 2023 9:35:40 AM	May 28, 2023 9:35:40 AM	Select
27	cms_block_edit	BLOCK_Women_Left_Menu_Block_ALL_7862.html	content	6	All Store Views	May 28, 2023 9:37:26 AM	May 28, 2023 9:37:26 AM	Select















5.3. Template History

The Template History view in

JaJuMa -> git 4 Page Builder -> Template History

provides a global history, listing all content changes done before in chronological order:

	Author	Commit	Message	Date
	jajuma	32e04f5	Update Category Grid Callout Summer	47 minutes ago
	jajuma	114b76a	Removed Promotion Homepage	51 minutes ago
	jajuma	2f9ffa9	Removed Promotion Men	52 minutes ago
	jajuma	31976d4	Replace Deleted Video on Video Widget Example 2	88 minutes ago
	jajuma	f20d6a4	Add Promotion Men	2 weeks ago
	jajuma	ff4c260	Add Promotion Homepage	3 weeks ago
	jajuma	7ae6add	Update Grid Item Demo 2	3 weeks ago
	jajuma	e7e0370	Export Grid Item Demo 2	3 weeks ago
	jajuma	1a26794	Export Banner Demo	4 weeks ago
	jajuma	bafea5c	Export Homepage	4 weeks ago

[Load More](#) 



6. git 4 Page Builder CLI

This extension includes a Command Line Interface (CLI) with commands as described below. Besides these commands you may use git normally as when managing code.

6.1. Import Page Builder Content Templates from CLI

Import one specific Page Builder Template.

Example with Template “*page-builder_Homepage*”:

```
php bin/magento jajuma:git4PageBuilder:import --object_name='page-builder_Homepage
```

Import ALL Page Builder Templates:

```
php bin/magento jajuma:git4PageBuilder:import --all
```

6.2. Export Page Builder Templates from CLI

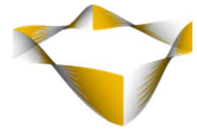
Export one specific Page Builder Template.

Example with Template “*page-builder_Homepage*”:

```
php bin/magento jajuma:git4PageBuilder:export --object_name='page-builder_Homepage
```

Export ALL Page Builder Templates:

```
php bin/magento jajuma:git4PageBuilder:export --all
```



7. Using git 4 Page Builder

Before using git 4 Page Builder, please make sure you have a git repository configured as described above, otherwise the module will not work.

Once git was set up, you can use the extension and manage your content as described below:

7.1. Map Templates To Page Builder Input

As first step, each template to be managed via git needs to be mapped to a Page Builder input.

E. g. a template "Homepage_New" will be mapped to static page with ID 10

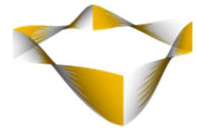
This mapping can be done by

- a) Define the template name and input into "Template Name" input in git 4 Page Builder Toolbar
- b) Click "Export"
- c) Define a change description &
- d) Confirm submit.

➔ Your content will be mapped to Page Builder input currently edited, exported and automatically saved (& committed/pushed to git).

No worries, you don't need to know about git. Everything happens automatically.

Once your content was exported, it will be available in template list as well as history for tracing any changes done, for cloning into another Page Builder input and for restoring previous status if needed.



7.2. Export Templates / Content Updates

Once a Template has been mapped to a Page Builder input via Backend, each content update can be exported as new version by:

- a) Click “Export”
- b) Define a change description
- c) Confirm submit

Notes:

- Templates can also be exported via command line (CLI) with commands described above
- If git Auto git Push is enabled, each export will also be pushed to remote git repository

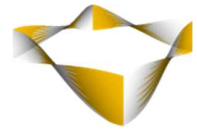
7.3. Import Templates

If you want to update a Page Builder content to the last version saved before for the mapped Template, simply click “Import”

→ Your Page Builder content will be updated to the latest version.

Notes:

- If git 4 Page Builder was setup with a remote repository, you can even do your content changes e. g. on a staging site, export from there and then directly import your latest version from staging into your live site.
- For importing an already mapped template, please use the “Clone” option (see next section)
- Templates can also be imported via command line (CLI) with commands described above

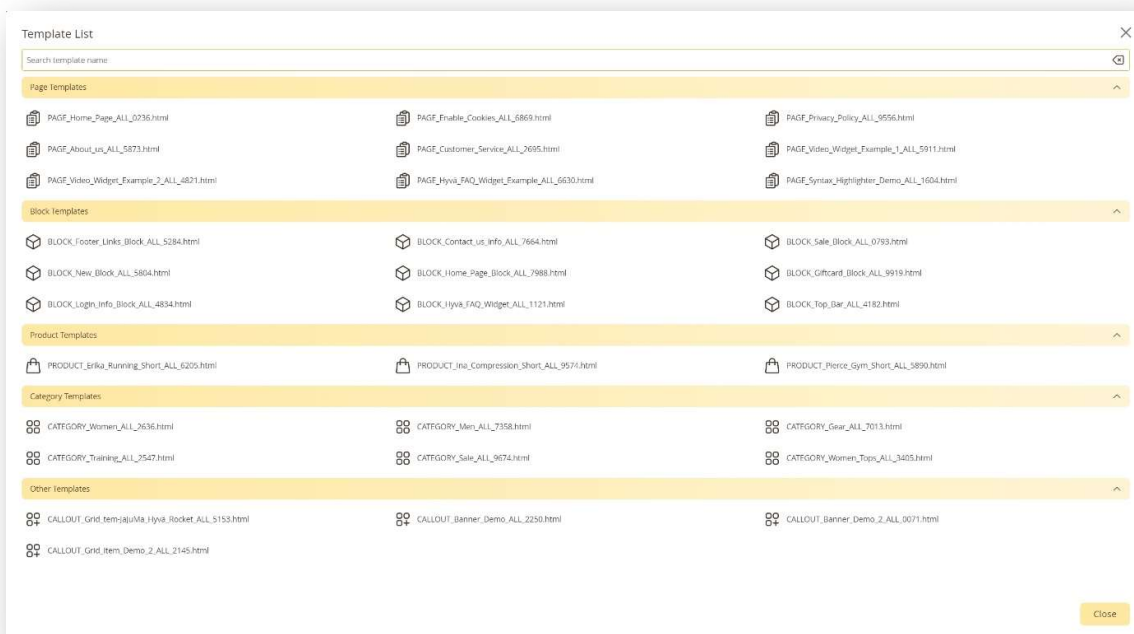


7.4. Clone Templates

If you want to clone an existing and already mapped Template into another Page Builder content you can do this as follows:

- First click the "Mapping" button
- In popup shown, select the template you want to clone.
- Then click the "Clone" Button.

➔ Your Page Builder content will be updated with content from cloned template and can be exported as a new template for further editing.





7.5. Restore Previous Version

If you want to restore a previous content version, please

- Click “History” button
- In popup shown, find the version you want to restore
- Click either
 - **“Restore”**
→ Template version will be restored to Page Builder input.
After reviewing/updating, please export again as a new version & save your content
or
 - **“Restore & Save”**
→ Template version will be restored and depending on extension configs, a git commit git push as well as save will be done automatically

Notes:

- You may also restore previous template versions using git as you would do with a code file, commit/push as needed and import restored object again.

Women (ID: 20) Delete Save

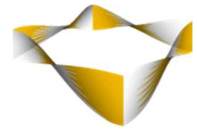
Category image (Store view) Maximum file size: 4 MB. Allowed file types: JPG, GIF, PNG.

Template History ✕

	Author	Commit	Message	Date	Action
●	jajuma	ff4c260	Update Category Image Women	2 days ago	Restore Restore & Save
●	jajuma	ff4c260	Add Category Image Women	4 days ago	Restore Restore & Save
●	jajuma	ff4c260	Update Women	5 days ago	Restore Restore & Save
●	jajuma	7a163bf	Export Women	6 days ago	Restore Restore & Save

Close

Import
Export



7.6. Deploy Page Builder Content

With git 4 Page Builder setup using a remote repository and Auto git Push/Pull enabled, you can deploy Page Builder content between environments as follows

- a) Create/Edit content on environment A (e. g. local/Test/Stage Site)
- b) Export template
- c) Switch to environment B backend (e .g. Live site)
- d) Navigate to the content you want to update
- e) Click "Import" & Save.

Done.

Your content is updated on environment B to the same status as on environment A.



8. Using git 4 Page Builder With Hyvä Themes

When updating content from backend, Tailwind styles will be generated normally as when not using git 4 Page Builder

Additionally, there are 2 options for generating Tailwind styles for your static content with this extension:

- A) Include your static content in TailwindCSS purge scope
- B) Use Hyvä Tailwind CMS JIT module

8.1. Include Static Content in TailwindCSS Purge

With git 4 Page Builder, static content is saved on disk as plain text files.

By this, you can include your git Directory in your purge scope by simply updating your `tailwind.config.js`

8.2. Hyvä Tailwind CMS JIT Integration

Git 4 Page Builder includes a Hyvä Tailwind CMS JIT integration.

When updating/importing content via CLI, Tailwind styles can be generated automatically same as with Hyvä Tailwind CMS JIT when saving content from backend.

For this to work, please make sure to

- have Hyvä Tailwind CMS JIT module installed and
- enabled for your Page Builder content
- “Deploy with TailwindCSS” option enabled and
- Hyvä Theme to be used configured correctly

Note: Generating Tailwind styles on import is currently only supported for content types as supported by Hyvä Tailwind CMS JIT module + Magefan blog with required patch installed.



9. Support

Please feel free to contact JaJuMa support team via support@jajuma.de.

In case any additional information is required. We'd be more than happy to assist in setting up the extension.