



docs-as-code what, who & how?

Petra Herbst | 28.01.2022

[#creatingtomorrow](#)



Petra Herbst

- 1996 – 2002 ITAT Graz University
- 2001 – 2009 Localization Coordinator and Technical Writer @Hyperwave Software
- 2009 – 2016 Head of Documenation @Unycom Software
- 2016 – 2019 Coordinator Technical Documentation @BearingPoint
- 2019 – 2020 Content Manager / Technical Writer @Travis CI
- 2020 – today Product Advocate & Technical Writer @viesure innovation center GmbH



LinkedIn:

[linkedin.com/in/petra-herbst](https://www.linkedin.com/in/petra-herbst)

XING:

https://www.xing.com/profile/Petra_Herbst/cv

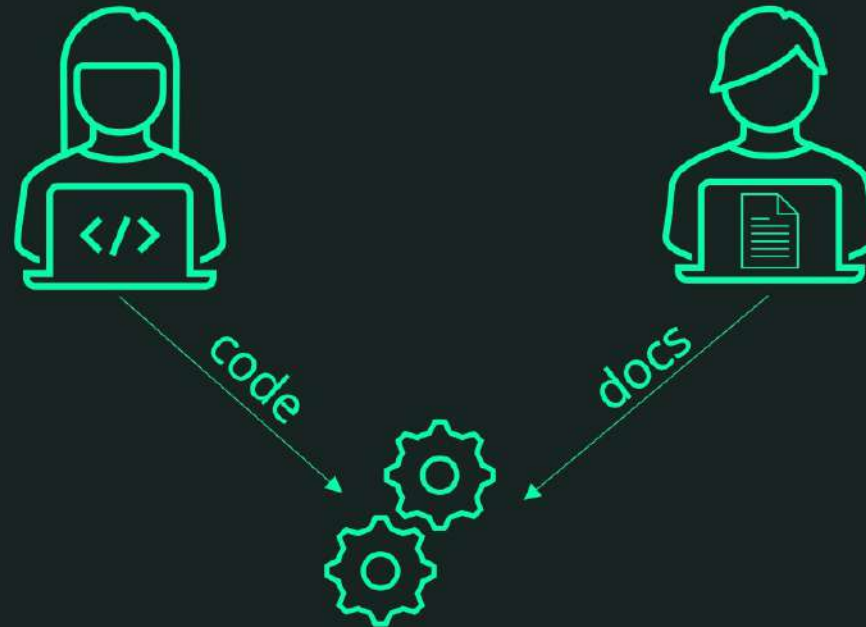
Agenda

- Term definition and answers to questions
 - What are docs-as-code?
 - Who needs them?
 - How do you implement them?
 - Is there an alternative?
- Main involved components
- Advantages & disadvantages
- Process including screenshots from viesure

docs-as-code

- Next to a variety of other documentation tools such as
 - Word processors (GoogleDocs, MS Word, etc.)
 - Authoring systems (FrameMaker, Madcap Flare, etc.)
 - Wikis (Altassian Confluence, etc.)
- **Markup Code Toolchains** (docToolchain, Gatsby, MkDocs, etc.) emerged mainly from the software documentation side

docs-as-code – in short



Documentation is treated similarly to code

Main components

- **Text files** (markup, markdown in contrast to binary formats such as in Adobe FrameMaker or Microsoft Word)
- **Text editor** like [Atom](#) or [Sublime](#)
- **Static site generator**, mostly open source such as [Sphinx](#), [Jekyll](#) or [Hugo](#)
- Storing in a **version control system**, mostly [git](#)
- Collaboration using [git](#) (branch, merge, push)
- **CI/CD** for automating the build of the target files
- **Automated tests** (link checker, formatting, terminology)

Who needs docs-as-code?

„Lazy people“ like me 😎

because the computer takes over a lot of tech writer tasks
and thus gives tech writers time for other tasks

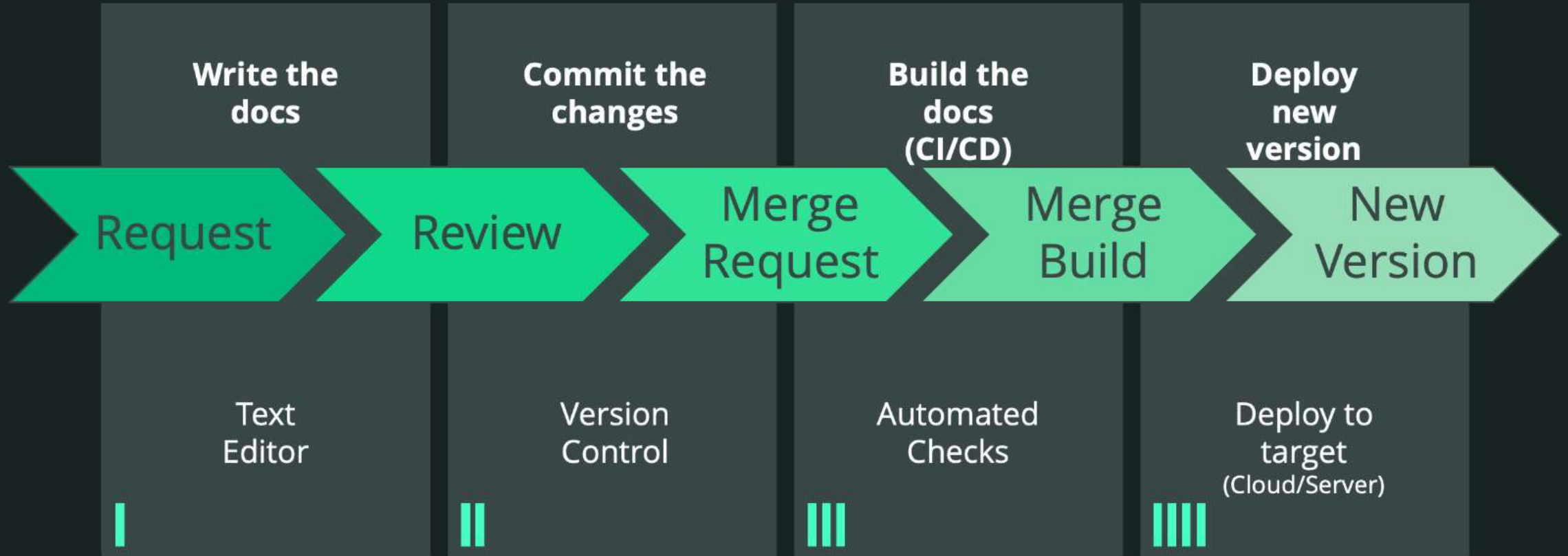
...and who doesn't?

- Marketing
- C Level, Management
 - 👉 they usually prefer fancy PowerPoint slides 😏

How do you implement them?

- Using the same infrastructure & tools as the developers
- Benefiting from the developers' process know-how and making their lives easier in the long run
- Profiting from open source tools
- Just do it 💪

Process



Markdown file

- In [Atom](#) editor

```
demo-text.md
1  ---
2  title: Authentication overview
3  description: General information on our authentication process
4  enableToc: false
5  ---
6
7  # Overview
8
9  Our Authentication API is used to authenticate your application.
10
11 :::info
12
13 You need to obtain an access token only once. See below how and where you can get it!
14
15 :::
16
17 You use your app's Key and Secret as client credentials to obtain an access token.
18
19 ## Where can I find my client credentials?
20
21 To retrieve your client credentials:
22
23 1. Sign in.
24
25 2. Click on your name in the top right corner.
26
27 3. Click on Apps.
28
29 4. Click on the name of the app for which you want to create a token.
30
31 5. Go to the API Keys section.
32
33 6. Here you can copy Key and Secret.
34
35
36 ## How can I get an access token?
37
```

Commit in GitLab

The screenshot shows the GitLab web interface for a repository named 'redocly-api-portal'. The left sidebar contains navigation options: Project information, Repository, Files (selected), Commits, Branches, Tags, Contributors, Graph, Compare, Issues (0), Jira, Merge requests (3), CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, Analytics, Wiki, Snippets, and Settings. The main content area is titled 'New file' and shows a file named 'demo-file.md' being created on the 'master' branch. The file content is as follows:

```
9 Our Authentication API is used to authenticate your application.
10
11 :::info
12
13 You need to obtain an access token only once. See below how and where you can get it!
14
15 :::
16
17 You use your app's Key and Secret as client credentials to obtain an access token.
18
19 ## Where can I find my client credentials?
20
21 To retrieve your client credentials:
22
23 1. Sign in.
24
25 2. Click on your name in the top right corner.
26
27 3. Click on Apps.
28
29 4. Click on the name of the app for which you want to create a token.
30
31 5. Go to the API Keys section.
32
33 6. Here you can copy Key and Secret.
34
35
36 ## How can I get an access token?
```

Below the code editor, the 'Commit message' field contains 'TRIN-000' and the 'Target Branch' dropdown is set to 'petra-demo'. A checkbox labeled 'Start a new merge request with these changes' is checked.

Successful build in GitLab

The screenshot shows the GitLab interface for the 'redocly-api-portal' project. The 'Pipelines' section is active, displaying a list of recent pipeline runs. The table below summarizes the visible data:

Status	Pipeline ID	Triggerer	Commit	Stages	Duration
passed	#95550 latest		change-doc-... -> a1a16339 TRIN-1973 added underli...		47 minutes ago
passed	#95542		change-doc-... -> 25f3ef83 TRIN-000 added MIT to ...		50 minutes ago
passed	#95387 latest		master -> 8fa08b6d Merge branch 'feature/T...		5 hours ago
passed	#94592		feature/TRI... -> bbf3c002 TRIN-1973 Update produ...		6 hours ago
passed	#94562		feature/TRI... -> 35902b30 TRIN-1973 Change all pr...		2 days ago

Source and target @viesure

The screenshot shows the GitLab interface for the 'redocly-api-portal' repository. The repository is owned by 'apigee' and has 156 commits, 22 branches, 3 tags, 14.8 MB files, and 14.8 MB storage. A commit by Petra Herbst is highlighted, showing a merge of 'feature/TRIN-1506' into 'master'. Below the commit, there is a table of files and their last commit details.

Name	Last commit	Last update
api	TRIN-1506 version upgrade to 1.1.0-beta.44	2 weeks ago
components	TRIN-1506 version upgrade to 1.1.0-beta.44	2 weeks ago
footer	TRIN-000 refactor & delete unused stuff	2 months ago
images	TRIN-000 refactor & delete unused stuff	2 months ago
openapi	TRIN-1498 tryout without session.page.yaml	2 months ago
snippets	TRIN-1608 adding link checker	3 weeks ago
static	TRIN-1448 fix robots.txt	2 months ago
.editorconfig	TRIN-000 refactor & delete unused stuff	2 months ago
.gitignore	TRIN-000 refactor & delete unused stuff	2 months ago
LICENSE	BETA-133 create erdocly portal from temp...	11 months ago
README.md	TRIN-000 refactor & delete unused stuff	2 months ago

The screenshot shows the Wiener Städtische Developer Portal website. The header includes 'API Catalog', 'Search the docs', and the user name 'Petra Herbst'. The main content area features the Wiener Städtische logo and the text 'WIENER STÄDTISCHE VIENNA INSURANCE GROUP Developer Portal'. Below this, it says 'A portal for Wiener Städtische APIs' and provides a 'Click here to get started' button. The footer contains three sections: 'API Catalog' (Have a look at our API Catalog and integrate the APIs you need to ease your customers' lives), 'Getting started' (Find out how to register as a developer and retrieve an API key to integrate Wiener Städtische APIs into your system), and 'FAQs' (Check out our FAQs and learn what others asked about our developer portal).

docs-as-code - advantages

- **Time** – any time on button click you can produce a new documentation version in just a few minutes
- **Developer friendly** > devs don't have to leave their IDEs
- **Versioning** and **traceability**
- **Separation of form and content**
- **Reusing** of contents
- **Low price** due to open source

docs-as-code - disadvantages

- **Time investment** of people involved – tech writer (team) and one or two developers
- Depending on the tools used potential additional **license costs**
- **Consultant costs** if the knowledge isn't available inhouse
- **Not suitable for any kind of documentation**, but definitely for software documentation

Are there any alternatives?

YES 😊 you can choose a all-in-one solution or simply start with *.asciidoc and ask a developer to do the automation

- for ordinary docs [Sphinx](#) or [Asciidoctor](#)
- for API documenation [redocly](#) (as we do @viesure)

Special case @viesure

- API documentation
- A tool was needed which not only generates documentation automatically out of the openapi.yaml file, but also offers the possibility to create additional „handwritten“ documentation and a Tryout feature and all this seamlessly displayed on a Developer Portal and (technical detail) providing the required Google Apigee integration) > [redocly](#)

A person wearing headphones and glasses is sitting at a desk with multiple computer monitors. The person is viewed from the side, looking at the screens. The background is a blurred office environment with a window and a plant.

Example

Wiener Städtische Developer Portal

<https://developers-sandbox.wienerstaedtische.at/>

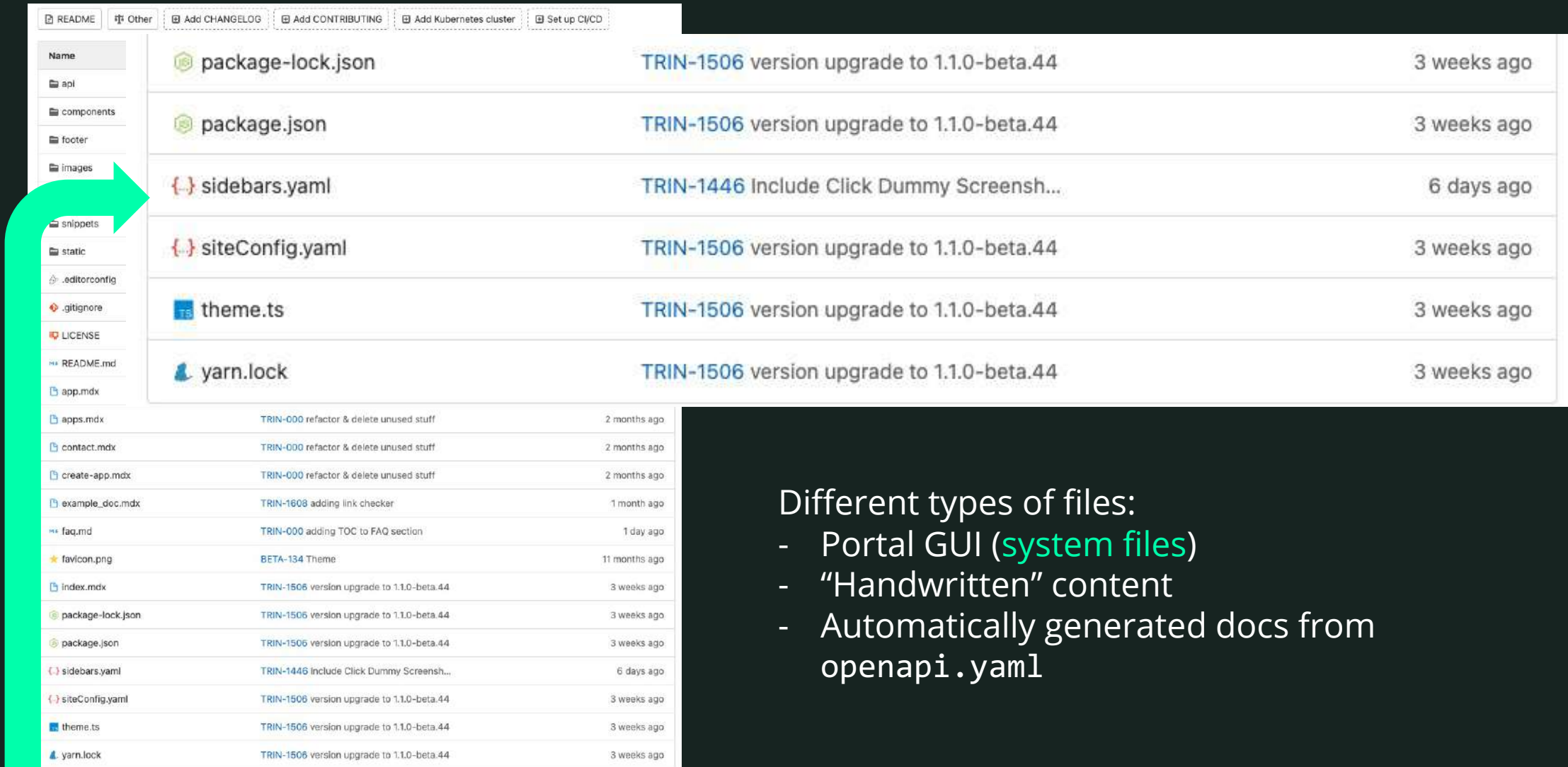
<https://developers.wienerstaedtische.at/>

#creatingtomorrow

Detailed components @viesure

- **Text files** – GitHub flavored markdown in combination with React Custom Components (solution offered out-of-the-box by [redocly](#))
- **Text editor** - Atom (locally) or [GitLab](#) Web IDE
- **Static site generator** – in redocly integrated [Gatsby](#)
- Storing in version control system - [GitLab](#)
- Collaboration via [GitLab](#) (branch, merge, push)
- **CI/CD** for automating the build of the target files – [GitLab](#) & [redocly](#)
- **Automated tests** (link checker, etc.) – [GitLab](#) & [redocly](#)

GitLab @viesure – system files



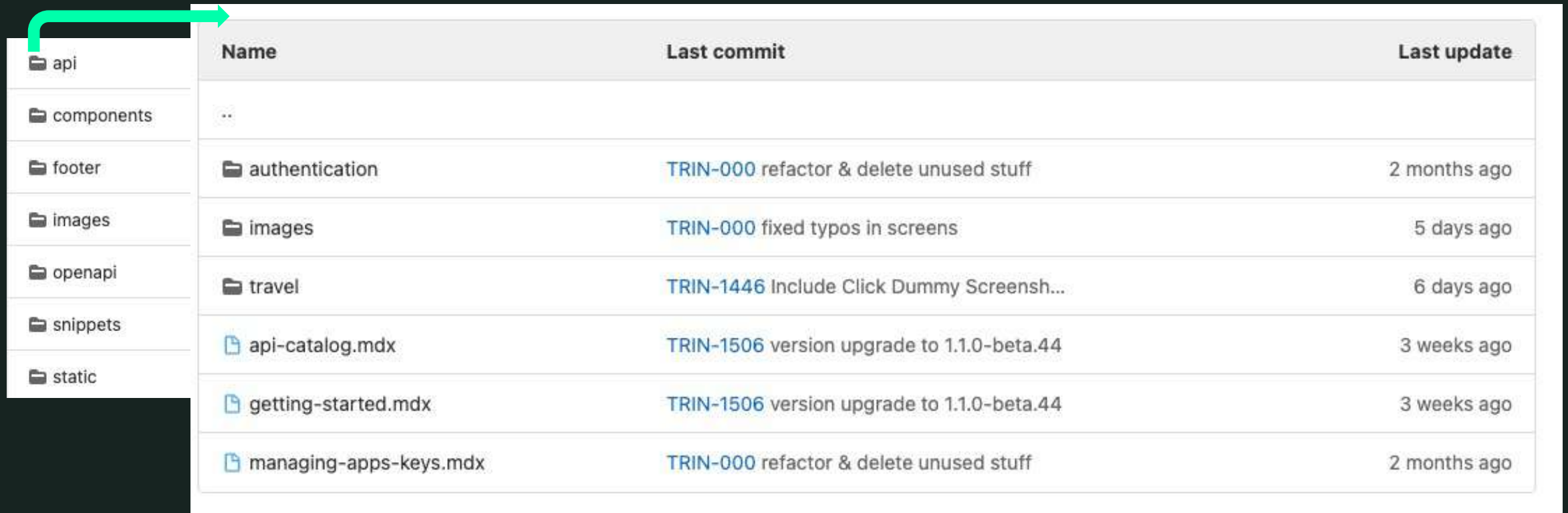
The screenshot shows a GitLab repository interface. The left sidebar contains a file tree with folders like 'api', 'components', 'footer', 'images', 'snippets', 'static', and files like '.editorconfig', '.gitignore', 'LICENSE', 'README.md', 'app.mdx', 'apps.mdx', 'contact.mdx', 'create-app.mdx', 'example_doc.mdx', 'faq.md', 'favicon.png', 'index.mdx', 'package-lock.json', 'package.json', 'sidebars.yaml', 'siteConfig.yaml', 'theme.ts', and 'yarn.lock'. A red arrow points to the 'static' folder. The main content area shows a list of files with their commit details:

Name	Commit	Time
package-lock.json	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago
package.json	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago
sidebars.yaml	TRIN-1446 Include Click Dummy Screensh...	6 days ago
siteConfig.yaml	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago
theme.ts	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago
yarn.lock	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago

Different types of files:

- Portal GUI (**system files**)
- “Handwritten” content
- Automatically generated docs from `openapi.yaml`

GitLab @viesure – additional content



Name	Last commit	Last update
..		
authentication	TRIN-000 refactor & delete unused stuff	2 months ago
images	TRIN-000 fixed typos in screens	5 days ago
travel	TRIN-1446 Include Click Dummy Screensh...	6 days ago
api-catalog.mdx	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago
getting-started.mdx	TRIN-1506 version upgrade to 1.1.0-beta.44	3 weeks ago
managing-apps-keys.mdx	TRIN-000 refactor & delete unused stuff	2 months ago

- **api** – “handwritten” content
- **openapi** – automatically generated docs from `openapi.yaml`

GitLab @viesure – automatic content

The screenshot shows a GitLab repository interface. On the left is a file tree with folders: `api`, `components`, `footer`, `images`, `openapi`, `snippets`, and `static`. On the right is a commit history table with columns `Name` and `Last commit`. A red arrow points from the `images` folder in the file tree to the commit history table.

Name	Last commit
..	
<code>{..} authentication.page.yaml</code>	TRIN-000 try out new options
<code>{..} travel-insurance.page.yaml</code>	TRIN-000 changed definitionId to camelC...

- `api` – “handwritten” docs
- `openapi` – automatically generated docs from `openapi.yaml`

Links

- <https://docs-as-co.de/>
- https://idratherbewriting.com/learnapidoc/pubapis_docs_as_code.html#what-docs-as-code-tools-means
- <https://www.informatik-aktuell.de/entwicklung/methoden/docs-as-code-die-grundlagen.html>
- <https://document360.com/blog/docs-like-code-is-it-worth-the-hassle>
- <https://www.writethedocs.org/>
- <https://redoc.ly/>
- <https://www.heise.de/hintergrund/Documentation-as-Code-mit-Asciidoctor-4642013.html>



Any questions?

#creatingtomorrow



Thank you so much

#creatingtomorrow