Svelte meets JAMstack

Fabian Clemenz

About us

Fabian

- Fullstack Developer @ devsuit

devsuit

- Software agency since 2014
- Custom software development (web apps, apps & more)

Svelte @ devsuit

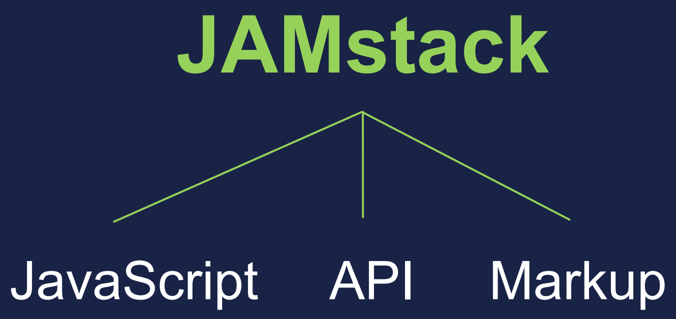
- Used since 2021
- In an internal project with a manageable scope -> *devsuit.de* website
- Brought from *WeAreDevelopers* conference
- "Newcomer Svelte takes the top spot as the most loved framework."[1]

1: 2021 Stackoverflow Developer Survey https://survey.stackoverflow.co/2021#technology

Table of contents

Overview of JAMstack

What is JAMstack?



Important features

- Decoupled
- Pre-rendering

Benefits of JAMstack architecture

Decoupled

✓ Integration using APIs
✓ Better developer experience

Pre-rendering

✓ Better scalability
✓ Better performance

Overall

✓ High security

Svelte and SvelteKit

What is Svelte and SvelteKit

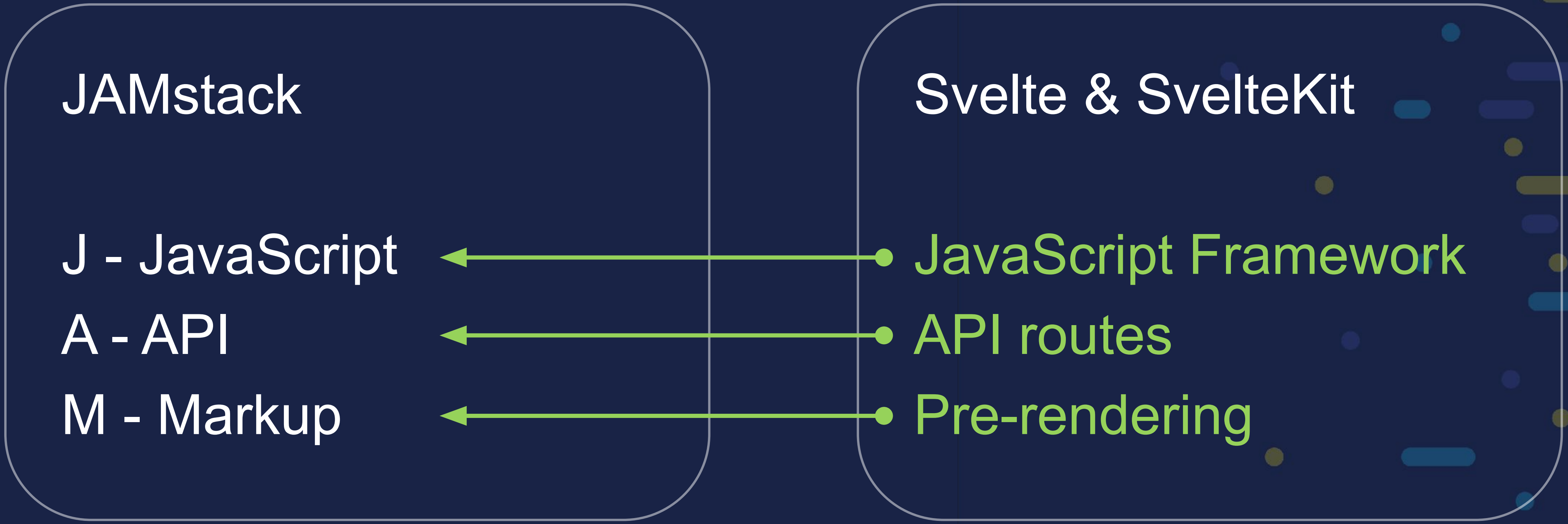
Svelte

- JavaScript Framework
- Compiles at build time

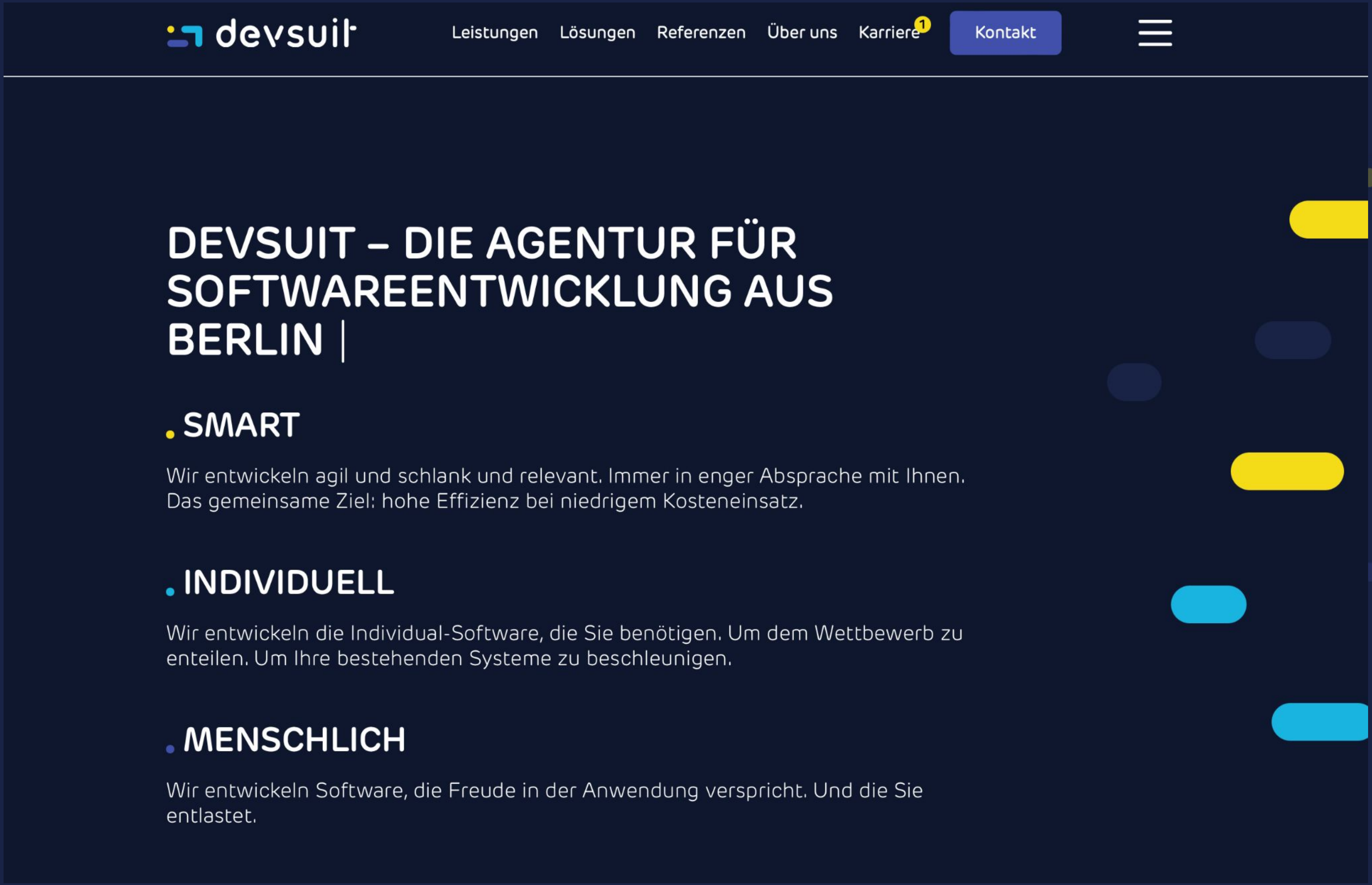
SvelteKit

- Svelte + server-side-rendering
- API routes
- file-based routing
- And more…

How SvelteKit fits into JAMstack



Project implementation



DEVSUIT – DIE AGENTUR FÜR SOFTWAREENTWICKLUNG AUS BERLIN |

SMART

Wir entwickeln agil und schlank und relevant. Immer in enger Absprache mit Ihnen. Das gemeinsame Ziel: hohe Effizienz bei niedrigem Kosteneinsatz.

INDIVIDUELL

Wir entwickeln die Individual-Software, die Sie benötigen. Um dem Wettbewerb zu enteilen. Um Ihre bestehenden Systeme zu beschleunigen.

MENSCHLICH

Wir entwickeln Software, die Freude in der Anwendung verspricht. Und die Sie entlastet.

devsuit.de website project

3 GitHub Repositories

- Frontend $\longrightarrow$ contains source files
- Static $\longrightarrow$ generated static website
- Builder $\longrightarrow$ builds and pushes

More technologies

strapi

tailwindcss

python™

Project implementation

Frontend

Frontend > project structure

src > routes

- Single pages and their corresponding subpages
- +page.svelte, +page.js and [slug]

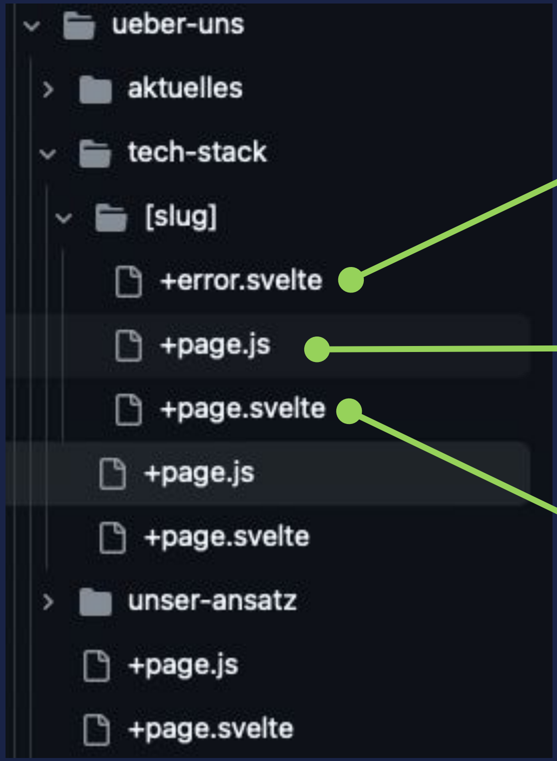
src > lib

- Reusable components: Header, Buttons, Sections, and more…

src > stores.js

- Breadcrumbs and base routes

src > routes



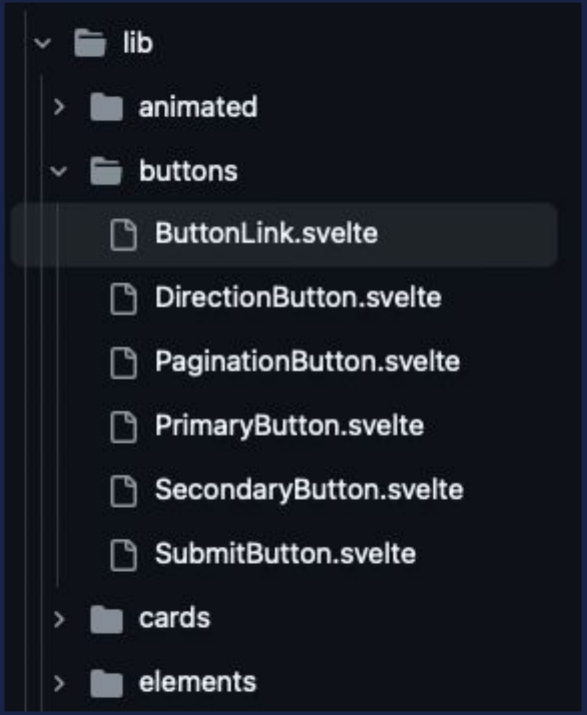
```
ueber-uns
  aktuelles
  tech-stack
    [slug]
      +error.svelte
      +page.js
      +page.svelte
    +page.js
    +page.svelte
  unser-ansatz
  +page.js
  +page.svelte
```

Base template for errors

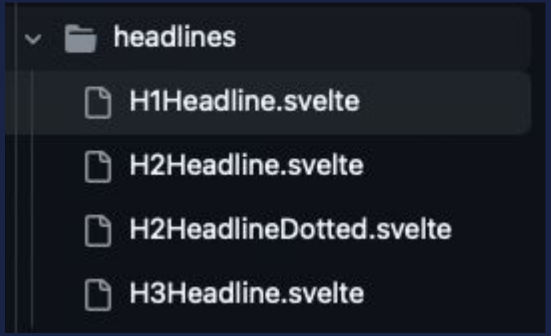
Loads data and entries

Template for the page

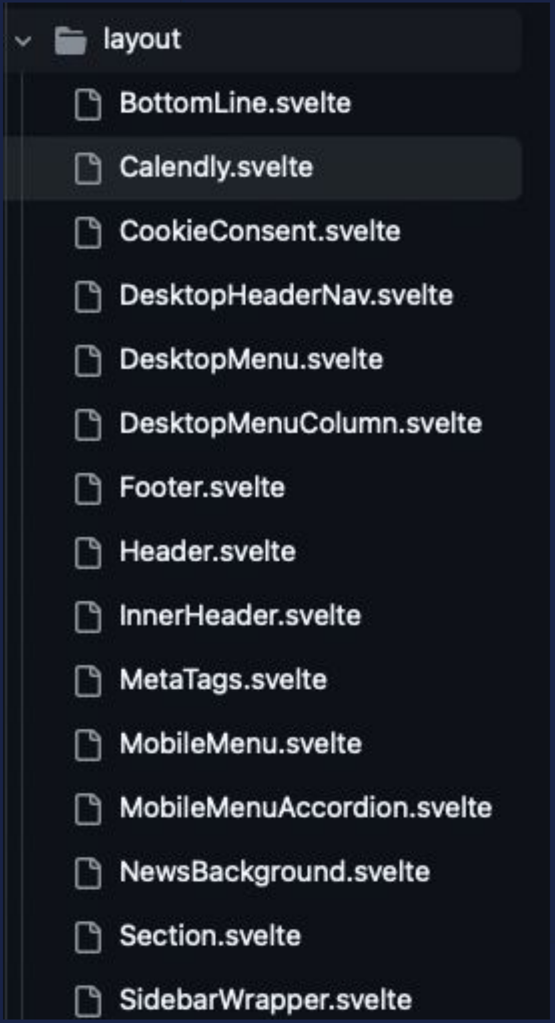
src > lib



```
lib
  animated
  buttons
    ButtonLink.svelte
    DirectionButton.svelte
    PaginationButton.svelte
    PrimaryButton.svelte
    SecondaryButton.svelte
    SubmitButton.svelte
  cards
  elements
```

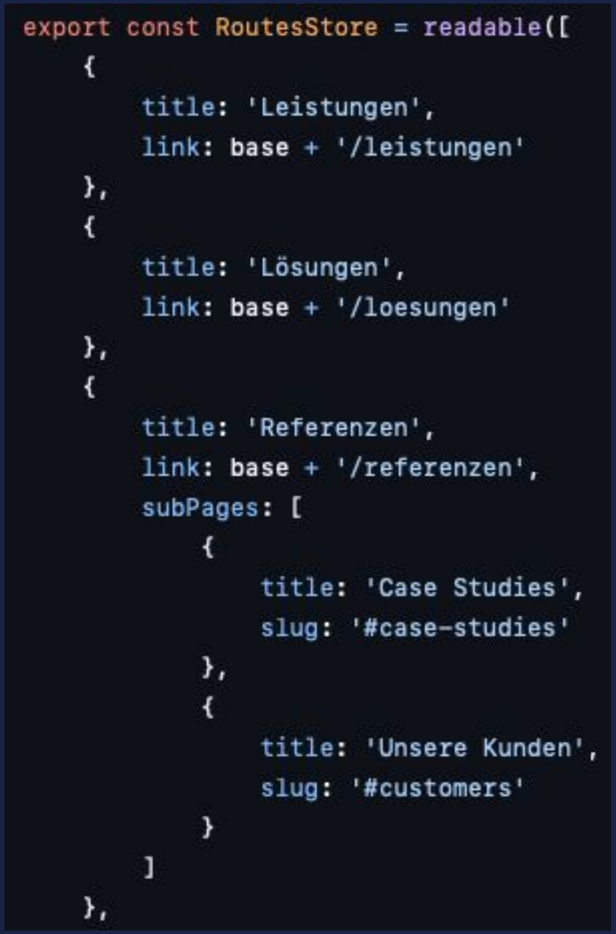


```
headlines
  H1Headline.svelte
  H2Headline.svelte
  H2HeadlineDotted.svelte
  H3Headline.svelte
```

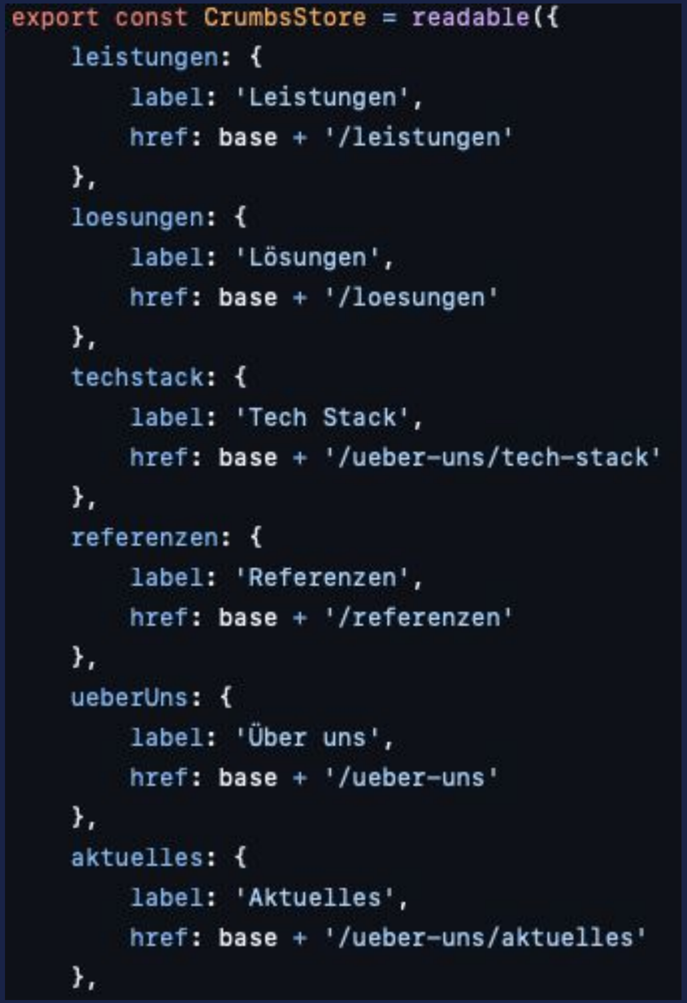


```
layout
  BottomLine.svelte
  Calendly.svelte
  CookieConsent.svelte
  DesktopHeaderNav.svelte
  DesktopMenu.svelte
  DesktopMenuColumn.svelte
  Footer.svelte
  Header.svelte
  InnerHeader.svelte
  MetaTags.svelte
  MobileMenu.svelte
  MobileMenuAccordion.svelte
  NewsBackground.svelte
  Section.svelte
  SidebarWrapper.svelte
```

src > stores.js



```
export const RoutesStore = readable([
    {
        title: 'Leistungen',
        link: base + '/leistungen'
    },
    {
        title: 'Lösungen',
        link: base + '/loesungen'
    },
    {
        title: 'Referenzen',
        link: base + '/referenzen',
        subPages: [
            {
                title: 'Case Studies',
                slug: '#case-studies'
            },
            {
                title: 'Unsere Kunden',
                slug: '#customers'
            }
        ]
    },
```



```
export const CrumbsStore = readable({
    leistungen: {
        label: 'Leistungen',
        href: base + '/leistungen'
    },
    loesungen: {
        label: 'Lösungen',
        href: base + '/loesungen'
    },
    techstack: {
        label: 'Tech Stack',
        href: base + '/ueber-uns/tech-stack'
    },
    referenzen: {
        label: 'Referenzen',
        href: base + '/referenzen'
    },
    ueberUns: {
        label: 'Über uns',
        href: base + '/ueber-uns'
    },
    aktuelles: {
        label: 'Aktuelles',
        href: base + '/ueber-uns/aktuelles'
    },
```

Project implementation

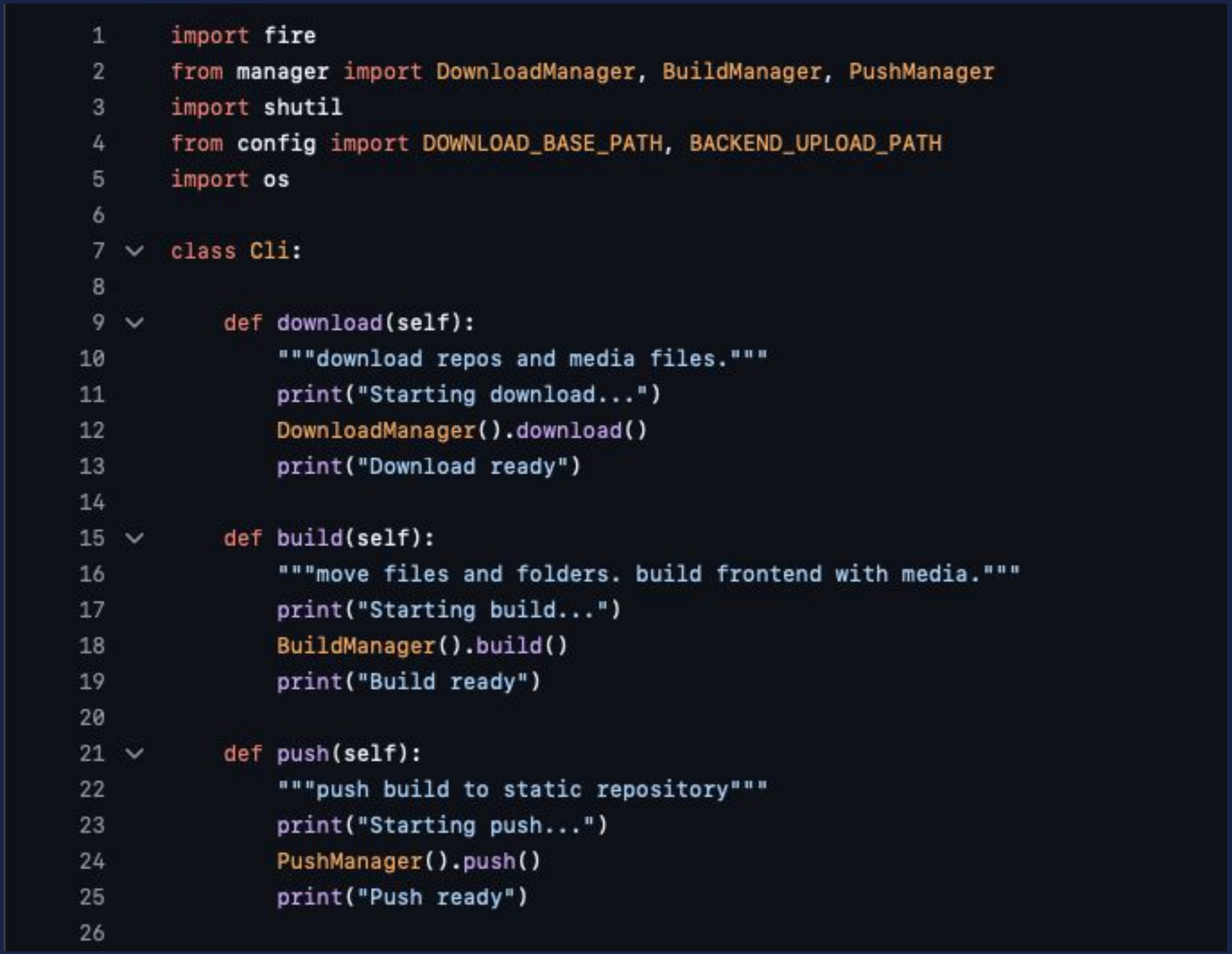
Builder

Builder

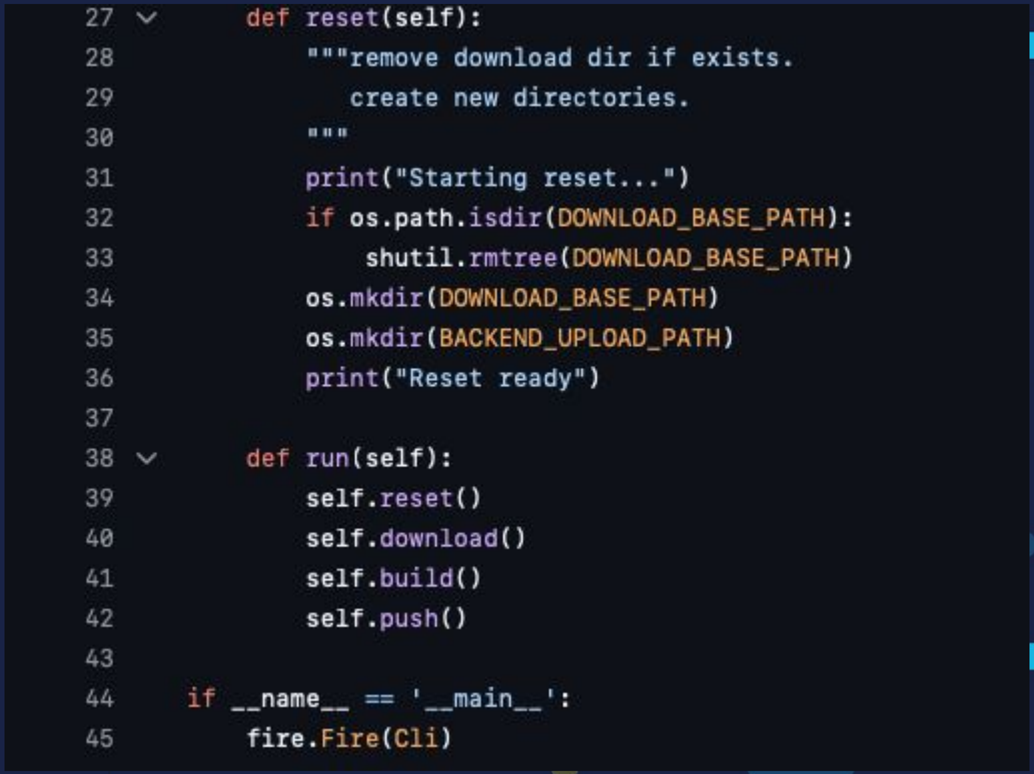
CLI Tool - Main Components

- DownloadManager
- BuildManager
- PushManager

Builder



```python
import fire
from manager import DownloadManager, BuildManager, PushManager
import shutil
from config import DOWNLOAD_BASE_PATH, BACKEND_UPLOAD_PATH
import os

class Cli:

    def download(self):
        """download repos and media files."""
        print("Starting download...")
        DownloadManager().download()
        print("Download ready")

    def build(self):
        """move files and folders. build frontend with media."""
        print("Starting build...")
        BuildManager().build()
        print("Build ready")

    def push(self):
        """push build to static repository"""
        print("Starting push...")
        PushManager().push()
        print("Push ready")

```



```python
    def reset(self):
        """remove download dir if exists.
           create new directories.
        """
        print("Starting reset...")
        if os.path.isdir(DOWNLOAD_BASE_PATH):
            shutil.rmtree(DOWNLOAD_BASE_PATH)
        os.mkdir(DOWNLOAD_BASE_PATH)
        os.mkdir(BACKEND_UPLOAD_PATH)
        print("Reset ready")

    def run(self):
        self.reset()
        self.download()
        self.build()
        self.push()

if __name__ == '__main__':
    fire.Fire(Cli)
```

Lessons learned

Lessons learned

Builder

Builder

The Issue

- Too complex
- Various points of possible failure

What would we change?

- Less custom development $\longrightarrow$ check for prebuilt packages
- Directly use CI tools $\longrightarrow$ GitHub Actions

Lessons learned

Generated static website

Generated static website

The Issue

- Code changes rarely
- Content changes often
- Every small change $\longrightarrow$ needs rebuilding and redeployment

What would we change?

- Dynamic Website

- Increase in loading times is insignificant
- Minimize maintenance

Lessons learned

Some subpages not rendered

Some subpages not rendered

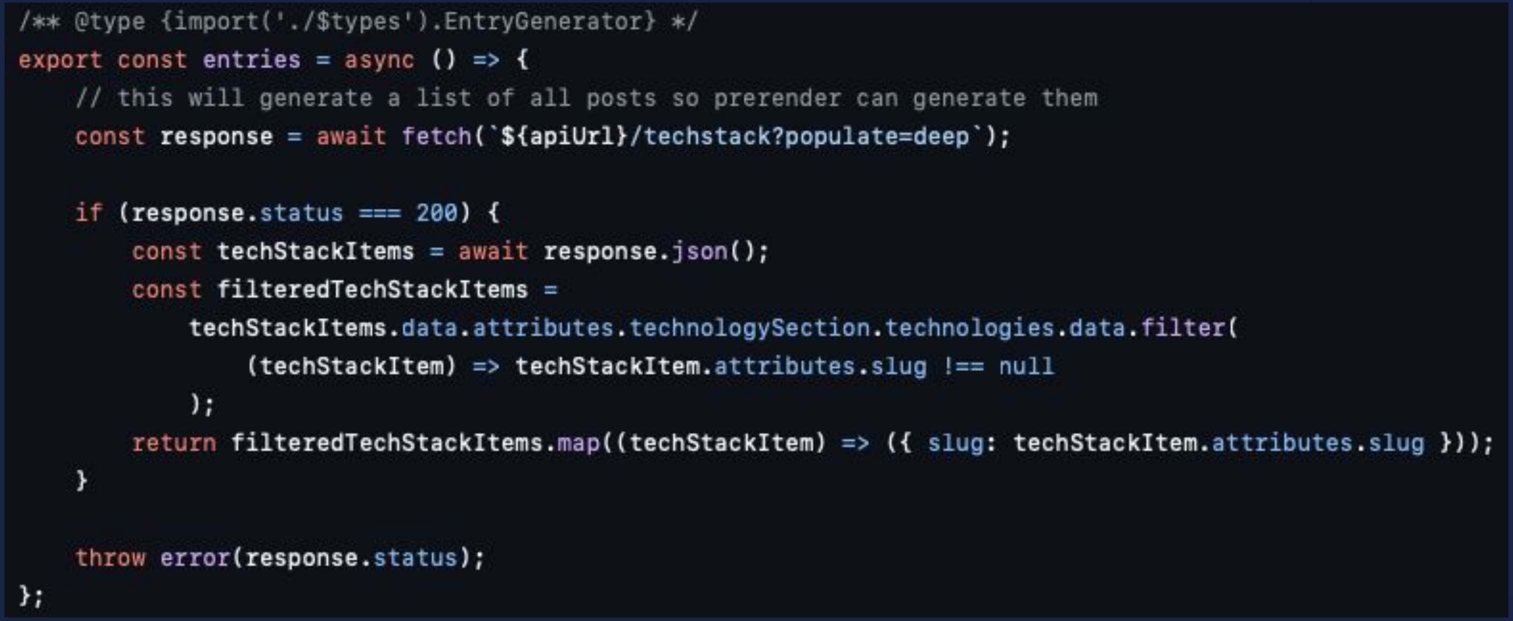
The Issue

- Not all *[slug]* pages retrieved from the backend were built and displayed on the website
- Can lead to multiple 404

Some subpages not rendered

FIX

- using entries function to prefetch all related services



```
/** @type {import('./$types').EntryGenerator} */
export const entries = async () => {
    // this will generate a list of all posts so prerender can generate them
    const response = await fetch(`${apiUrl}/techstack?populate=deep`);

    if (response.status === 200) {
        const techStackItems = await response.json();
        const filteredTechStackItems =
            techStackItems.data.attributes.technologySection.technologies.data.filter(
                (techStackItem) => techStackItem.attributes.slug !== null
            );
        return filteredTechStackItems.map((techStackItem) => ({ slug: techStackItem.attributes.slug }));
    }

    throw error(response.status);
};
```

Conclusion

Conclusion

Advantages

- super fast delivery
- better scalability
- high security

Disadvantages

- needs redeployment on every content change
- complicated setup
- more development resources

Recommendation

- decide on a project-specific basis
- static site generation only suitable for infrequent content changes

Svelte meets JAMstack

Q&A

Thank you for your time and attention. I'm happy to answer any questions you might have.