Kotlog

A static site generator to speed up my content creation

Noob Edition!

Kotlin
Blog

Why?

I‘m a software engineer,

I’m a nerd, I‘m a maker!

If something does not feed my needs, I’ll build it!

Why?

- I wanna have a project that does not remind me at my day job
- I wanna understand what our Kotlin developers code
- I wanna get back to the roots to build what you need
- I wanna start low but have the possibility to add new features
- I wanna have a pet project that is possible to complete

I just wanna have fun! *sing*

What?

!

Simple static blog generator

- Markdown based - no more databases!
- Easy templating using drop-in css files
- Speed up other content creation flows
- Creates social media preview images

Jekyll but in simple and !Ruby

Constraints

- Free of charge hosting on Guthabe (must be static)
- I‘m not a web dev and I do not like it
- No custom JavaScript or CSS files
- No fancy build scripts, transpilers, etc
- Shall be completed in 4 weeks time

How?

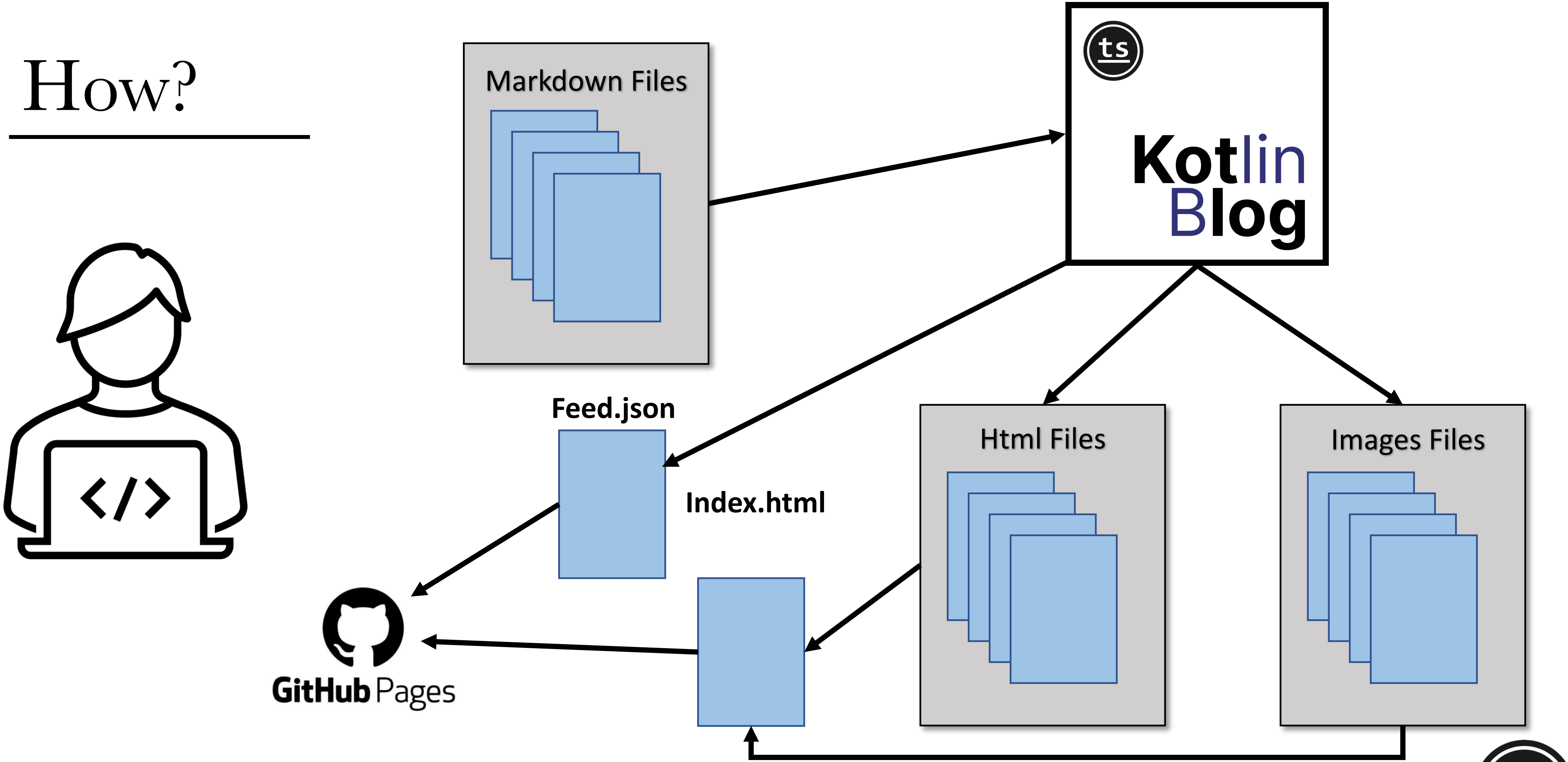
All the way Kotlin!

- Console application

Just a few dependencies

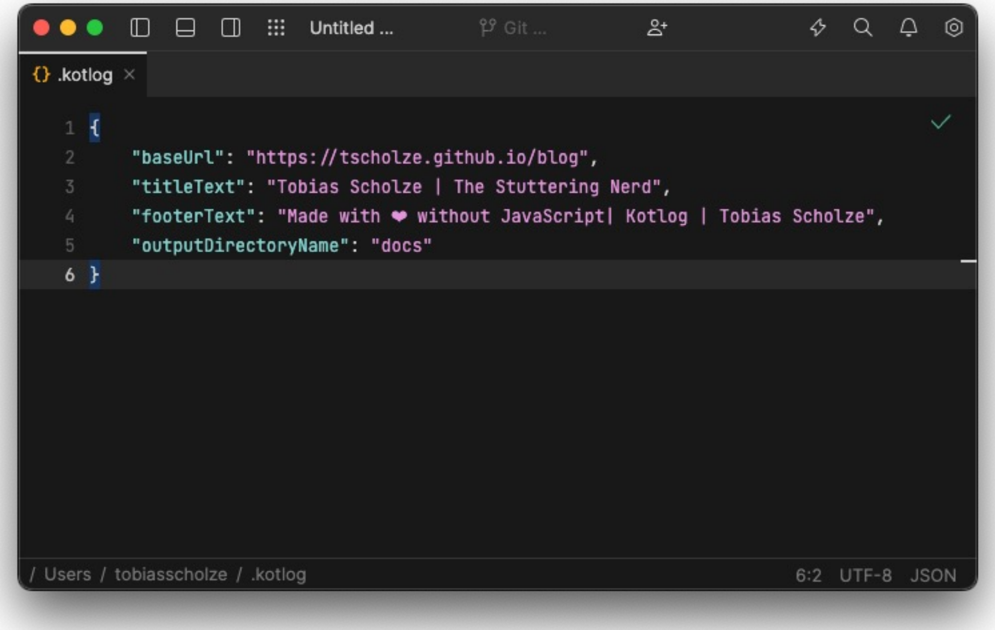
- *org.jetbrains.kotlinx* JSON deserializion
- *org.commonmark:commonmark* Markdown and Frontmatter parser
- *org.jsoup:jsoup* HTML DOM parser
- *com.lordcodes.turtle:turtle* Terminal utility

How?


Markdown Files
ts
Kotlin
Blog
Feed.json
Index.html
Html Files
Images Files
GitHub Pages

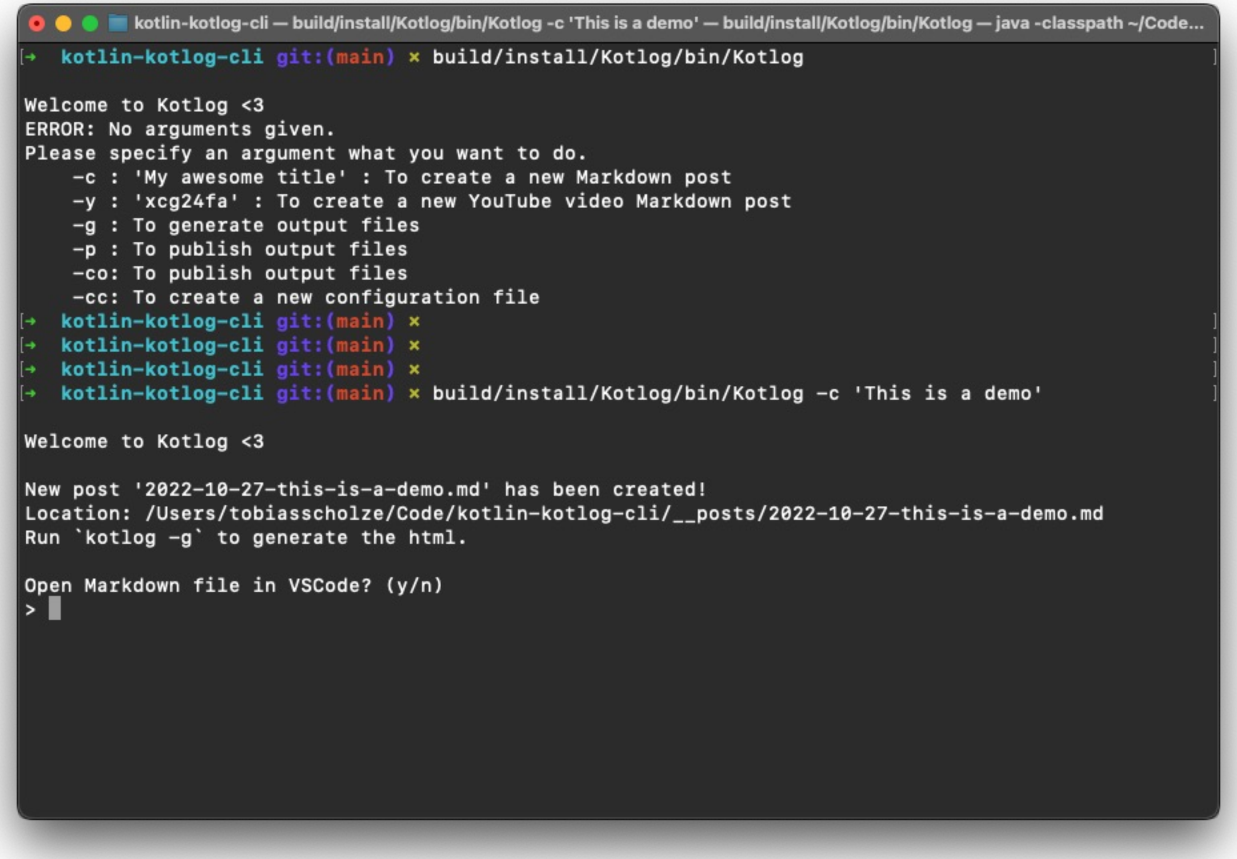
Result

~/.kotlog | Config file



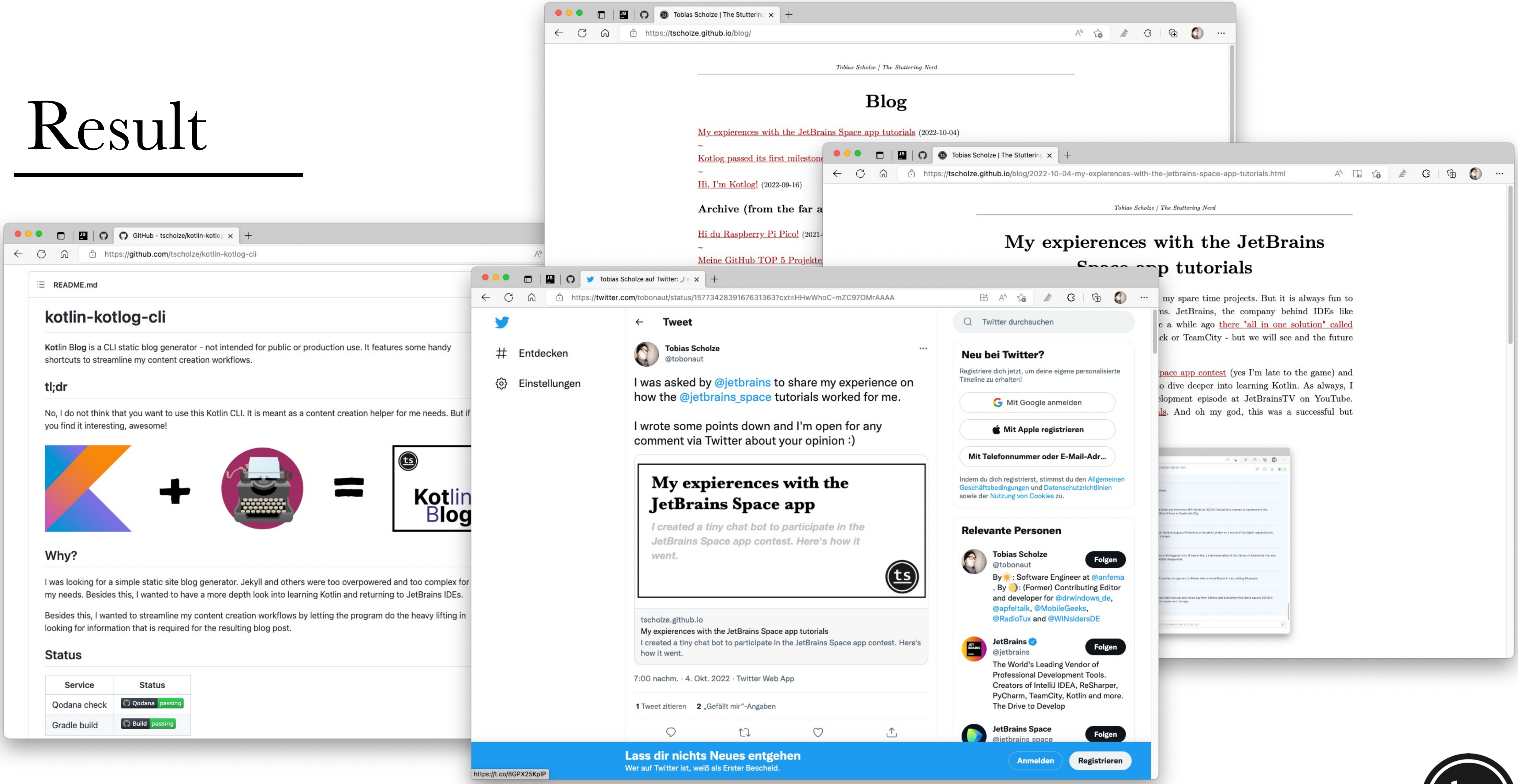
```json
{
    "baseUrl": "https://tscholze.github.io/blog",
    "titleText": "Tobias Scholze | The Stuttering Nerd",
    "footerText": "Made with ❤ without JavaScript| Kotlog | Tobias Scholze",
    "outputDirectoryName": "docs"
}
```

Simple CLI | Terminal


kotlin-kotlog-cli — build/install/Kotlog/bin/Kotlog -c 'This is a demo' — build/install/Kotlog/bin/Kotlog — java -classpath ~/Code...

```
→  kotlin-kotlog-cli git:(main) ✗ build/install/Kotlog/bin/Kotlog

Welcome to Kotlog <3
ERROR: No arguments given.
Please specify an argument what you want to do.
    -c : 'My awesome title' : To create a new Markdown post
    -y : 'xcg24fa' : To create a new YouTube video Markdown post
    -g : To generate output files
    -p : To publish output files
    -co: To publish output files
    -cc: To create a new configuration file
→  kotlin-kotlog-cli git:(main) ✗
→  kotlin-kotlog-cli git:(main) ✗
→  kotlin-kotlog-cli git:(main) ✗
→  kotlin-kotlog-cli git:(main) ✗ build/install/Kotlog/bin/Kotlog -c 'This is a demo'

Welcome to Kotlog <3

New post '2022-10-27-this-is-a-demo.md' has been created!
Location: /Users/tobiasscholze/Code/kotlin-kotlog-cli/__posts/2022-10-27-this-is-a-demo.md
Run `kotlog -g` to generate the html.

Open Markdown file in VSCode? (y/n)
>
```

Result



Tobias Scholze | The Stuttering Nerd
https://tscholze.github.io/blog/

Tobias Scholze / The Stuttering Nerd

Blog

My expierences with the JetBrains Space app tutorials (2022-10-04)

~

Kotlog passed its first milestone

~

Hi, I'm Kotlog! (2022-09-16)

Archive (from the far a

Hi du Raspberry Pi Pico! (2021-

~

Meine GitHub TOP 5 Projekte

Tobias Scholze | The Stuttering
https://tscholze.github.io/blog/2022-10-04-my-expierences-with-the-jetbrains-space-app-tutorials.html

Tobias Scholze / The Stuttering Nerd

My expierences with the JetBrains Space app tutorials

my spare time projects. But it is always fun to
ms. JetBrains, the company behind IDEs like
e a while ago there "all in one solution" called
ck or TeamCity - but we will see and the future

pace app contest (yes I'm late to the game) and
o dive deeper into learning Kotlin. As always, I
elopment episode at JetBrainsTV on YouTube.
ls. And oh my god, this was a successful but

GitHub - tscholze/kotlin-kotlog
https://github.com/tscholze/kotlin-kotlog-cli
README.md

kotlin-kotlog-cli

Kotlin Blog is a CLI static blog generator - not intended for public or production use. It features some handy shortcuts to streamline my content creation workflows.

tl;dr

No, I do not think that you want to use this Kotlin CLI. It is meant as a content creation helper for me needs. But if you find it interesting, awesome!

Why?

I was looking for a simple static site blog generator. Jekyll and others were too overpowered and too complex for my needs. Besides this, I wanted to have a more depth look into learning Kotlin and returning to JetBrains IDEs.

Besides this, I wanted to streamline my content creation workflows by letting the program do the heavy lifting in looking for information that is required for the resulting blog post.

Status

Service	Status
Qodana check	Qodana passing
Gradle build	Build passing

Tobias Scholze auf Twitter: „I w
https://twitter.com/tobonaut/status/1577342839167631363?cxt=HHwWhoC-mZC97OMrAAAA
Entdecken
Einstellungen
Tweet
Twitter durchsuchen

Tobias Scholze
@tobonaut

I was asked by @jetbrains to share my experience on how the @jetbrains_space tutorials worked for me.

I wrote some points down and I'm open for any comment via Twitter about your opinion :)

My expierences with the JetBrains Space app

I created a tiny chat bot to participate in the JetBrains Space app contest. Here's how it went.

tscholze.github.io
My expierences with the JetBrains Space app tutorials
I created a tiny chat bot to participate in the JetBrains Space app contest. Here's how it went.

7:00 nachm. · 4. Okt. 2022 · Twitter Web App

1 Tweet zitieren 2 „Gefällt mir"-Angaben

Neu bei Twitter?

Registriere dich jetzt, um deine eigene personalisierte Timeline zu erhalten!

Indem du dich registrierst, stimmst du den Allgemeinen Geschäftsbedingungen und Datenschutzrichtlinien sowie der Nutzung von Cookies zu.

Relevante Personen

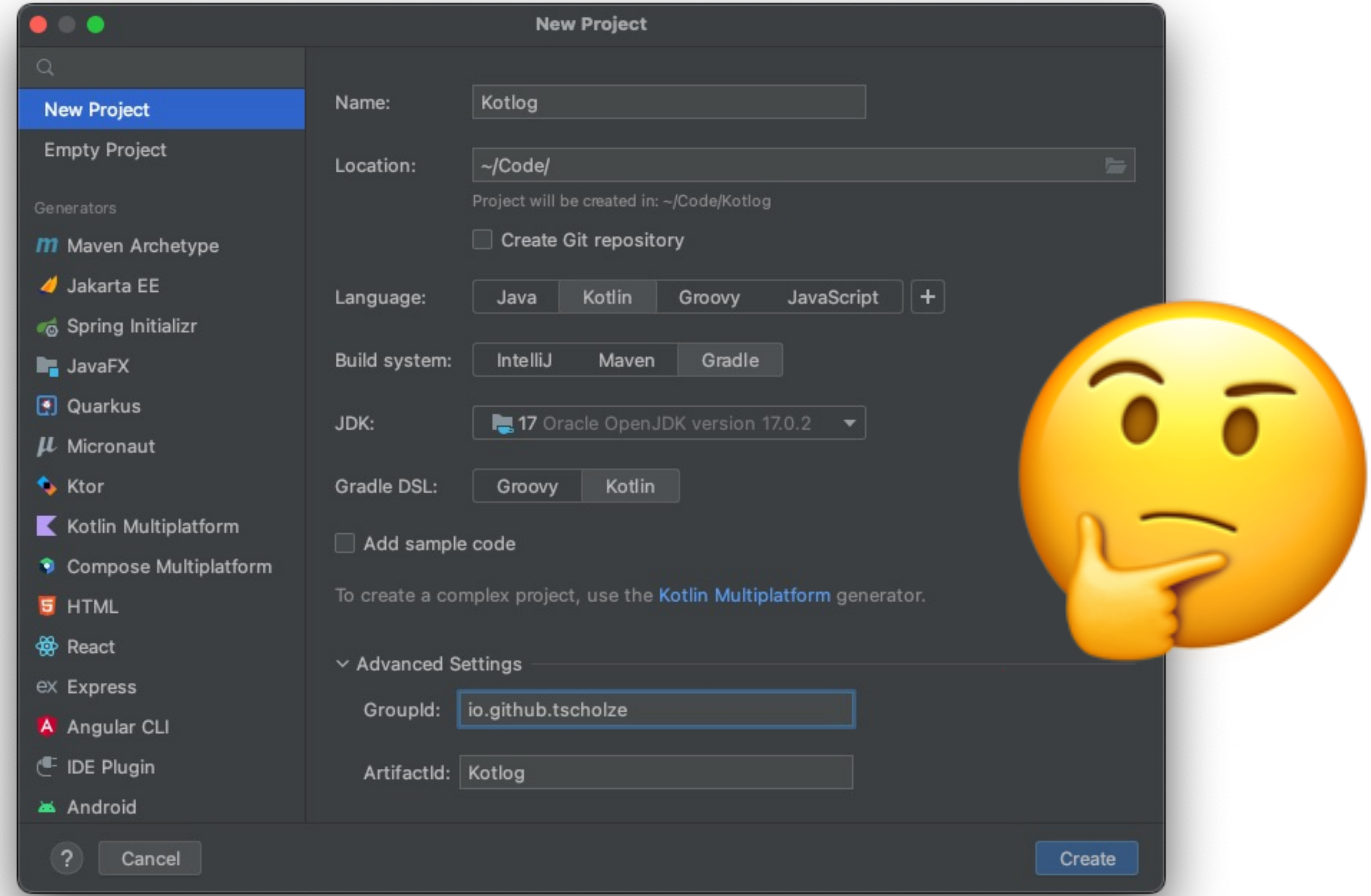
Tobias Scholze
@tobonaut
By☀: Software Engineer at @anfema, By🌖: (Former) Contributing Editor and developer for @drwindows_de, @apfeltalk, @MobileGeeks, @RadioTux and @WINsidersDE

JetBrains
@jetbrains
The World's Leading Vendor of Professional Development Tools. Creators of IntelliJ IDEA, ReSharper, PyCharm, TeamCity, Kotlin and more. The Drive to Develop

JetBrains Space
@jetbrains_space

Lass dir nichts Neues entgehen
Wer auf Twitter ist, weiß als Erster Bescheid.

A month earlier ...



Name: Kotlog

Location: ~/Code/

Project will be created in: ~/Code/Kotlog

☐ Create Git repository

Language: Java | **Kotlin** | Groovy | JavaScript | +

Build system: IntelliJ | Maven | **Gradle**

JDK: 17 Oracle OpenJDK version 17.0.2

Gradle DSL: Groovy | **Kotlin**

☐ Add sample code

To create a complex project, use the Kotlin Multiplatform generator.

Advanced Settings

GroupId: io.github.tscholze

ArtifactId: Kotlog

In detail

Everything is built on gaint‘s shoulders

Aka Dependencies

In detail

build.gradle.kts

```
dependencies {
  // Markdown
  implementation("org.commonmark:commonmark")
  implementation("org.commonmark:commonmark-ext-yaml-front-matter")
  implementation("org.commonmark:commonmark-ext-gfm-tables")
  implementation("org.commonmark:commonmark-ext-autolink")

  // Html
  implementation("org.jsoup:jsoup")

  // Json
  implementation("org.jetbrains.kotlinx:kotlinx-serialization-json")

  // CLI
  implementation("org.jetbrains.kotlinx:kotlinx-cli")
  implementation("com.lordcodes.turtle:turtle")
}
```

Markdown

2022-10-04-my-expierences-with-the-jetbrains-space-app-tutorials.md

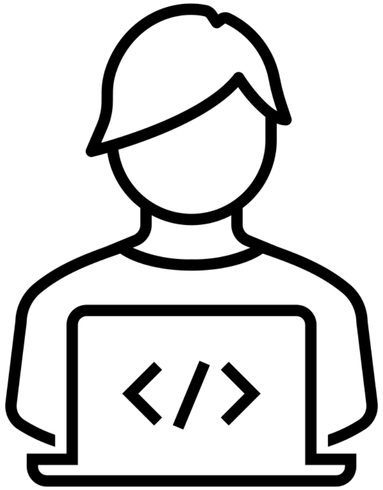
```
---
title: 'My expierences with the JetBrains Space ...'
abstract: 'I created a tiny chat bot ...‘
date: '2022–10–04'
---

I use GitHub during my day job and also for my spare time
projects. But it is always fun to play around with other
developer ecosystems.
```

Markdown

Generator.kt

```kotlin
fun fromFile(file: File): PostConfiguration {
    val node = parser.parse(file.readText())

   // Check for front matter data
   val fm = YamlFrontMatterVisitor()
   node.accept(fm)

   // Get simple objects
   val title = fm.data["title"]?.first()

  // Get complex objects
  val date = LocalDate.parse(ds, DATE_FORMATTER)
  val tags = fm.data["tags"] ?: listOf("none")

  // Generate inner html string
  val html = HtmlRenderer.builder().build().render(node)

  // Return created configuration
  return PostConfiguration(...)
}
```

SMP Images

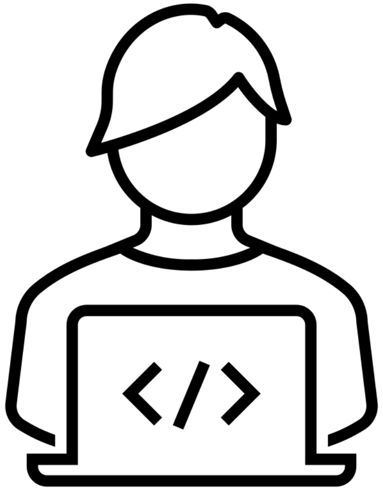


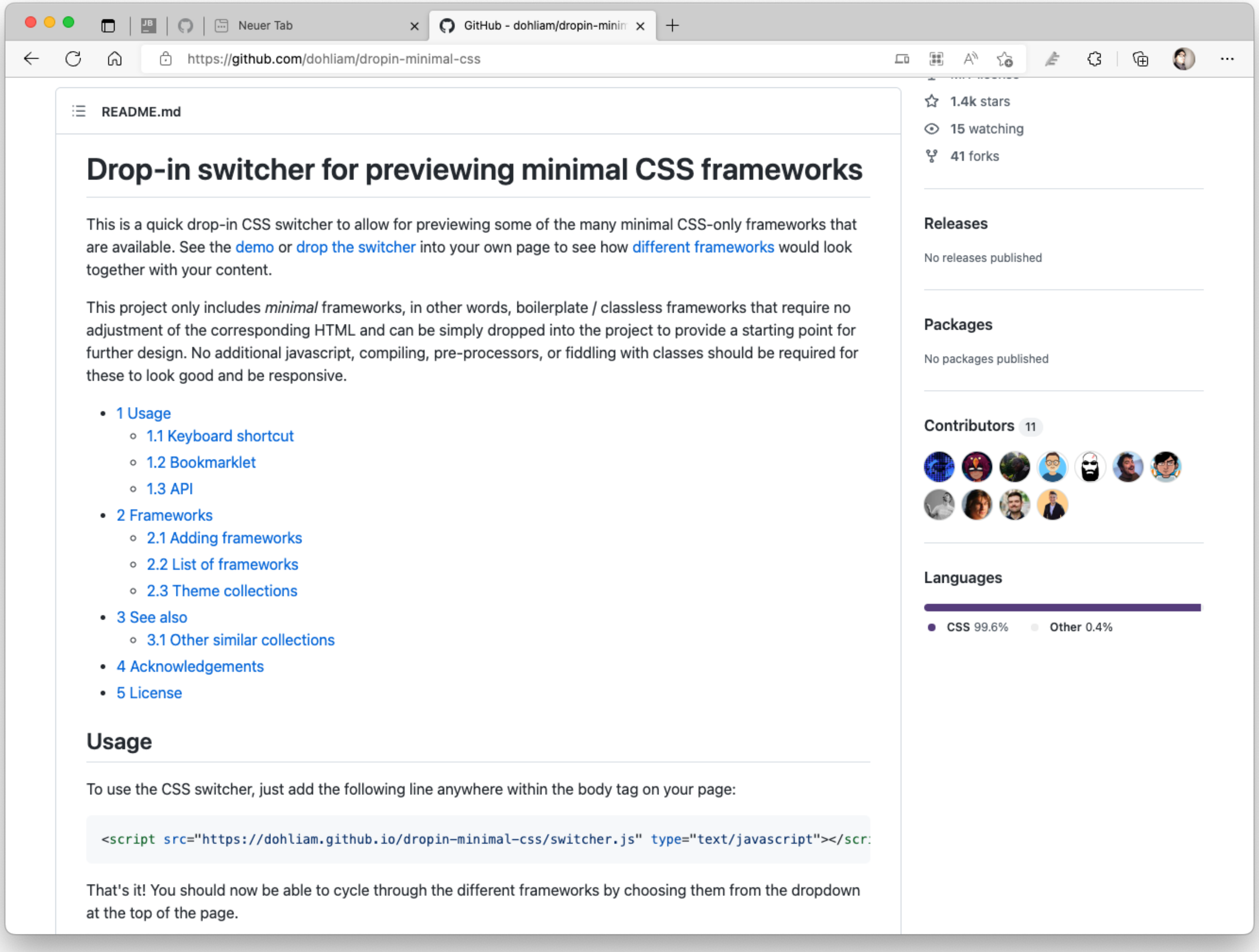
```kotlin
fun generate(pc: PostConfiguration) {
  // Create canvas
  val image = BufferedImage(...)

  // Create image with background
  val graphics = image.createGraphics()
  graphics.drawImage(BACKGROUND_IMAGE, 0, 0, null)

  // Add title
  graphics.font = TITLE_FONT
  graphics.color = TITLE_COLOR
  val titleLines = wrap(pc.title,graphics.fontMetrics)

  for ((index, value) in titleLines.withIndex()) {
    graphics.drawString(...)
  }

...
```

SMP Images

```
  ...

  // Add abstract
  graphics.font = ABSTRACT_FONT
  graphics.color = ABSTRACT_COLOR

  val abstractLines = wrap(
    postConfiguration.abstract,
    graphics.fontMetrics
  ).take(3)

  for ((index, value) in abstractLines.withIndex()) {
    graphics.drawString(
      value,
      100,
      140 + (index * 60) + (titleLines.count() * 88))
  }

  // Write file
  ImageIO.write(...)
}
```

Styling




Neuer Tab
GitHub - dohliam/dropin-minin
https://github.com/dohliam/dropin-minimal-css
README.md

Drop-in switcher for previewing minimal CSS frameworks

This is a quick drop-in CSS switcher to allow for previewing some of the many minimal CSS-only frameworks that are available. See the demo or drop the switcher into your own page to see how different frameworks would look together with your content.

This project only includes *minimal* frameworks, in other words, boilerplate / classless frameworks that require no adjustment of the corresponding HTML and can be simply dropped into the project to provide a starting point for further design. No additional javascript, compiling, pre-processors, or fiddling with classes should be required for these to look good and be responsive.

Usage

To use the CSS switcher, just add the following line anywhere within the body tag on your page:

```
<script src="https://dohliam.github.io/dropin-minimal-css/switcher.js" type="text/javascript"></scr
```

That's it! You should now be able to cycle through the different frameworks by choosing them from the dropdown at the top of the page.

1.4k stars

15 watching

41 forks

Releases

No releases published

Packages

No packages published

Contributors 11

Languages

● CSS 99.6% ● Other 0.4%

Most important!

Without mentors and friends, I would not got this far

Marcel for getting me started with Kotlin

Dave Herman for having always an open ear for me

Klaus for mentioning the Drop-In-CSS files

Caution
Bugs, Bugs, Bugs

Links

Kotlog

- https://github.com/tscholze/kotlin-kotlog-cli
- https://github.com/tscholze/kotlin-kotlog-template

Usage

- https://github.com/tscholze/blog

Discord tip

- https://discord.gg/5NZ2GKV5Cs

- The End -

Any questions?