BrunoBernardino /
deno-boilerplate-simple-website

<> **Code** ⊙ Issues Pull requests Actions Projects Security Insights

Use this template ▾

This is a simple framework-less boilerplate for a Deno website, deployed with Deno Deploy. 🦕

simple-deno-website-boilerplate.onbrn.com/

MIT license

☆ **19** stars **5** forks **4** watching Branches Activity
Tags

Public template repository

66M

1 Branch **0** Tags Go to file t Go to file + Add file ▾ <> Code ▾ ···

BrunoBernardino Remove outdated/unmaintained frameworks ✓

8db68f2 · 2 months ago

.github	Update Deno to 2.1 LTS!	2 months ago
components	Prettify import for react co...	2 years ago
lib	Update Deno + dependenc...	3 months ago
pages	Update Deno to 2.1 LTS!	2 months ago
public	Update Deno to 2.1 LTS!	2 months ago
.dvmrc	Update Deno to 2.1 LTS!	2 months ago
LICENSE	Add web components with ...	3 years ago
Makefile	Update Deno + dependenc...	3 months ago
README.md	Remove outdated/unmaint...	2 months ago
deno.json	Update Deno to 2.1 LTS!	2 months ago
main.ts	Add custom 404 page.	2 years ago
main_test.ts	Update Deno + dependenc...	3 months ago
routes.ts	Update Deno to 2.1 LTS!	2 months ago

README MIT license

Simple Deno Website Boilerplate

Run Tests passing

This is a simple website boilerplate built using Deno and deployed using Deno Deploy.

Demo at simple-deno-website-boilerplate.onbrn.com.

Framework-less

This right here is vanilla TypeScript and JavaScript. It's very easy to update and maintain.

It's meant to have no extra dependencies, builders, packagers, bundlers, or pre/post-processors. Just vanilla stuff.

It does include some examples for building complex apps, like:

- No JS form submission in `/form`
- Dynamic client-side updates in `/dynamic` with vanilla JS
- Hydration in `/ssr` (when some JS is dynamically created to update the client state once it's finished the initial load) with vanilla JS
- Using Just-In-Time-transpiled-TypeScript (inspired by ts-serve) Web Components for dynamic client-side updates in `/web-component` (it doesn't work on Safari)
- Using Just-In-Time-transpiled-SASS for `scss` → `css` output
- React with SSR and hydration in `/react` (and `/react/5`, for example)
- Also, if you want to see things like Google Sign-in or GitHub Sign-in, running PostgreSQL database migrations, filling PDFs, sending emails with attachments, dynamic sitemaps, or uploading files to AWS S3, I've also published a Deno Code Examples repo.

You can build pretty complex and complicated things without frameworks or loads of dependencies, but if you need a framework, I'd suggest you try fresh to still be able to enjoy a lot of Deno. It's pretty nice.

Requirements

This was tested with `deno`'s version stated in the `.dvmrc` file, though it's possible other versions might work.

There are no other dependencies. **Deno**!

Development

```
$ make start
$ make format
$ make test
```

Structure

> Backend routes are defined at `routes.ts`.
>
> Static files are defined at `public/`.
>
> Pages are defined at `pages/`.

Everything else can be structured differently.

Deployment

> Deno Deploy: Just push to the `main` branch. Any other branch will create a preview deployment.

TODOs

Here are some things you will likely want to change before "publishing" this, or after cloning it:

- ☐ `baseUrl`, `defaultTitle`, `defaultDescription`, and `content-security-policy` in `lib/utils.ts`
- ☐ Title, description, and links in this `README.md` file
- ☐ `robots.txt` and `sitemap.xml` files

Sponsor this project

BrunoBernardino Bruno Bernardino

https://paypal.me/brunobernardino

https://gist.github.com/BrunoBernardino/ff5b54c13dd96ac7f9fee6fbfd825b09

Learn more about GitHub Sponsors

Contributors 2

BrunoBernardino Bruno Bernardino

cdoremus Craig Doremus

Deployments 41

✓ **Production** 2 months ago

✓ **Preview** 3 years ago

+ 39 deployments

Languages

● **TypeScript** 71.8% ● **SCSS** 16.4% ● **CSS** 6.3% ● **JavaScript** 4.8% ● **Makefile** 0.7%