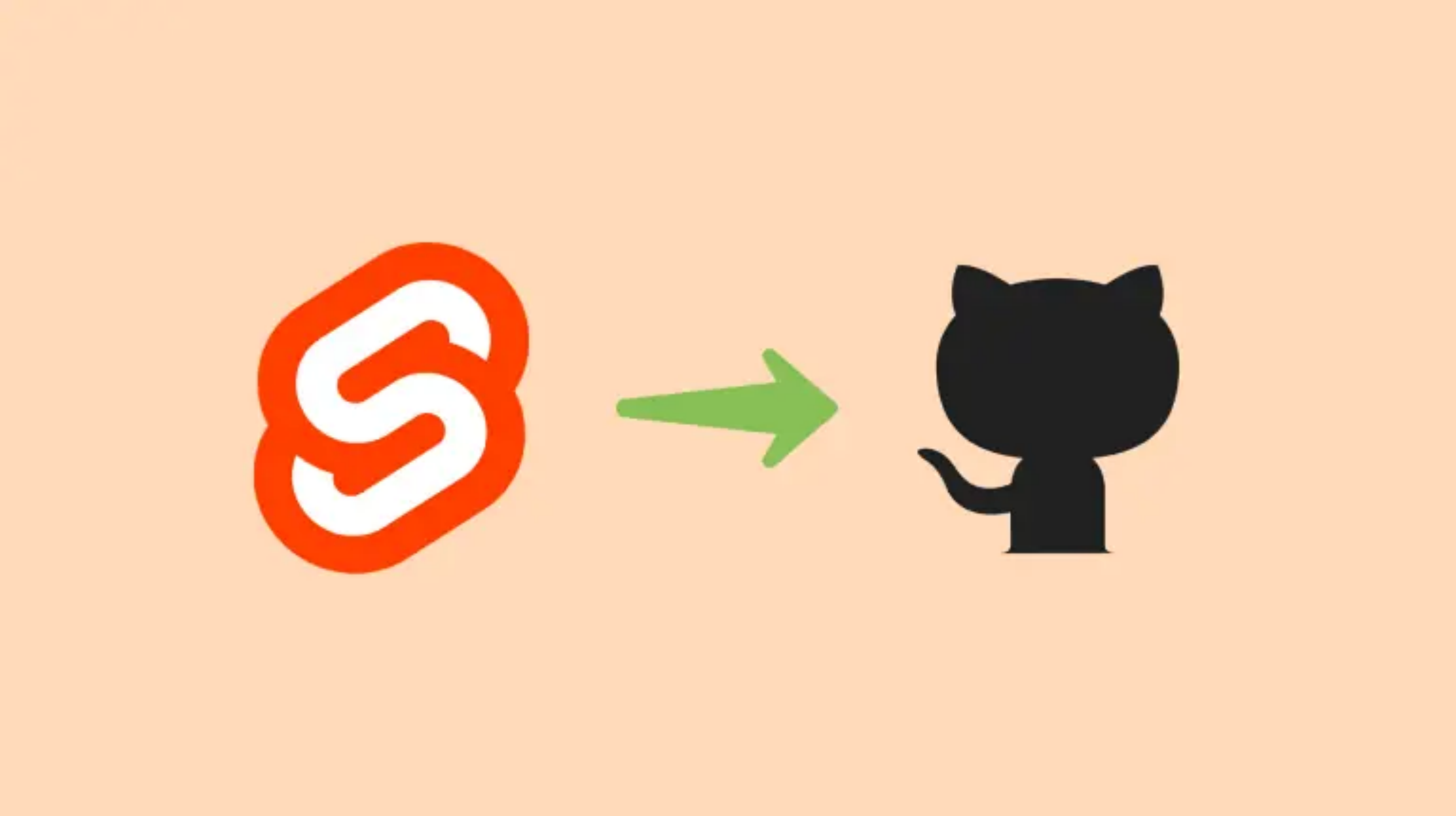
Deploy svelte app in Github Pages

svelte web



Showcasing your new project is always special. If you are building something for months and after its completion finally you can show the project to the masses. They can give feedback and inspires you to do more.

Let's assume you are making a tool for other developers. Instead of writing paragraph after paragraph in your documentation, you can just provide a live link to your project.

Having a live demo of your projects is important for your potential recruiters. Instead of showing the code, just show them the live preview. It will boost your probability to get shortlisted.

Getting a freelancing gig is also very easier for you if you have a portfolio with a live showing of your projects.

What are GitHub pages

Github Pages is a static site hosting service for your projects. You just have to provide static files from your project and Github pages will do the rest for you.

You can also add your custom domain with SSL to make your site more professional looking.

Before GitHub pages, we need to use another service like Netlify or vercel. We have to hook that service with our GitHub or other online git repositories (Gitlab or Bitbucket etc.). Then at every push, they build and deploy a new version of our site automatically.

But these features are not free. After the trial ends, you have to pay some money to continue using their services. It is sensible for a business but indie developers like you and me, it is not what we want to go after.

While on the other hand, GitHub pages are always free for hobby projects. If you want to do some serious task and attract millions of viewers, then Github pages is probably not for you.

What to expect in this article

In this article, we are going to build and deploy a svelte app on Github pages.

When I was new to svelte and wanted to deploy my app, I got some errors. It became a pain for me. So, I researched a little and find a solution. Today I am here to share this solution with you.

Build a svelte app

To build your svelte app, go to the svelte homepage. Start a new project by following the commands. In your terminal, run

```
npm init vite my-app --template svelte
cd my-app
npm install
npm run dev

```

Now you have a svelte app in the `my-app` directory. You can also give some other names to your project.

Open your browser and go to `localhost:3000` . There you can find your newly created svelte app.

Now create a Github repository. Let's named your repository as `my-app` . Now upload your project to the newly created Github repo.

If you don't know how to upload your project to the Github repository, then follow these commands.

```
git add .
git commit -m"First commit"
git branch -M main
git remote add origin https://www.github.com/<username>/<reponame>.
git push -u origin main

```

In the above command replace `<username>` and `<reponame>` with your original username and repository name.

deploy it in Github pages

Before deploying our project, we have to install a dev-dependency called gh-pages in our project.

This `gh-pages` package takes our build directory. Then make a new remote branch called `gh-pages` in the remote GitHub repository. Then it pushes the build directory to the newly created remote `gh-pages` branch.

In your project root, run

```
npm install --save-dev gh-pages

```

Now open your `vite.config.js` file present in your project root. Add a new field called `base` and enter your repository name as a value with trailing `/` .

For example, if your repository name is `my-app` , then you should add `base: "/my-app/"` inside your vite config file. The example looks like this.

```
import { defineConfig } from "vite";
import { svelte } from "@sveltejs/vite-plugin-svelte";

// https://vitejs.dev/config/
export default defineConfig({
  base: "/my-app/",
  plugins: [svelte()],
});

```

Now add a new script called `deploy` inside your `package.json` file for `gh-pages` package.

```
"scripts": {
  "dev": "vite",
  "build": "vite build",
  "preview": "vite preview",
  "deploy": "npx gh-pages -d dist"
},
```

When you run the deploy script, gh-pages automatically take out `dist/` directory and deploy it in Github pages.

Now after all this setup, building and deploying is become very easy. Go to your project root and simply run these two commands.

```
npm run build
npm run deploy
```

After some minutes, when the github build of your project is completed, just visit this `https://<username>.github.io/<repository name>` URL. Don't forget to change the `username` and `repository name` to your personal Github username and repository.

conclusion

I think now you can easily deploy your svelte project to Github pages. If you like this article do share it with your friends. If you have any questions, I am available on Twitter as @hrishikshpathak.

Do share your latest project with me. I will be very glad to see your project. Until then, have a nice day.

PAGES

Home

EXTRA

Newsletter

SOCIAL

About	Resume
Blog	Snippets