

SvelteKit Full Stack Project Structure

Complete app with form actions, server routes, and database. Full-stack TypeScript.

#sveltekit #svelte #typescript #prisma #fullstack

PNG

PDF

Copy

</> Prompt

Project Directory

my-app/

>

src/

app.html

app.css

app.d.tsApp-level types

hooks.server.tsServer hooks (a...

lib/

index.ts

server/Server-only code

db.tsPrisma client

auth.tsAuth utilities

components/

Button.svelte

Input.svelte

Card.svelte

utils/

validation.tsZod schemas

routes/

+layout.svelte

+layout.server.tsRoot load funct...

+page.svelte

auth/Route group for...

+layout.svelteAuth layout

login/

+page.svelte

+page.server.tsForm action

register/

+page.svelte

+page.server.ts

app/Protected routes

+layout.svelte

+layout.server.tsAuth guard

dashboard/

+page.svelte

+page.server.ts

settings/

+page.svelte

+page.server.ts

api/API endpoints

users/

+server.tsGET/POST

[id/

+server.ts

prisma/Database layer

schema.prisma

migrations/

static/

favicon.png

svelte.config.js

vite.config.ts

package.json

tsconfig.json

.envDATABASE_URL, s...

.env.example

.gitignore

Why This Structure?

This structure combines SvelteKit's form actions with Prisma for a complete full-stack setup. Route groups separate auth pages from protected app routes. Server-only code lives in `$lib/server/` and is never bundled for the client.

Key Directories

src/lib/server/ - Server-only utilities—never sent to client
src/routes/(app)/ - Protected routes with auth guard in layout
src/routes/api/ - REST API endpoints using `+server.ts`
prisma/ - Database schema and migrations
hooks.server.ts - Handle auth on every request

Form Actions

+page.server.ts - Define `actions` object for form handling
use:enhance - Progressive enhancement for forms
ActionData - Type-safe form response data
fail() - Return validation errors to the form

Getting Started

```
npm create svelte@latest my-app
npm install prisma @prisma/client
npx prisma init
# Edit prisma/schema.prisma
npx prisma migrate dev
npm run dev
```

When To Use This

- SaaS applications and dashboards
- Apps requiring user authentication
- Forms with server-side validation
- Projects with database-backed features

Trade-offs

More files - Each route may have `.svelte` + `.server.ts`
Server dependency - Cannot deploy as static site
Initial setup - Database and auth require configuration