

# spur-express

---

- [spur-express github repo](#)

## Journey

- Setup Github Repo: [spur-express github repo](#)
- chore: bootstrap Express + TypeScript backend with health check and gitignore
- Project initialization

```
mkdir backend frontend
cd backend
npm init -y
npm install express cors dotenv
npm install -D typescript ts-node-dev @types/node @types/express
npx tsc --init
```

- tsconfig.json: Configured TypeScript for a stable Node.js backend with strict type safety and a clear source-to-build output flow.

```
{
  "compilerOptions": {
    "target": "ES2020",
    "module": "commonjs",
    "rootDir": "src",
    "outDir": "dist",
    "strict": true,
    "esModuleInterop": true
  }
}
```

- src/app.ts: Created a minimal Express app with CORS and JSON middleware, plus a /health endpoint to verify server liveness.

```
import express from "express";
import cors from "cors";

const app = express();

app.use(cors());
app.use(express.json());

app.get("/health", (_, res) => {
  res.json({ status: "ok" });
});
```

```
export default app;
```

- src/server.ts: Separated server startup from app configuration and enabled environment variable loading for deployment readiness.

```
import app from "./app";
import dotenv from "dotenv";

dotenv.config();

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {
  console.log(`spur-express backend running on port ${PORT}`);
});
```

- package.json: Added development, build, and production scripts to support local development and deployment on Render.

```
"scripts": {
  "dev": "ts-node-dev --respawn src/server.ts",
  "build": "tsc",
  "start": "node dist/server.js"
}
```

- Started the backend in watch mode to ensure the server runs correctly during development.

```
npm run dev
```

- Health check validation

```
http://localhost:3000/health
```

- Confirmed the backend is running successfully and responding to requests.

```
{ "status": "ok" }
```