Steps to set up an Express and TypeScript project locally and deploy it to Vercel: Initialize the Project.

|  |
| --- |
| mkdir my-express-app  cd my-express-app  npm init -y |

Install Dependencies.

|  |
| --- |
| npm install express typescript @types/express ts-node concurrently nodemon  npm install -D @types/node |

Configure TypeScript.

|  |
| --- |
| npx tsc --init |

Modify tsconfig.json (example):

|  |
| --- |
| {  "compilerOptions": {  "target": "es2016",  "module": "commonjs",  "outDir": "./dist",  "esModuleInterop": true,  "forceConsistentCasingInFileNames": true,  "strict": true,  "skipLibCheck": true  }  } |

Create Express Server.

Create src/index.ts:

|  |
| --- |
| import express, { Request, Response } from 'express';   const app = express();  const port = process.env.PORT || 3000;   app.get('/', (req: Request, res: Response) => {  res.send('Hello from Express and TypeScript!');  });   app.listen(port, () => {  console.log(`Server is running on port ${port}`);  });    export default app; |

Update package.json Scripts.

|  |
| --- |
| "scripts": {  "build": "tsc",  "start": "node dist/index.js",  "dev": "concurrently \"npx tsc --watch\" \"nodemon -q dist/index.js\""  } |

Create vercel.json (for Vercel deployment).

|  |
| --- |
| {  "version": 2,  "builds": [  {  "src": "dist/index.js",  "use": "@vercel/node"  }  ],  "routes": [  {  "src": "/(.\*)",  "dest": "/dist/index.js"  }  ]  } |

Run Locally.

|  |
| --- |
| npm run dev |

* **Deploy to Vercel:**
  + Create a Vercel account and install the Vercel CLI: npm i -g vercel
  + Run vercel in your project directory and follow the prompts.