

# Welcome to React Router!








---

A modern, production-ready template for building full-stack React applications using React Router.



Open in StackBlitz

## Features

-  Server-side rendering
-  Hot Module Replacement (HMR)
-  Asset bundling and optimization
-  Data loading and mutations
-  TypeScript by default
-  TailwindCSS for styling
-  [React Router docs](#)

## Getting Started

### Installation

Install the dependencies:

```
npm install
```

### Development

Start the development server with HMR:

```
npm run dev
```

Your application will be available at <http://localhost:5173>.

## Building for Production

Create a production build:

```
npm run build
```

## Deployment

### Docker Deployment

To build and run using Docker:

```
docker build -t my-app .

# Run the container
docker run -p 3000:3000 my-app
```

The containerized application can be deployed to any platform that supports Docker, including:

- AWS ECS
- Google Cloud Run
- Azure Container Apps
- Digital Ocean App Platform
- Fly.io
- Railway

## DIY Deployment

If you're familiar with deploying Node applications, the built-in app server is production-ready.

Make sure to deploy the output of `npm run build`

```
├─ package.json
├─ package-lock.json (or pnpm-lock.yaml, or bun.lockb)
├─ build/
│   └─ client/      # Static assets
│   └─ server/      # Server-side code
```

## Styling

This template comes with [Tailwind CSS](#) already configured for a simple default starting experience. You can use whatever CSS framework you prefer.

## Creating Admin Users

To create an admin user, you can use any of these formats:

### Using named parameters with init-admin

```
npm run init-admin -- --email admin@example.com --password securePassword --
name "Admin User" --role manager
```

### Using positional parameters with init-admin

```
npm run init-admin -- admin@example.com securePassword "Admin User" manager
```


### Using named parameters with create-admin

```
npm run create-admin -- --email admin@example.com --password securePassword --  
name "Admin User" --role manager
```

### Using positional parameters with create-admin

```
npm run create-admin -- admin@example.com securePassword "Admin User" manager
```

---

Built with  using React Router.