Deployment

To deploy your application, you have to compile it, and then host the JavaScript, CSS, and HTML on a web server. Built Angular applications are very portable and can live in any environment or served by any technology, such as Node, Java, .NET, PHP, and many others.

> Whether you came here directly from Your First App, or completed the entire online store application through the Routing, Managing Data, and Forms sections, you have an application that you can deploy by following the instructions in this section.

Share your application

StackBlitz projects are public by default, allowing you to share your Angular app via the project URL. Keep in mind that this is a great way to share ideas and prototypes, but it is not intended for production hosting.

1. In your StackBlitz project, make sure you have forked or saved your project.
2. In the preview pane, you should see a URL that looks like `https://<Project ID>.stackblitz.io`.
3. Share this URL with a friend or colleague.
4. Users that visit your URL will see a development server start up, and then your application will load.

Building locally

To build your application locally or for production, you will need to download the source code from your StackBlitz project. Click the `Download Project` icon in the left menu across from `Project` to download your files.

Once you have the source code downloaded and unzipped, use the Angular Console to serve the application, or you install `Node.js` and have the Angular CLI installed.

From the terminal, install the Angular CLI globally with:

```
npm install -g @angular/cli
```

This will install the command `ng` into your system, which is the command you use to create new workspaces, new projects, serve your application during development, or produce builds that can be shared or distributed.

Create a new Angular CLI workspace using the `ng new` command:

```
ng new my-project-name
```

From there you replace the `/src` folder with the one from your `StackBlitz` download, and then perform a build.

```
ng build --prod
```

This will produce the files that you need to deploy.

> If the above `ng build` command throws an error about missing packages, append the missing dependencies in your local project's `package.json` file to match the one in the downloaded StackBlitz project.

Hosting the built project

The files in the `dist/my-project-name` folder are static and can be hosted on any web server capable of serving files (`Node.js`, Java, .NET) or any backend (Firebase, Google Cloud, App Engine, others).

Hosting an Angular app on Firebase

One of the easiest ways to get your site live is to host it using Firebase.

1. Sign up for a firebase account on Firebase.
2. Create a new project, giving it any name you like.
3. Install the `firebase-tools` CLI that will handle your deployment using `npm install -g firebase-tools`.
4. Connect your CLI to your Firebase account and initialize the connection to your project using `firebase login` and `firebase init`.
5. Follow the prompts to select the `Firebase` project you are creating for hosting.

- Select the `Hosting` option on the first prompt.
- Select the project you previously created on Firebase.
- Select `dist/my-project-name` as the public directory.

1. Deploy your application with `firebase deploy`, because the command `firebase init` has created a `firebase.json` file that tells Firebase how to serve your app.
2. Once deployed, visit https://your-firebase-project-name.firebaseapp.com to see it live!

Hosting an Angular app anywhere else

To host an Angular app on another web host, you'll need to upload or send the files to the host. Because you are building a Single Page Application, you'll also need to make sure you redirect any invalid URLs to your `index.html` file. Learn more about development and distribution of your application in the Building & Serving and Deployment guides.

Join our community

You are now an Angular developer! Share this moment, tell us what you thought of this Getting Started, or submit suggestions for future editions.

Angular offers many more capabilities, and you now have a foundation that empowers you to build an application and explore those other capabilities:

- Angular provides advanced capabilities for mobile apps, animation, internationalization, server-side rendering, and more.
- Angular Material offers an extensive library of Material Design components.
- Angular Protractor offers an end-to-end testing framework for Angular apps.
- Angular also has an extensive network of 3rd-party tools and libraries.

Keep current by following the Angular blog.