

Deploy to Firebase

After we've built a full-fledged application in React, the final step is deployment. It is the tipping point of getting your ideas into the world, from learning how to code to producing applications. We will use Firebase Hosting for deployment.

Firebase works for create-react-app, as well as most libraries and frameworks like Angular and Vue. First, install the Firebase CLI globally to the node modules:

```
npm install -g firebase-tools
```

Using a global installation of the Firebase CLI lets us deploy applications without concern over project dependency. For any globally-installed node package, remember to update it to a newer version with the same command as often as you can:

```
npm install -g firebase-tools
```

If you don't have a Firebase project yet, sign up for a [Firebase account](#) and create a new project there. Then you can associate the Firebase CLI with the Firebase account (Google account):

```
firebase login
```

- A URL will display in the command line that you can open in a browser, or the Firebase CLI opens it.
- Choose a Google account to create a Firebase project, and give Google the necessary permissions. Return to the command line to verify a successful login.
- Next, move to the project's folder and execute the following command.

Next, move to the project's folder and execute the following command, which initializes a Firebase project for the Firebase hosting features:

```
firebase init
```

- Next, choose the Hosting option. If you're interested in using another tool next to Firebase Hosting, add other options:
 - Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices.
 - Database: Deploy Firebase Realtime Database Rules
 - Firestore: Deploy rules and create indexes for Firestore
 - Functions: Configure and deploy Cloud Functions
 - Hosting: Configure and deploy Firebase Hosting sites
 - Storage: Deploy Cloud Storage security rules

Google becomes aware of all Firebase projects associated with an account after login, and we can select one from the Firebase platform:

First, let's associate this project directory with a Firebase project. You can create multiple project aliases by running `firebase use --add`, but for now we'll just set up a default project.

- Select a default Firebase project for this directory:

```
my-react-project-abc123 (my-react-project)
```

Using project my-react-project-abc123 (my-react-project)

- There are a few other configuration steps to define. Instead of using the default *public/* folder, we want to use the *build/* folder from create-react-app. If you set up the bundling with a tool like Webpack yourself, you can choose the appropriate name for the build folder:
 - What do you want to use as your public directory? build
 - Configure as a single-page app (rewrite all urls to /index.html)? Yes
 - File public/index.html already exists. Overwrite? No

The create-react-app application creates a *build/* folder after we perform the `npm run build` for the first time. The folder contains all the merged content

from the *public/* folder and the *src/* folder. Since it is a single page application,

we want to redirect the user to the *index.html* file, so the React router can handle client-side routing.

Now your Firebase initialization is complete. This step created a few configuration files for Firebase Hosting in your project's folder. You can read more about them in [Firebase's documentation](#) for configuring redirects, a 404 page, or headers. Finally, deploy your React application with Firebase in the command line:

```
firebase deploy
```

After a successful deployment, you should see a similar output with your project's identifier:

```
Project Console: https://console.firebase.google.com/project/my-react-project-abc123/overview
```

```
Hosting URL: https://my-react-project-abc123.firebaseio.com
```

Visit both pages to observe the results. The first link navigates to your Firebase project's dashboard, where you'll see a new panel for the Firebase Hosting. The second link navigates to your deployed React application.

If you see a blank page for your deployed React application, make sure the `public` key/value pair in the *firebase.json* is set to `build`, or whichever name you chose for this folder. Second, verify you've run the build script for your React app with `npm run build`. Finally, check out the [official troubleshoot area for deploying create-react-app applications to Firebase](#). Try another deployment with `firebase deploy`.

Exercises

- Read more about [Firebase Hosting](#).
- Connect your domain to your [Firebase deployed application](#).
- Optional: If you want to have a managed cloud server, check out [DigitalOcean](#). It's more work, but it allows more control. I host all my websites, web applications, and backend APIs [there](#).

