Project Configuration

This lesson showcases the configuration code which contains key pieces of information about the project.

WE'LL COVER THE FOLLOWING ∧

- Copying the Configuration to React
  - Environment Variables
  - Setting Up Your Own Environment

Copying the Configuration to React

We will use a JavaScript class to encapsulate all Firebase functionalities, Real-time Database, authentication, and a well-defined API for the whole application. We only need to instantiate the class once, after which it can be used to interact with the Firebase API (your custom Firebase interface).

Let's start by copying the configuration from our Firebase project's dashboard on their website to our application as a configuration object in a new `src/components/` **`Firebase/firebase.js`** file. Make sure to replace the capitalized keys with the corresponding keys from your copied configuration:

```
const config = {
  apiKey: YOUR_API_KEY,
  authDomain: YOUR_AUTH_DOMAIN,
  databaseURL: YOUR_DATABASE_URL,
  projectId: YOUR_PROJECT_ID,
  storageBucket: '',
  messagingSenderId: YOUR_MESSAGING_SENDER_ID,
};
```

Configuration

Environment Variables

Alternatively, we can also use React environment variables, but we must use the REACT_APP prefix when using `create-react-app` to set up the application:

```
const config = {
  apiKey: process.env.REACT_APP_API_KEY,
  authDomain: process.env.REACT_APP_AUTH_DOMAIN,
  databaseURL: process.env.REACT_APP_DATABASE_URL,
  projectId: process.env.REACT_APP_PROJECT_ID,
  storageBucket: process.env.REACT_APP_STORAGE_BUCKET,
  messagingSenderId: process.env.REACT_APP_MESSAGING_SENDER_ID,
};
```

Now we can define the environmental variables in a new `.env` file in the project's root folder. The `.env` file can also be added to the `.gitginore` file (in case someone is using **git**), so that our Firebase credentials are not exposed publicly on a platform like GitHub.

```
REACT_APP_API_KEY=XXXXxxxx
REACT_APP_AUTH_DOMAIN=xxxxXXXX.firebaseapp.com
REACT_APP_DATABASE_URL=https://xxxXXXX.firebaseio.com
REACT_APP_PROJECT_ID=xxxxXXXX
REACT_APP_STORAGE_BUCKET=xxxxXXXX.appspot.com
REACT_APP_MESSAGING_SENDER_ID=xxxxXXXX
```

Both ways work. We can either define the configuration inline in the source code or as environment variables.

Environment variables are more secure and should be used when uploading projects to a version control system like **git**, though we will be continuing with the Firebase setup.
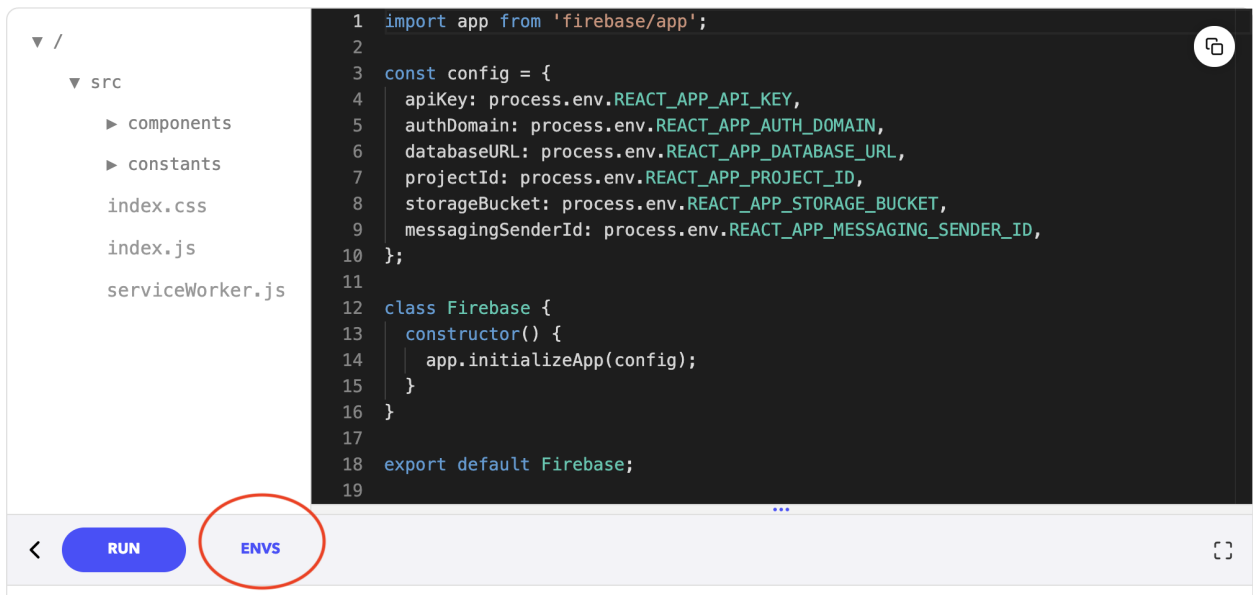
Setting Up Your Own Environment

From this point onward in the course, we'll be dealing with Firebase using a project configuration.

You must have obtained your own Firebase configuration when we were setting up a new project.
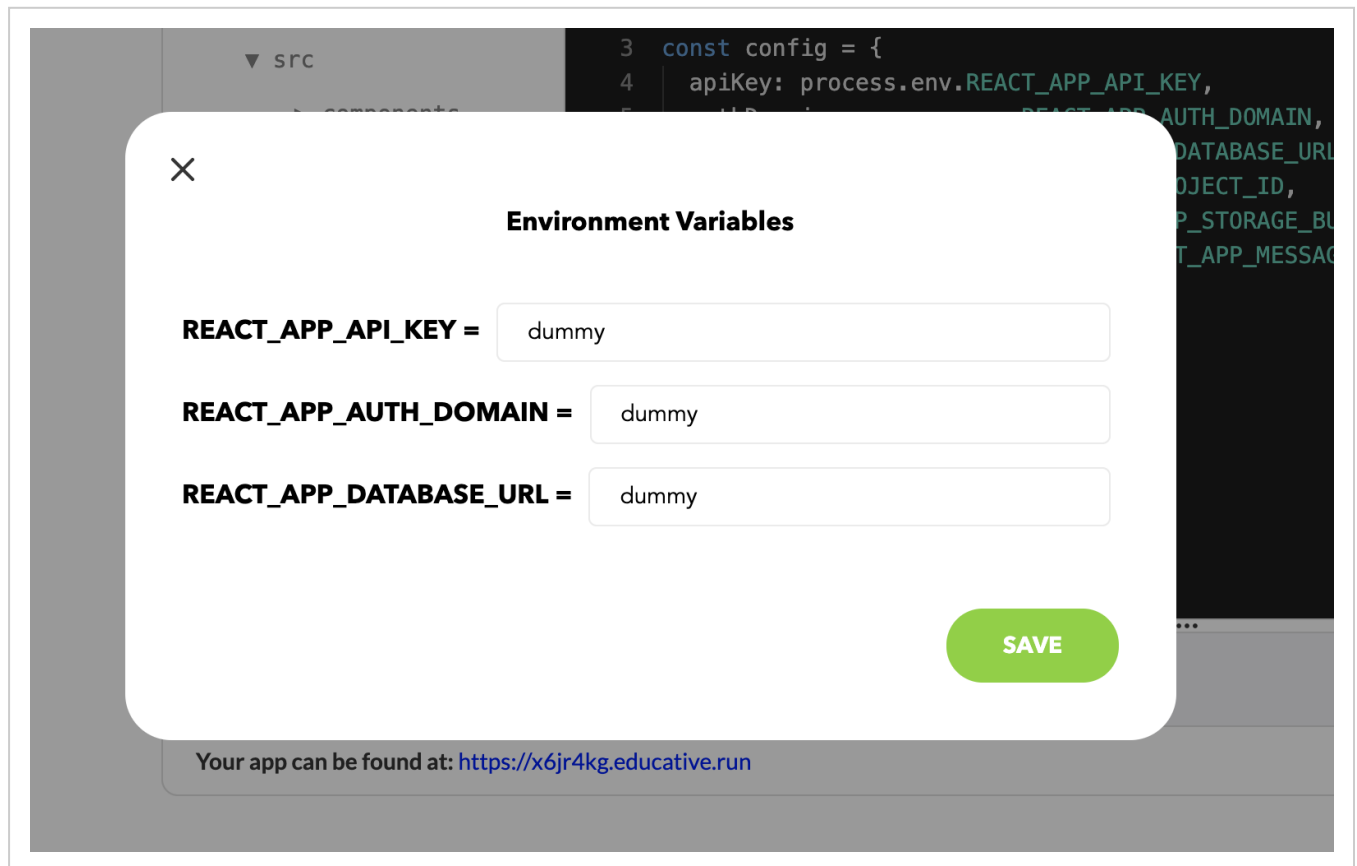
In order to see the React-Firebase communication later on in the course, **you'll need to set your environment variables on our platform**. These will be used as the predefined configuration.

You will come across a similar interface as shown in the image below in the upcoming lessons. Click on the **ENV** tab which is located right next to the **RUN**

button.

A prompt similar to the image below will appear with the value "dummy" written in each of the access tokens.

Scroll down to replace the dummy value with your own **value** in each field according to the configuration you have and save your settings by clicking the save button located at the bottom right side of the prompt. Make sure that the inverted commas have been removed from the values in the configuration.

For example, for an **API key**, `"abcdef"`, the value in the **REACT_APP_API_KEY**

will be `abcdef` .

Now, you'll be able to make changes in your own database later on in the course!

Keep in mind, that this is a **requirement** for the rest of the course. If these values are not set, the codes won't work.

> **Disclaimer**: Educative does not store any configuration details. The environment you set up will be stored on your local machine.

In the next lesson, we will import Firebase and create a new Firebase class to initialize Firebase with this configuration.