강의 소개

리액트 앱 설치하기

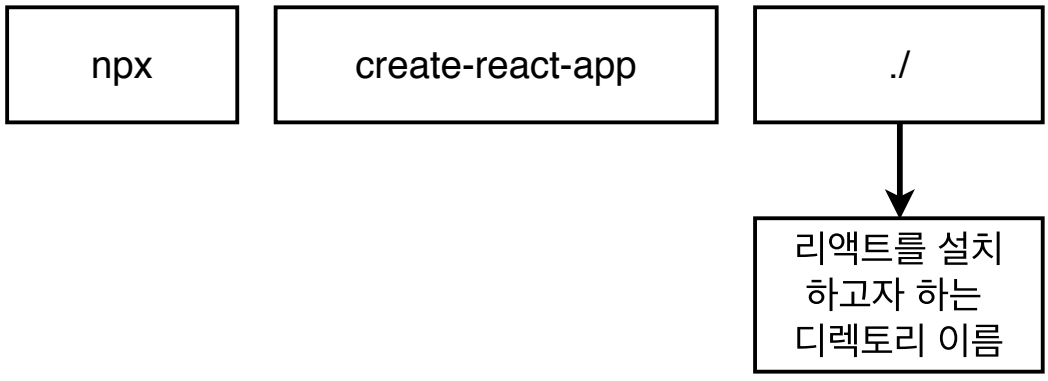
먼저는 리액트를 다운 받기 위해서
노드가 컴퓨터에 다운받아져있어야 합니다.

이미 컴퓨터에 노드가 받아져있는지 확인 하기 위해서는
node -v 해서 체크가 가능합니다.

만약 다운이 안받아져 있다면 node 공식 웹사이트에 가서
다운을 받으면 됩니다.

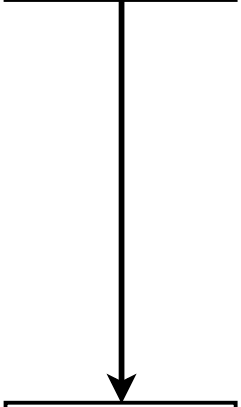
노드가 받아져있다면 이제는 리액트를 설치할 차례입니다.
리액트를 다운받아서 설치하기위해서는 굉장히 간단하게
아래에 보이는 명령어를 입력하면 됩니다.
이 강의는 리액트 강의가 아니기에 리액트에 대한 자세한 설명
은 생략하겠습니다.

리액트를 설치하기 위한 명령어



이렇게만 하면 이제 다 설치가 된 것이며
이제 한번 실행을 해보겠습니다.
실행 할때는 아래 명령어를 입력하시면 됩니다.

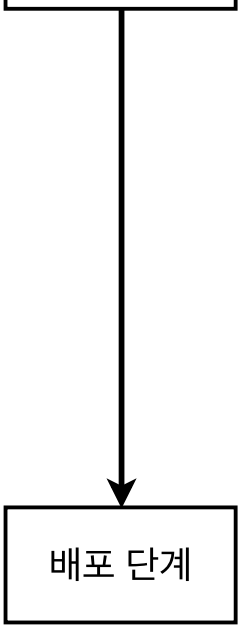
개발 단계



npm run start

개발이 완료 되었다면
개발한것에 문제가 있는지 없는지
테스트를 해볼수 있습니다.
테스트는 아래 명령어로 할 수 있습니다.

npm run test



이렇게 테스트까지 완성이 된다면
이제는 배포를 해서
다른 사람들도 이용 할 수 있게 해줘야겠죠.
그러기 위해서는 배포를 위한
폴더와 파일들을 생성해야 합니다
생성하기 위해서는 아래 명령어를 입력하시면 됩니다.

npm run build

이렇게 배포를 위한 명령어까지 작성하면
배포를 할때 사용할수 있는 build폴더와
그안에 많은 파일들이 생성이 됩니다.
이렇게 기본적인 리액트 어플리케이션 설치를 완성했습니다.

Create React App 구조 (Structure)

```
my-app/
  README.md
  node_modules/
  package.json
  public/
    index.html
    favicon.ico
  src/
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
```

참조
https://create-react-app.dev/docs/folder-structure/

Files that must exist with exact filenames

1. public/index.html -> page template
2. src/index.js -> Javascript entry point

Files that are editable and deletable

```
my-app/
  README.md
  node_modules/
  package.json
  public/
    index.html
    favicon.ico
  src/
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
```

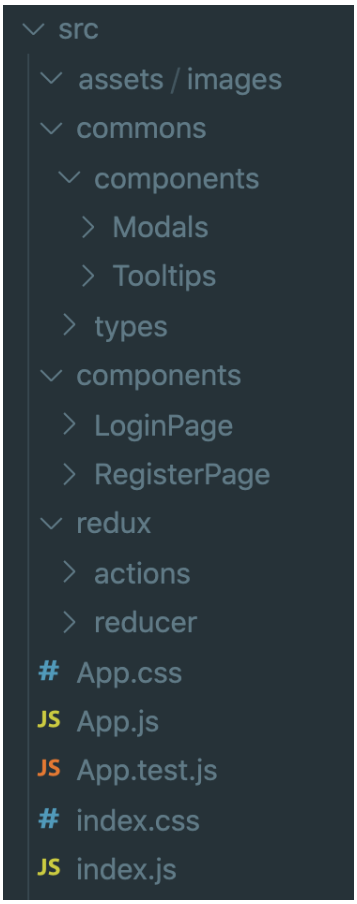
여기에 쓰인 파일들은 오직 public/index.html만 쓰일수 있다.

이곳에 JS 파일과 CSS 파일들을 넣으면 된다.
그리고 Webpack은 여기에 있는 파일만 본다.
그래서 이 폴더 이외에 넣는것은
webpack에 의해서 처리되지 않음.

You may create subdirectories inside `src`. For faster rebuilds, only files inside `src` are processed by Webpack. You need to put any JS and CSS files inside `src`, otherwise Webpack won't see them.

Only files inside `public` can be used from `public/index.html`.

여기서 부터는 이 Boiler Plate에 특성화된 구조 설명



assets/ → 이미지, CSS, JS 파일들 보관

commons/
- components → 여러 페이지에서 쓰일수 있는 것들
- types → typescript를 위해서 type 지정

components → 이 안에는 Page들을 넣는다

redux/
- actions
- reducer

→ Redux 를 위한 폴더들

App.js → Routing 관련 일을 처리한다.

React Router Dom

페이지 이동을 할때
React Router Dom 이라는 것을
사용합니다.

어떻게 사용하는지 웹사이트 참조 → https://reacttraining.com/react-router/web/example/basic

Dependency 다운로드 → npm install react-router-dom --save

react router dom 코드
Documentation에서 복사해서 붙여넣기

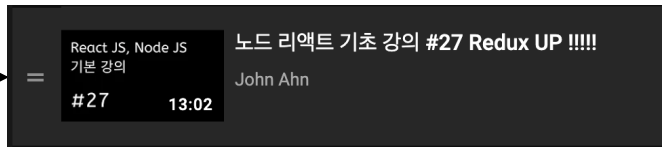
그 후 우리 앱에 맞게 바꿔서 넣기

```
import {
  BrowserRouter as Router,
  Switch,
  Route,
} from "react-router-dom";
import ChatPage from './components/ChatPage/ChatPage';
import LoginPage from './components/LoginPage/LoginPage';
import RegisterPage from './components/RegisterPage/RegisterPage';

function App() {
  return (
    <Router>
      <Switch>
        <Route exact path="/" component={ChatPage} />
        <Route exact path="/register" component={LoginPage} />
        <Route exact path="/login" component={RegisterPage} />
      </Switch>
    </Router>
  );
}
```

Setting Up Redux !

Redux 기본 설명은...



다운 받아야 할 Dependency들

1. redux

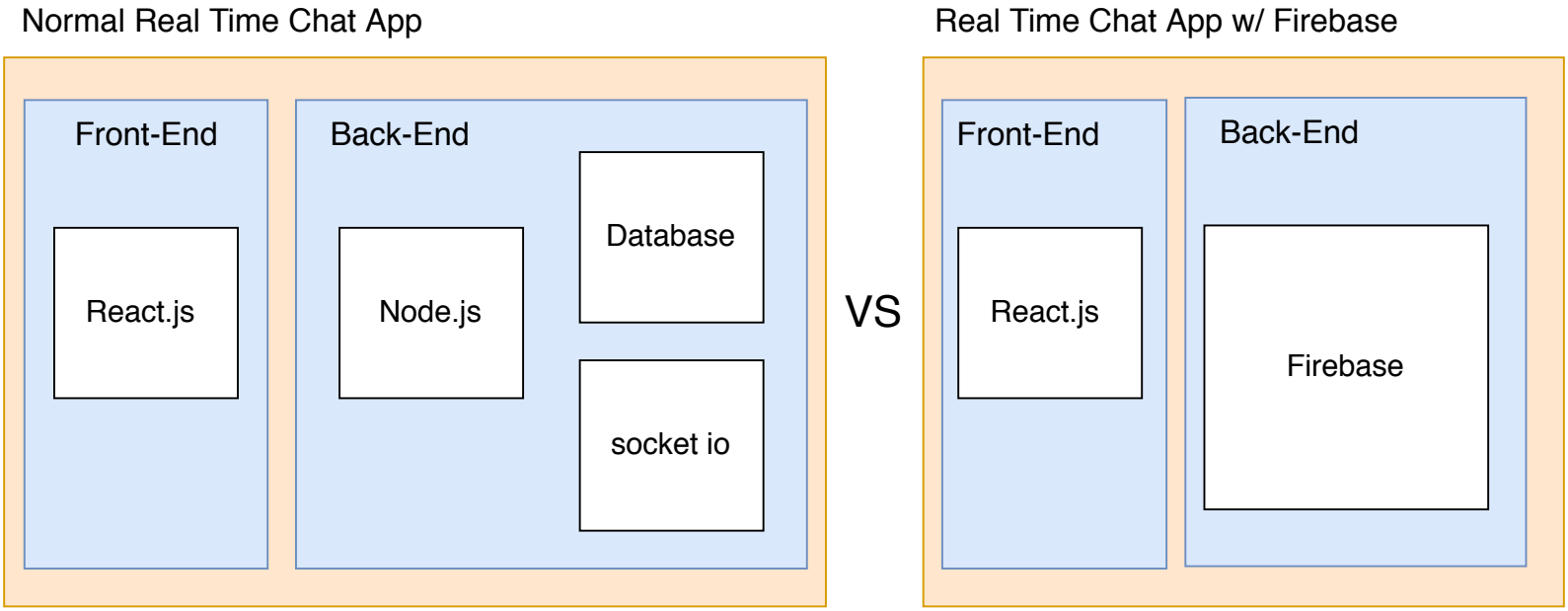
2.react-redux

3.redux-promis

4.redux-thunk

Redux 기본 구조(scaffolding) 만들기

애플리케이션 구조



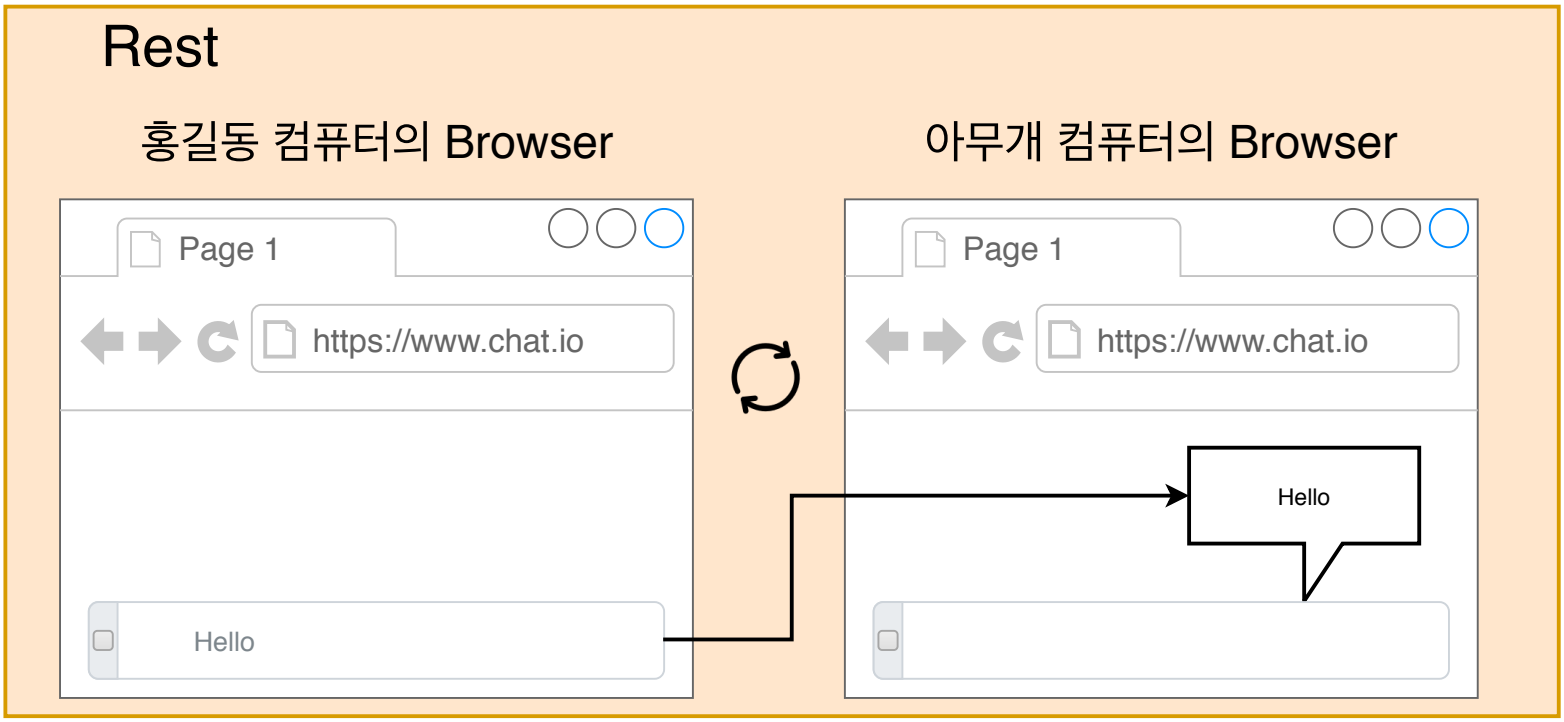
어떻게 이렇게 다른 구조가...?
이 부분을 이해하기 위해서...

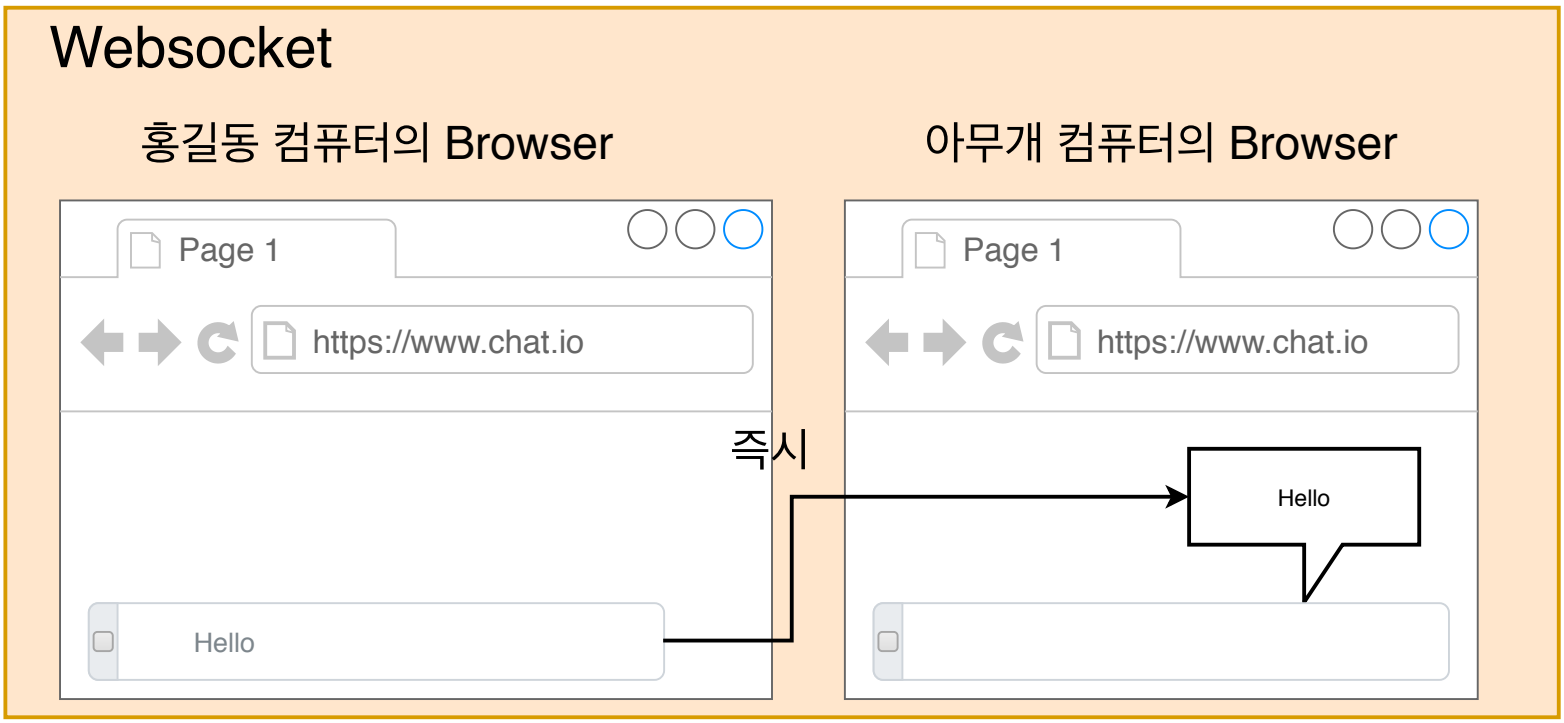
1.REST vs Websockets

2.Firebase 에 대해서

REST vs WebSockets

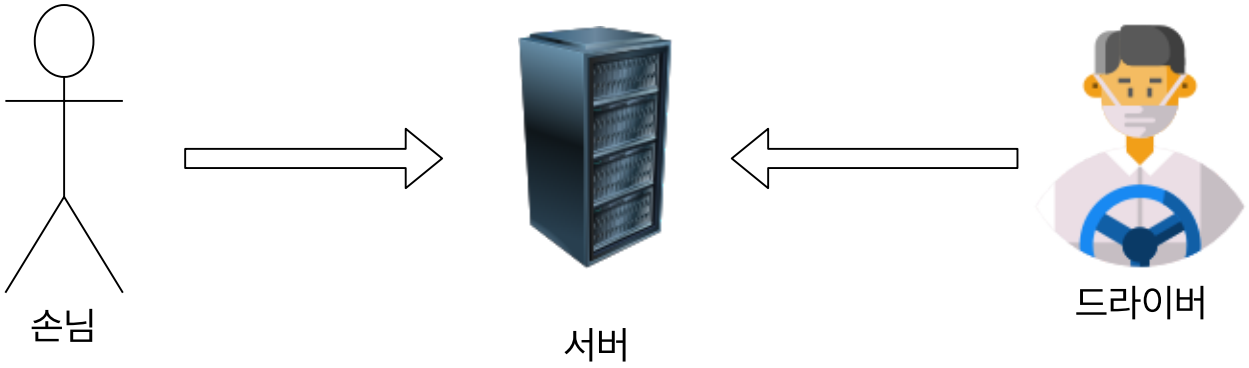
1.REST vs Websockets





예를 들어서 살펴보기. 택시 애플리케이션 !

Rest

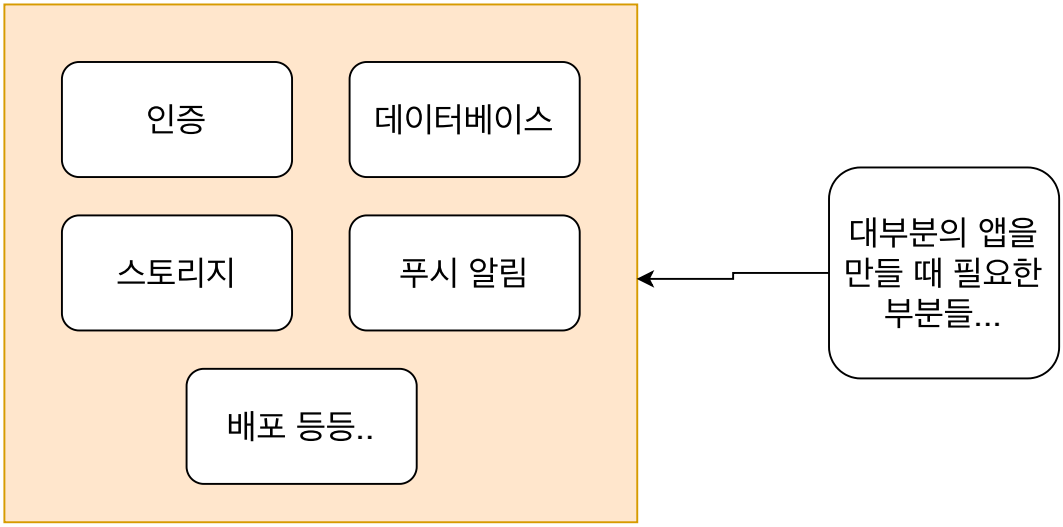


Web Socket

Rest API 같은 경우는 계속 한 방향으로 손님이 서버에 드라이버가 어디에 있는지 지속해서 요청을 보내서 드라이버의 위치를 알 수가 있다.
하지만 Web Socket 같은 경우는 양방향으로 드라이버가자신의 위치를 나타내서 바로 손님에게 그 정보가 가게 할 수 있다.

Firebase ?

Firebase is a platform developed by Google for creating mobile and web applications.



위에와 같이 모든 어플리케이션을 만들 때 필요한 부분들을 자동으로 만들어 주는 플랫폼입니다.

이중에서 많이 특별한 데이터베이스!

Firebase에서 사용하는 데이터베이스는 Mysql이나 오라클 같은 관계형 데이터베이스가 아닌 MongoDB 같은 NoSQL 기반의 Document 형식의 빠르고 간편한 데이터베이스 입니다.
그리고 정말 특별한 점은 RTSP라는 Real Time Stream Protocol 방식을 지원합니다.

RTSP는 실시간으로 데이터들을 전송해주는 방식입니다.
이 방식 덕분에 실시간 기능이 요하는 채팅 앱이나 택시앱같은 걸 쉽게 구현이 가능합니다.

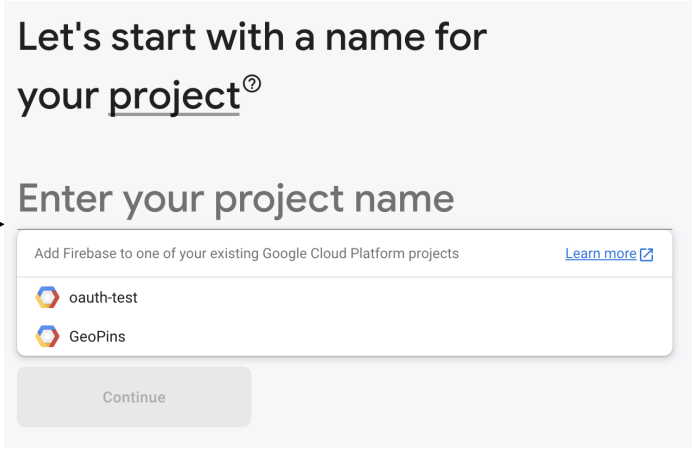
실제로 데이터를 하나 넣어서 실시간으로 확인해보기 !

Firebase 사용하기

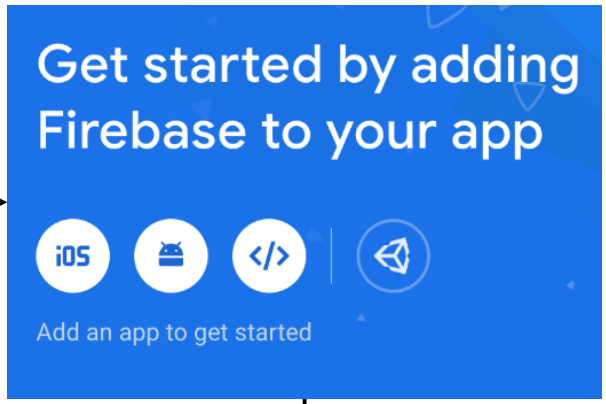
Firebase 사이트로 이동 → https://firebase.google.com/

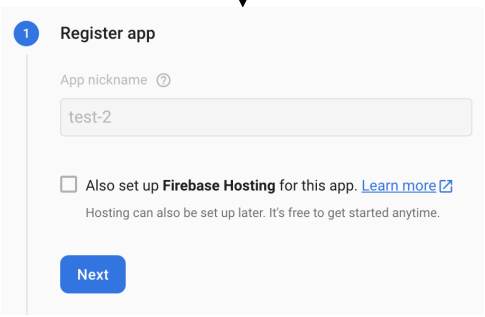
회원 가입 & 로그인

프로젝트 생성

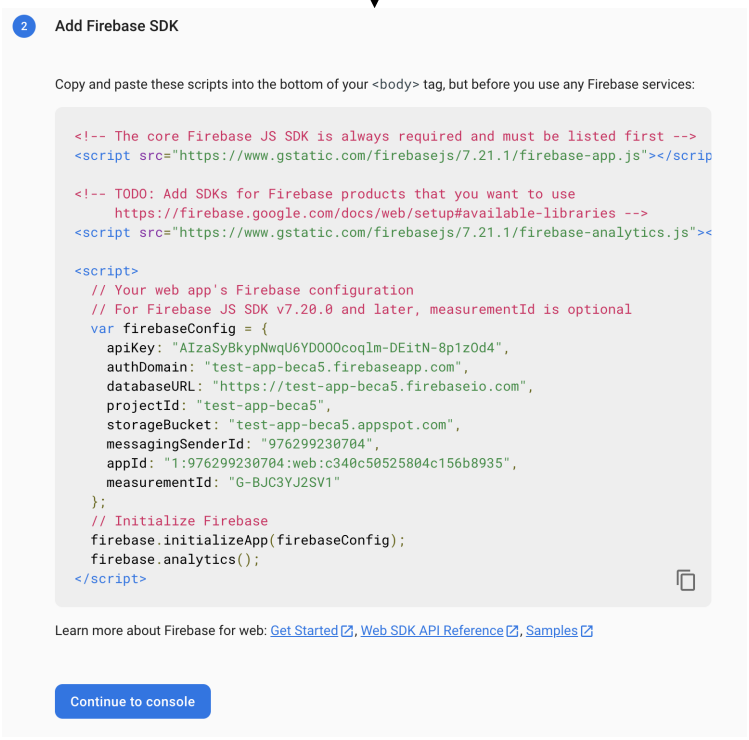


Firebase를 어플리케이션에 연결하기





설명에 맞게 따라하기



npm install firebase --save