

Node.js 프로젝트 설정 및 실행 가이드

1. 프로젝트 디렉토리 생성 및 초기화

먼저, 프로젝트를 위한 새 디렉토리를 생성하고, Node.js 프로젝트를 초기화합니다.

```
mkdir react-component-tutorial  
cd react-component-tutorial  
npm init -y
```

이 명령어는 `react-component-tutorial` 이라는 새 디렉토리를 만들고, 그 디렉토리로 이동하여 기본 설정 파일인 `package.json` 을 생성합니다.

2. 필요한 패키지 설치

프로젝트에서 사용할 몇 가지 패키지를 설치합니다.

```
npm install express
```

이 명령어는 웹 서버를 실행하기 위한 `express` 패키지를 설치합니다.

3. 프로젝트 구조 설정

프로젝트 구조를 아래와 같이 설정합니다.

```
react-component-tutorial/  
├── public/  
│   └── index.html  
├── server.js  
└── package.json
```

- `public/` : HTML 파일을 포함하는 폴더입니다.
- `server.js` : Express.js 서버 설정 파일입니다.
- `package.json` : 프로젝트 정보와 의존성을 정의한 파일입니다.

4. index.html 파일 작성

public/index.html 파일을 생성하고 다음과 같이 작성합니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>React Component Example</title>
</head>
<body>

  <h1>React Component with LEGO Blocks Example</h1>
  <button id="callApiButton">Call API</button>
  <p id="responseText">API response will appear here.</p>

  <script>
    document.getElementById('callApiButton').addEventListener('click', function() {
      fetch('/api/greet')
        .then(response => response.json())
        .then(data => {
          document.getElementById('responseText').innerText = data.message;
        })
        .catch(error => console.error('Error:', error));
    });
  </script>

</body>
</html>
```

5. `server.js` 파일 작성

루트 디렉토리에 `server.js` 파일을 생성하고 다음과 같이 작성합니다.

```
const express = require('express');
const path = require('path');
const app = express();
const port = 3000;

// 정적 파일을 서빙
app.use(express.static(path.join(__dirname, 'public')));

// 간단한 API 엔드포인트
app.get('/api/greet', (req, res) => {
  res.json({ message: 'Hello, World!' });
});

// 서버 실행
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

이 코드에서:

- `express.static` 을 사용하여 `public` 디렉토리 내의 정적 파일(`index.html` 등)을 서빙합니다.
- `/api/greet` 경로로 간단한 API를 설정합니다.
- 서버는 포트 `3000` 에서 실행됩니다.

6. 프로젝트 실행

이제 서버를 실행하여 프로젝트를 확인할 수 있습니다.

```
node server.js
```

이 명령을 실행하면 서버가 시작되고, 다음과 같은 메시지가 출력됩니다:

```
Server is running on http://localhost:3000
```


7. 웹 브라우저에서 확인

웹 브라우저를 열고 `http://localhost:3000` 으로 이동합니다. 다음과 같은 페이지가 나타납니다:

- "React Component with LEGO Blocks Example"이라는 제목이 표시된 페이지가 나타납니다.
- "Call API" 버튼을 클릭하면 `/api/greet` API를 호출하여 "Hello, World!" 메시지가 페이지에 표시됩니다.

이로써, 간단한 Node.js 프로젝트 설정과 실행 과정을 완료했습니다. 이 프로젝트 구조를 사용하면, 추가적인 React 컴포넌트를 통합하거나 다른 API 엔드포인트를 추가하는 등 더 복잡한 애플리케이션으로 확장할 수 있습니다.

← → ↻ ⓘ localhost:3000

React Component with LEGO Blocks Example

Call API

API response will appear here.

← → ↻ ⓘ localhost:3000

React Component with LEGO Blocks Example

Call API

Hello, World!