

React, Express 연동하여 사용하기

일단 react app 을 하나 생성합니다. 이름은 cs-test

cmd 로 프로젝트를 생성할 폴더로 가서

```
npx create-react-app cs-test
```

```
# 설치 다되면
```

```
# 폴더로 이동해서
```

```
cd cs-test
```

```
#npm 프로젝트설정
```

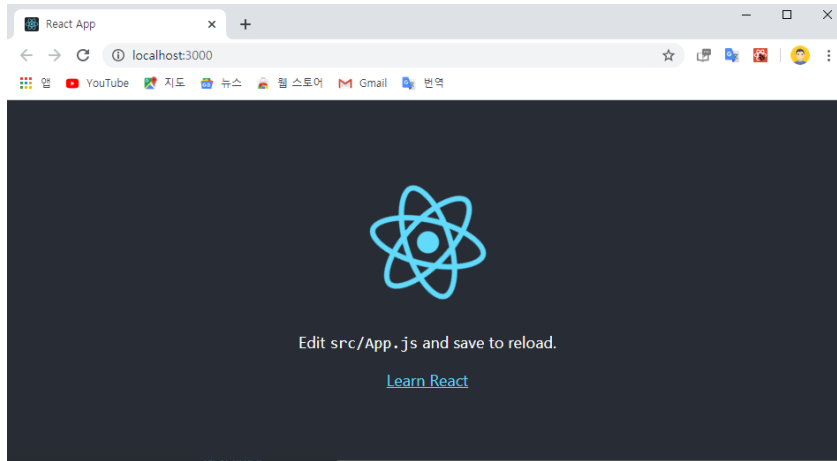
```
npm init
```

```
#기본설정으로 계속 Enter
```

```
# 일단 실행해봅니다.
```

```
npm start
```

그럼 익숙한 웹페이지가 뜹니다.



express 도 추가해 줍니다.

```
npm add express --save
```

서버에서 json 형식으로 데이터를 주고받을 거기때문
에 body-parser 도 설치해줍니다.

```
npm install body-parser
```

그리고 사용하시는 에디터로 소스를 열어서 수정합
니다.

```
src 아래에 App.js 를 수정합니다.
```

```
import React from 'react';
import logo from './logo.svg';
import './App.css';
```

```
class App extends React.Component {
```

```
  constructor(props) {
    super(props);
    this.state = {
      username: null
    };
  }
```

```
  render() {
    const {username} = this.state;
    return (
      <div className="App">
        <header className="App-header">
          {username ? `Hello
${username}` : 'Hello World'}
        </header>
      </div>
    );
  }
}
```

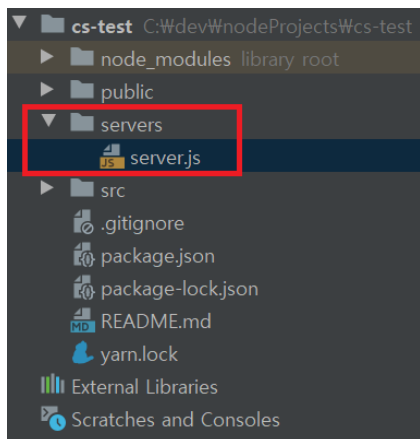
```
export default App;
```

이러면 페이지에 Hello World 가 표시되겠죠.

저 username 을 서버단에서 받아오도록 해보겠습니다.

root 에서 servers 라는 폴더를 만들고

server.js 파일을 생성합니다.



```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
const port = process.env.PORT || 3001;
```

```
app.use(bodyParser.json());
app.use('/api', (req, res) =>
  res.json({username: 'bryan'}));
```

```
app.listen(port, () => {
  console.log(`express is running on
  ${port}`);
```

```
})
```

이렇게 하고 프로젝트 root 에서 express 만 실행해
봅니다.

```
C:\dev\nodeProjects\cs-  
test>node ./servers/server.js  
express is running on 3001
```

웹페이지에서 확인. <http://localhost:3001/api>

{"username":"bryan"} 라고만 나옵니다. 된겁니다. ㅎㅎ

이제 client 쪽에서 이 데이터를 받아와야겠죠.

App.js 에서 componentDidMount() 를 추가합니다.

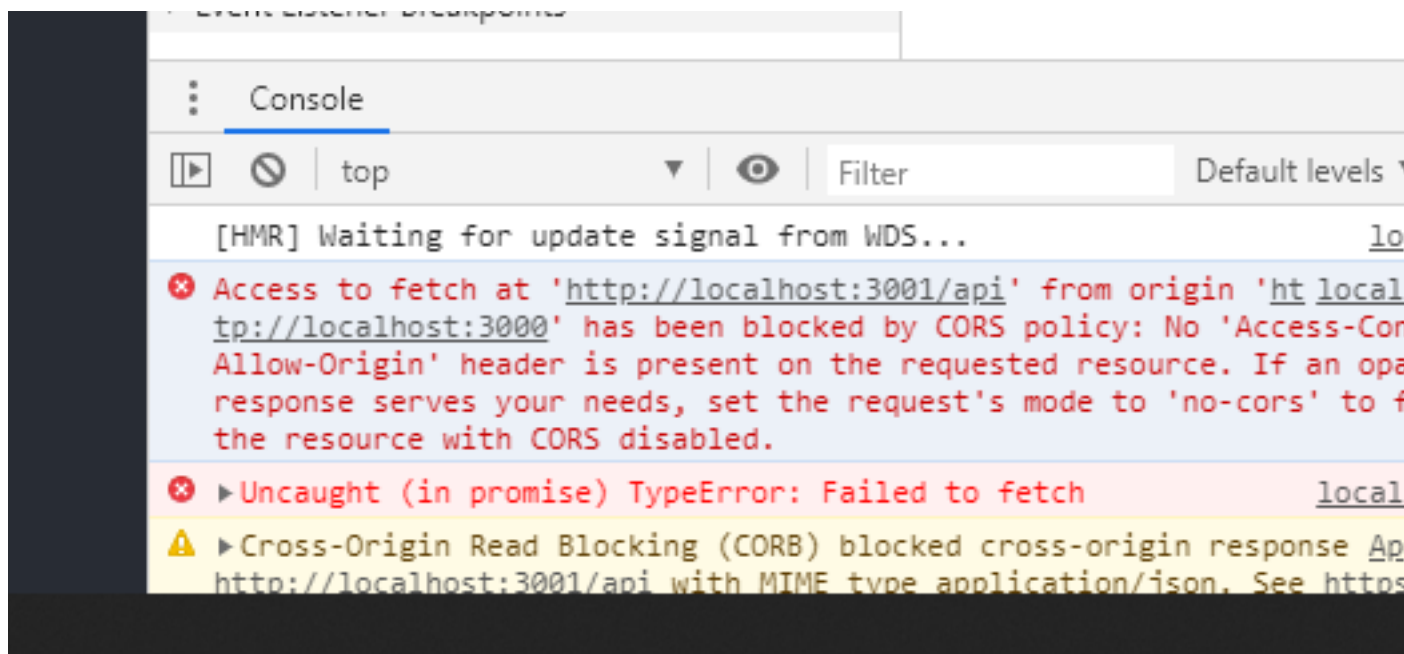
```
componentDidMount() {  
    fetch('http://localhost:3001/api')  
        .then(res=>res.json())  
        .then(data=>this.setState({us  
ername:data.username}));
```

```
}  
render() {  
  // ...  
}
```

그리고 ctrl+s 눌러서 저장하면 (webstorm 기준) 페이지가 새로고침되죠.

(다른 에디터 쓰시는분들 중 안되시는분들은 Ctrl+C로 중지하고 다시 npm start 하세요)

결과는~



오류. ㅎㅎ

Access-Control-Allow-Origin 관련 오류죠. 도메인이 다른곳에서 요청이 왔기때문에 허용안함! 이라고 서버 쪽에서 막아둔것입니다.

이건 서버가 어디까지 api 를 제공할 것인가에 따라 보안으로 막을수도 있고 열수도 있습니다.

일단 테스트니까 cors 를 허용하는 방법으로 해보겠습니다.

server.js 에서 cors 를 사용할것이기 때문에 우선 설치.

```
npm install cors --save
```

그리고 server.js 수정

```
const express = require('express');  
const app = express();  
const cors = require('cors');  
const bodyParser = require('body-parser');
```

```
const port =process.env.PORT || 3001;
```

```
app.use(cors());
```

```
app.use(bodyParser.json());
```

```
app.use('/api', (req, res)=>
```

```
res.json({username:'bryan'}));
```

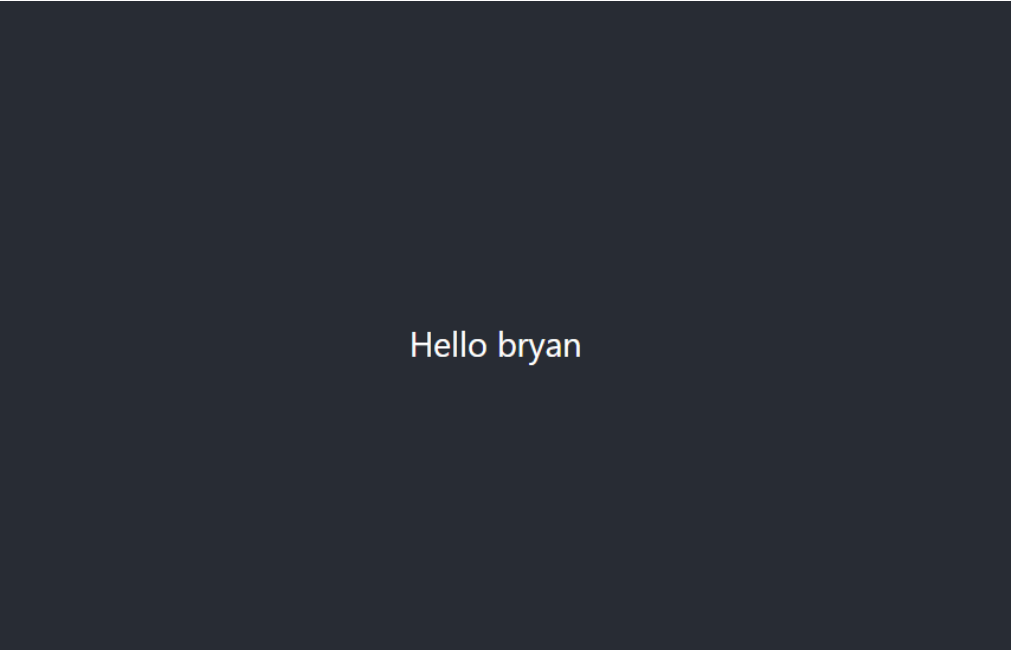
```
app.listen(port, ()=>{
```

```
  console.log(`express is running on  
${port}`);
```

```
})
```

서버 재시작. (Ctrl+C 후에 `node ./servers/server.js`)

react 페이지 새로고침하면



Hello bryan

원하던 결과가 나왔습니다.

React 에서 nodejs 의 데이터를 가져와서 뿌려준것이
죠.

이제 nodejs 에서 database 에 연결해서 데이터를 불러오면 react-node-db 연결이 되겠죠.