



Node.js + Reactjs + chat-gpt

본 강의교안을 무단 배포 및 게재를 할 경우 법적인 처벌을 받을 수 있음을 알려드립니다.



Overview

- What you will learn
 - Nodejs + reactjs + chat-gpt
 - Nodejs로 back-end 작성
 - Reactjs로 front-end 작성

Back-end 작성

- 환경설정

- 디렉토리생성

```
C:\>mkdir dev  
C:\>cd dev  
C:\dev>mkdir back-end  
C:\dev>cd back-end  
C:\dev\back-end>|
```

- 프로젝트 초기화

```
C:\dev\back-end>npm init -y  
Wrote to C:\dev\back-end\package.json:  
  
{  
  "name": "back-end",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```

Back-end 작성

■ 환경설정

■ 필요 라이브러리 다운로드 및 설정

```
C:\dev\back-end>npm install express@4.18.2 body-parser@1.20.2 cors@2.8.5 openai@3.2.1
```

■ 생성 후 아래와 같이 node_modules가 추가된 것을 확인

```
C:\dev\back-end>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 44A0-FD22

C:\dev\back-end 디렉터리

2023-11-10 오후 02:59 <DIR>          .
2023-11-10 오후 02:56 <DIR>          ..
2023-11-10 오후 03:02 <DIR>          node_modules
2023-11-10 오후 03:03          29,004 package-lock.json
2023-11-10 오후 03:02          348 package.json
                        2개 파일          29,352 바이트
                        3개 디렉터리 128,414,429,184 바이트 남음
```

Back-end 작성

- 환경설정
 - 필요 라이브러리 다운로드 및 설정

```
C:\dev\back-end>npm install express@4.18.2 body-parser@1.20.2 cors@2.8.5 openai@3.2.1
```

- 생성 후 아래와 같이 node_modules가 추가된 것을 확인

```
C:\dev\back-end>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 44A0-FD22

C:\dev\back-end 디렉터리

2023-11-10 오후 02:59 <DIR>          .
2023-11-10 오후 02:56 <DIR>          ..
2023-11-10 오후 03:02 <DIR>          node_modules
2023-11-10 오후 03:03          29,004 package-lock.json
2023-11-10 오후 03:02          348 package.json
                2개 파일          29,352 바이트
                3개 디렉터리 128,414,429,184 바이트 남음
```

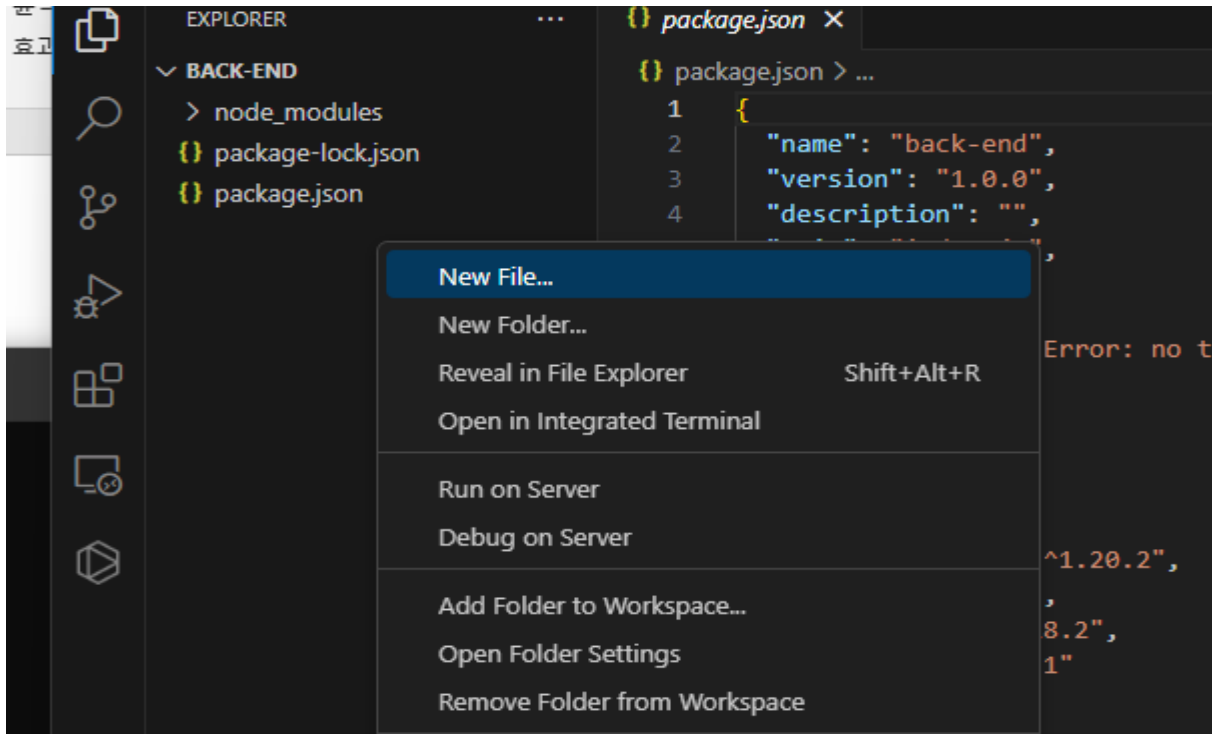
Back-end 작성

- 환경설정

- VSCode로 해당 폴더를 오픈한다

```
C:\dev\back-end>code .
```

- index.js 생성



Back-end 작성

■ 파일 작성

```
index.js x package.json
index.js > ...
1 import { Configuration, OpenAIApi } from "openai";
2 import express from "express";
3 import bodyParser from "body-parser";
4 import cors from "cors";
5
6 const app = express();
7 const port = 8000;
8 app.use(bodyParser.json());
9 app.use(cors());
10
11 const configuration = new Configuration({
12   organization: "org-NcFeGspmcbktUPFTEcilcyIA",
13   apiKey: "sk-pjJ7zUD2VcpkzbKsFkhXT3B1bkFJZaurloaTx3mETaa6jjCP",
14 });
15 const openai = new OpenAIApi(configuration);
16
17 app.post("/", async (request, response) => {
18   const { chat } = request.body;
19
20   const result = await openai.createChatCompletion({
21     model: "gpt-3.5-turbo",
22     messages: [
23       {
24         role: "system",
25         // content: "You are a EbereGPT. You can help with graphic design tasks",
26         content: "You are a helpful assistant",
27       },
28       chat
29     ],
30   });
31
32   response.json({
33     output: result.data.choices[0].message,
34   });
35 });
36
37 app.listen(port, () => {
38   console.log(`listening on port ${port}`);
39 });
40
```

Back-end 작성

■ 실행

- 터미널에서 아래 명령으로 실행한다

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\dev\back-end> node index
(node:11984) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.
(Use `node --trace-warnings ...` to show where the warning was created)
C:\dev\back-end\index.js:1
import { Configuration, OpenAIApi } from "openai";
^^^^^

SyntaxError: Cannot use import statement outside a module
    at internalCompileFunction (node:internal/vm:73:18)
    at wrapSafe (node:internal/modules/cjs/loader:1153:20)
    at Module._compile (node:internal/modules/cjs/loader:1205:27)
    at Module._extensions..js (node:internal/modules/cjs/loader:1295:10)
    at Module.load (node:internal/modules/cjs/loader:1091:32)
    at Module._load (node:internal/modules/cjs/loader:938:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:83:12)
    at node:internal/main/run_main_module:23:47

Node.js v20.9.0
PS C:\dev\back-end>
```


Back-end 작성

■ 실행

- package.json 파일에 `"type": "module"`을 아래와 같이 추가한다

```
{ } package.json X JS index.js
{ } package.json > ...
1  {
2    "name": "back-end",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "body-parser": "^1.20.2",
14     "cors": "^2.8.5",
15     "express": "^4.18.2",
16     "openai": "^3.2.1"
17   },
18   "type": "module"
19 }
20
```

Back-end 작성

- 실행

- 터미널에서 아래 명령으로 실행한다

```
35  },
36
37  app.listen(port, () => {
  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\dev\source\edu\react-nodejs-chatgpt-tutorial\backend> node index
listening on port 8000
|
```



Front-end 작성

- 환경설정

- 프로젝트 초기화

- cra를 이용하여 react 프로젝트 생성

```
C:\dev>npx create-react-app front-end

Creating a new React app in C:\dev\front-end.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1462 packages in 27s

242 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

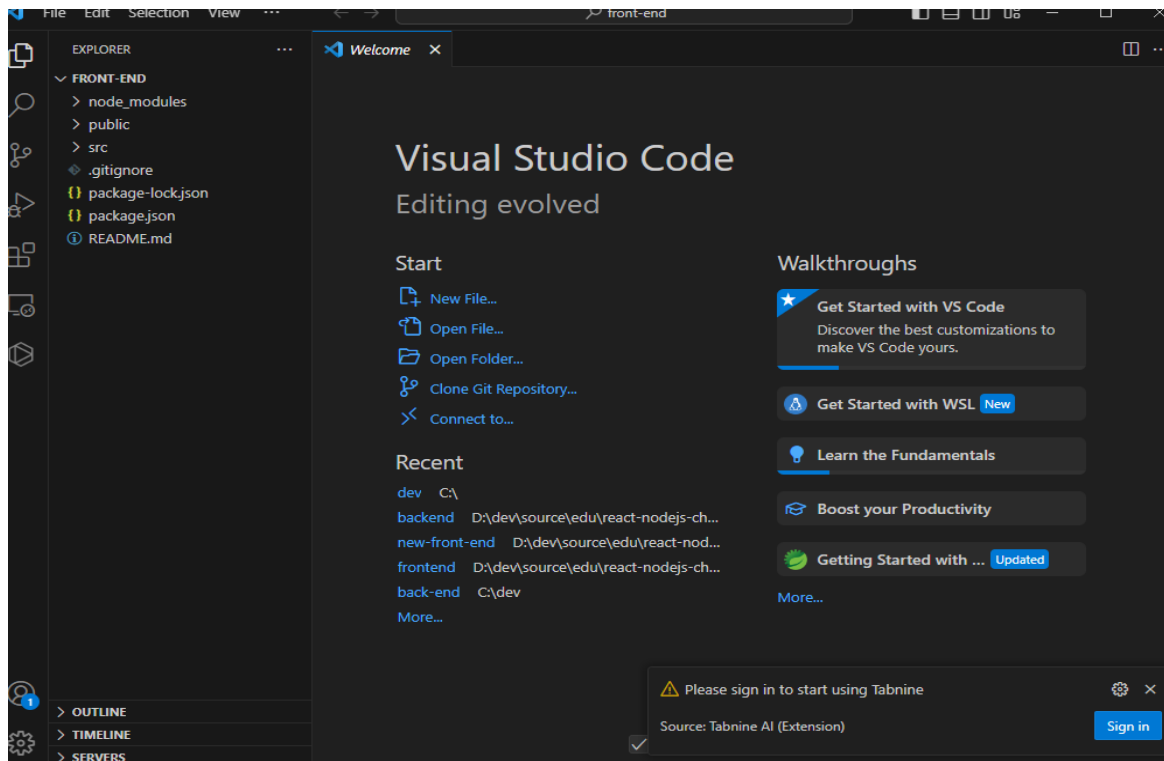
Installing template dependencies using npm...
```

Front-end 작성

■ 환경설정

■ 프로젝트 폴더 오픈

```
C:\dev>cd front-end  
  
C:\dev\front-end>code .
```



Front-end 작성

- 파일 작성
 - src/components 폴더 생성
 - src/components/ChatList.js 파일 생성

```
src > Components > JS ChatList.js > ChatList
1  import styles from "../ChatComponent.module.css";
2
3  export default function ChatList({chats}) {
4    return (
5      <section className={styles.section_container}>
6        {chats && chats.length
7          ? chats.map((chat, index) => (
8            <div key={index} className={chat.role === "user" ? styles.user_msg : styles.assitant_msg}>
9              <span>
10                <b>{chat.role.toUpperCase()}</b>
11              </span>
12              <span>:</span>
13              <span>{chat.content}</span>
14            </div>
15          ))
16          : ""}
17      </section>
18    );
19  }
```

Front-end 작성

■ 파일 작성

■ src/components/ChatComponent.module.css 파일 생성

```
components / # ChatComponent.module.css / .chat_header
.chat_header {
  background-color: #f44336;
}

.section_container {
  background-color: #fce4ec;
  margin: 0.5em;
  height: 80vh;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  overflow: scroll;
  /* justify-content: space-between; */
}

.user_msg {
  width: 50%;
  border: 1px solid #483b52;
  background-color: #3f50b5;
  color: lightgrey;
  align-self: self-end;
  margin-top: 1em;
  padding: 1em;
  border-radius: 10px;
}
```

```
.assitant_msg {
  width: 50%;
  border: 1px solid #8650af;
  background-color: #f44336;
  color: black;
  align-self: self-start;
  margin-top: 1em;
  padding: 1em;
  border-radius: 10px;
}

.enter {
  position: fixed;
  /* height: 4em; */
  bottom: 10px;
  left: 50%;
  transform: translate(-50%, 0);
  border: 1px solid #483b52;
}

.input {
  width: 100%;
  height: 80px;
  padding: 1em;
  border-radius: 10px;
}
```

Front-end 작성

■ 파일 작성

■ src/components/ChatConponent.js 파일 생성 - 1/2

```
import { useState } from "react";
import styles from "../ChatComponent.module.css";
import axios from "axios";
import ChatList from "../ChatList";

export function ChatComponent() {
  const [message, setMessage] = useState("");
  const [chats, setChats] = useState([]);

  const chat = async (e, message) => {
    e.preventDefault();
    if (!message) return;
    setMessage("");
    setChats((prev) => {
      return [...prev, { role: "user", content: message }];
    });
    try {
      const response = await axios.post("http://localhost:8000/", {
        chat: { role: "user", content: message }
      });
      setChats((prev) => {
        return [...prev, response.data.output];
      });
    } catch (e) {
      console.log(e);
    }
  }
}
```

Front-end 작성

- 파일 작성
 - src/components/ChatConponent.js 파일 생성 – 2/2

```
return (  
  <div>  
    <h1 className={styles.chat_header}>chat gpt tutorial</h1>  
    <ChatList chats={chats}></ChatList>  
    <form action="" onSubmit={(e) => chat(e, message)}  
      className={styles.enter}>  
      <input  
        type="text"  
        name="message"  
        value={message}  
        className = {styles.input}  
        placeholder="Type a message here and hit Enter..."  
        onChange={(e) => setMessage(e.target.value)}  
      />  
    </form>  
  </div>  
>);  
}
```


Front-end 작성

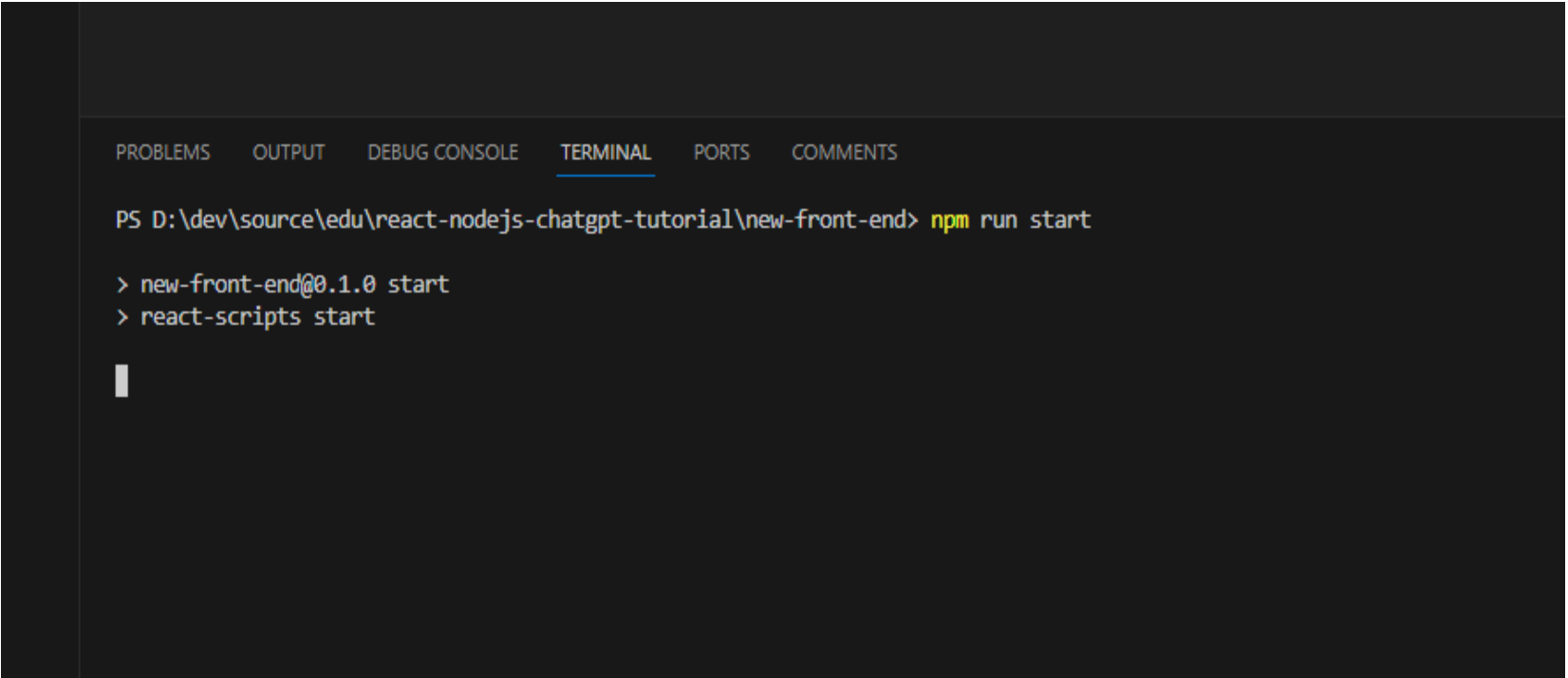
- 파일 수정
 - src/App.js 파일 수정

```
src > JS App.js > ...  
1  import './App.css';  
2  import { ChatComponent } from './components/ChatComponent';  
3  
4  function App() {  
5    return (  
6      <ChatComponent></ChatComponent>  
7    );  
8  }  
9  
10 export default App;  
11
```



Front-end 실행

- 프론트 엔드 **http** 서비스 하기
 - 터미널에서 명령어 실행



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\dev\source\edu\react-nodejs-chatgpt-tutorial\new-front-end> npm run start

> new-front-end@0.1.0 start
> react-scripts start

█
```