SpringBoot Project+Vue Project 개발환경 구축 가이드

1. 시작하기 전에

* 본 가이드는 하나의 프로젝트 내에서 백엔드와 프론트엔드를 분리하여 구성하는 방식입니다.
* 프로젝트 구성 후 개발환경에서 Spring의 백엔드 서버와 Vue의 프론트엔드 서버는 각각 개별로 동작합니다.
* 백엔드는 Eclipse의 maven project, 프론트엔드는 Vue 개발도구인 Vue CLI에서 제공하는 프로젝트 구조를 기반으로 구성합니다.

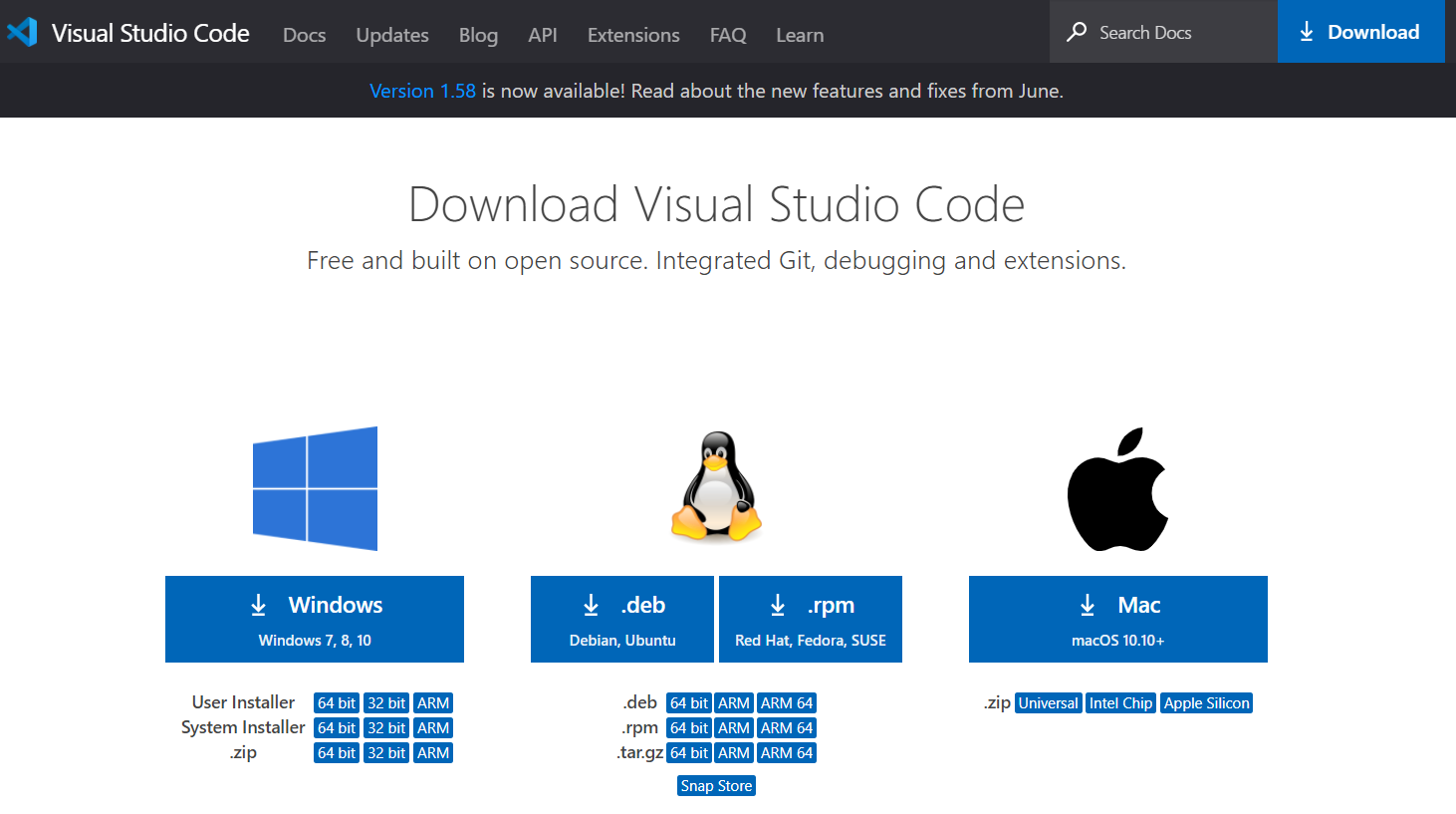
1. 개발환경

|  |  |  |
| --- | --- | --- |
|  | Back-End` | Front-End |
| IDE | Eclipse | Visual Studio Code |
| 언어 | Java (jdk 1.8) | javascript |
| framework | SpringBoot v2.5.3 | Vue.js v2.9.6 |
| server | tomcat 9.0 (SpringBoot 내장) | node.js v14.16.0 |

1. 필요 프로그램 및 라이브러리 설치

3-1. Visual Studio Code(이하 VS Code) 설치

1. VS Code는 무설치형(.zip)도 제공하고 있습니다. 필요에 따라 선택하여 다운로드를 진행합니다. (공식 웹페이지: <https://code.visualstudio.com/Download> )

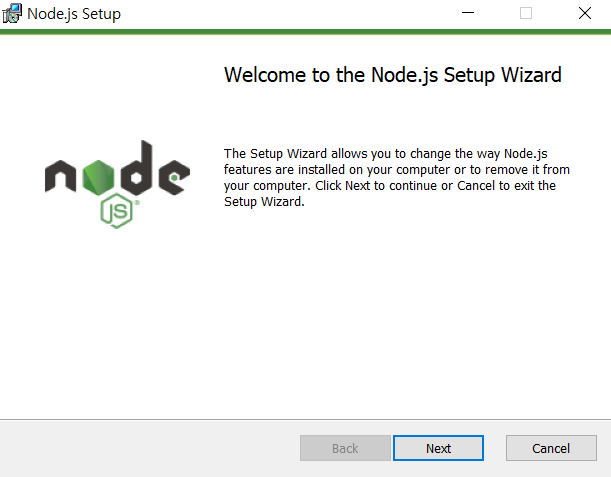


3-2. Node.js 설치

1. 공식 웹페이지에서 가이드 작성일 기준(2021-07-16)으로 안정된 버전인 14.17.3 버전으로 설치하기를 권장합니다. (공식 웹페이지: <https://nodejs.org/ko/> )

Node.js 설치 시 NPM(Node Package Manager)도 함께 설치됩니다.

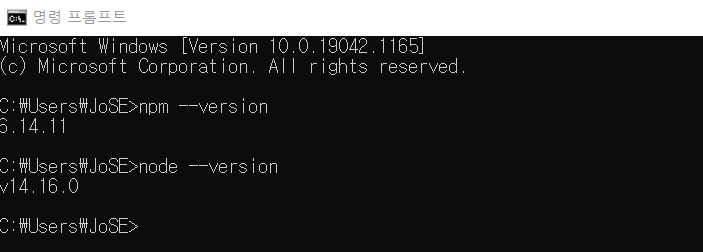




1. Node.js 설치 완료 후 cmd에서 다음의 명령어를 사용하여 node.js 및 npm의 설치 여부를 확인합니다.

**node -–version**

**npm –-version**



3-3. Vue-CLI 설치

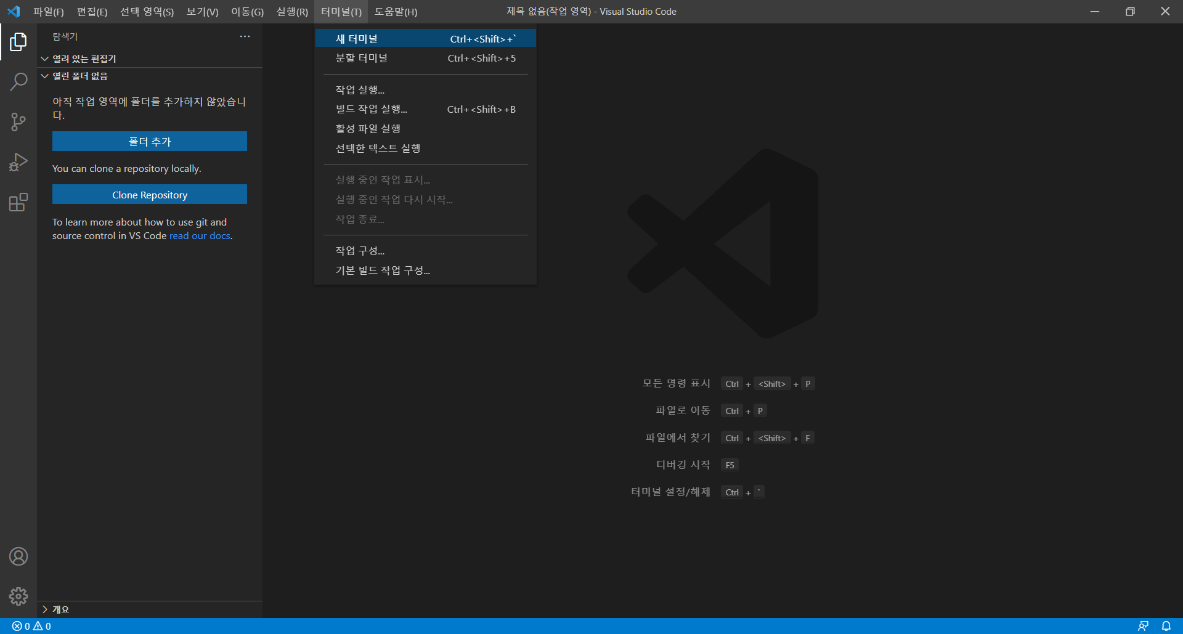
1. VS Code의 터미널 또는 cmd에서 다음의 명령어로 Vue.js 개발도구인 Vue CLI를 설치합니다.

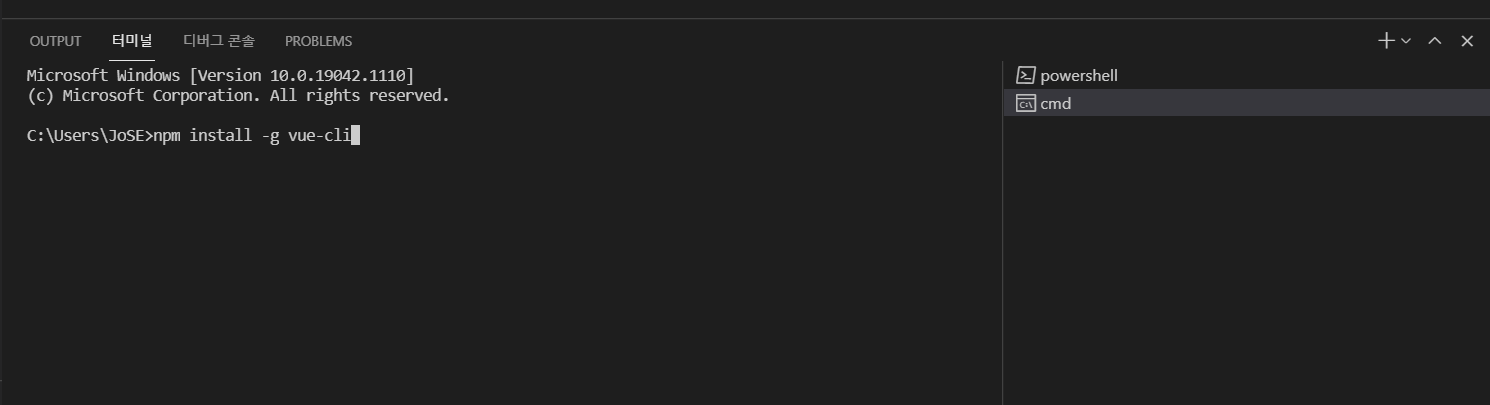
Vue CLI 설치 시 Vue.js도 함께 설치됩니다.

(Vue CLI 2.x.x: Vue.js 2.x.x, Vue CLI 3.x.x 이상: Vue.js 3.x.x 이상)

**npm install -g vue-cli** \* ‘-g‘는 전역으로 설치함을 의미합니다.

[VS Code 터미널 예시]





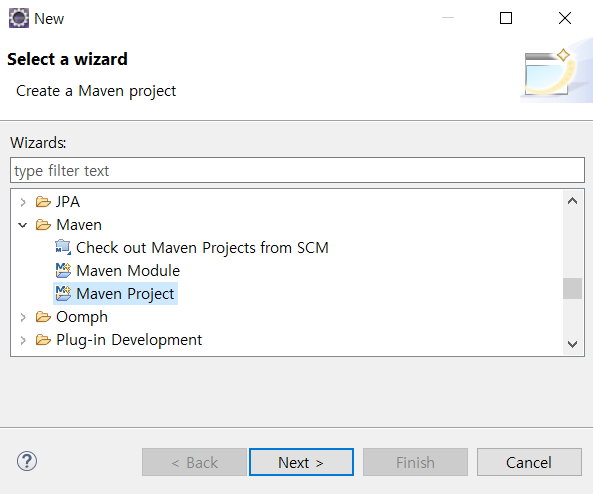
1. Vue CLI 설치 완료 후 node.js와 동일한 방식으로 vue.js의 설치 여부를 확인합니다.

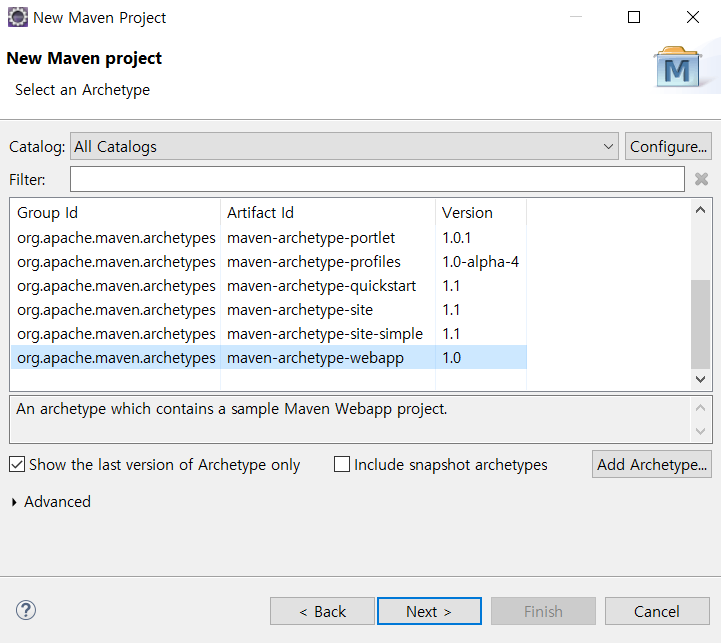


1. 프로젝트 구성

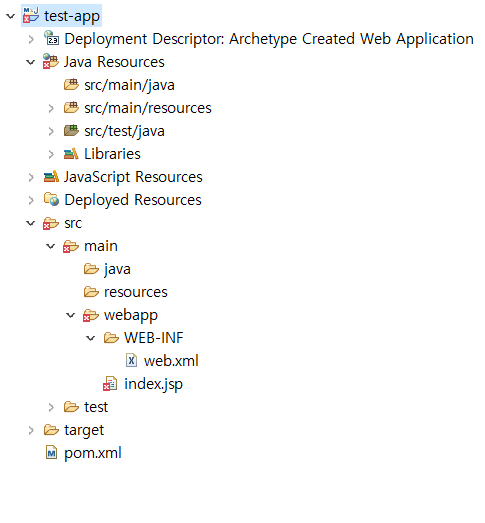
4-1. 백엔드(SpringBoot) 프로젝트 구성

1. Eclipse에서 maven 프로젝트(webapp)를 생성합니다.

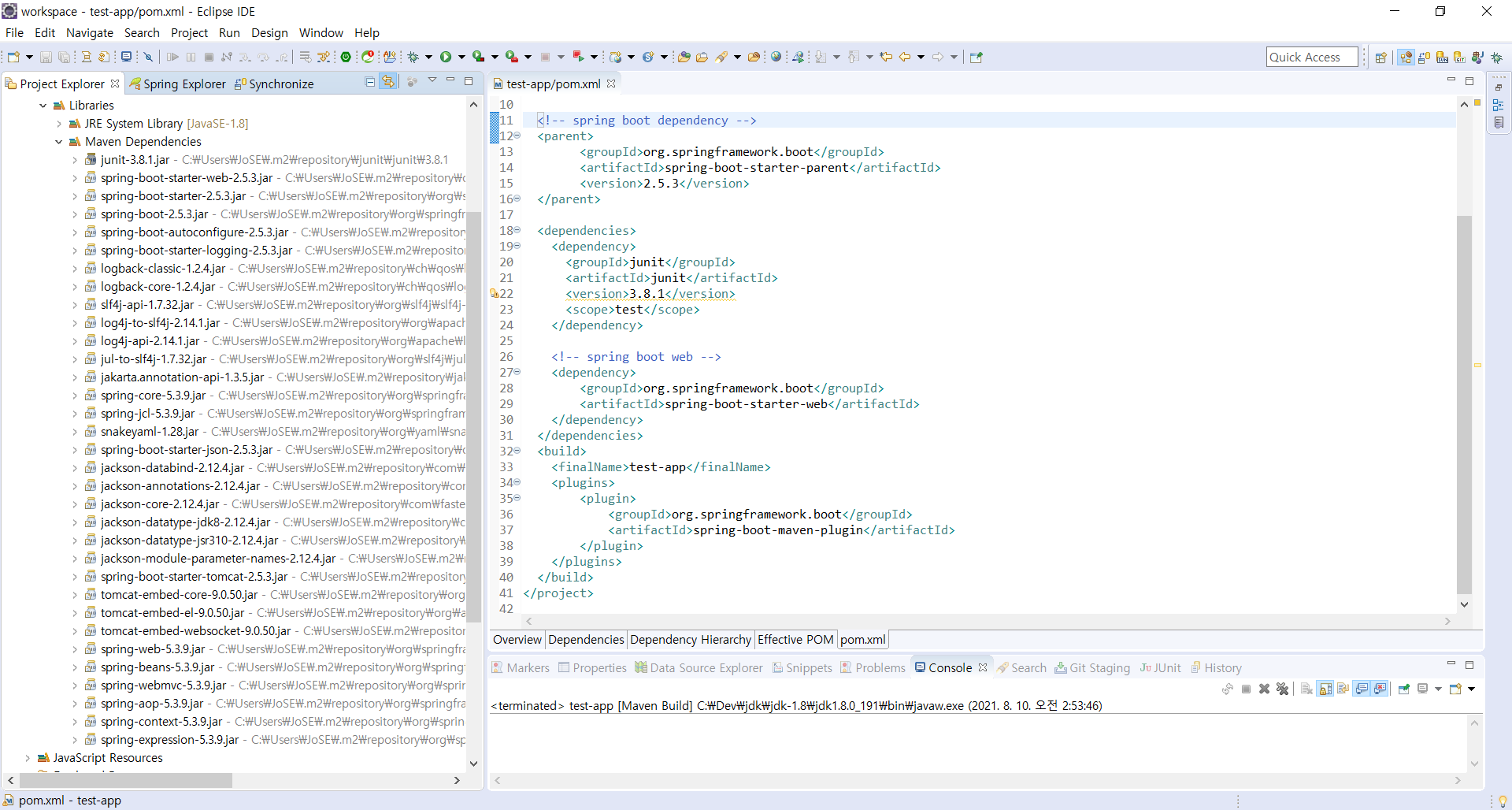




프로젝트 생성 직후의 기본 구조는 다음과 같습니다.



1. pom.xml 파일에 SpringBoot framework 사용을 위한 dependency를 추가합니다.



<parent> … spring-boot-starter-parent </parent>

* SpringBoot의 의존성 패키지인 starter에 대해 충돌 없는 버전 정보까지 통합으로 제공합니다.

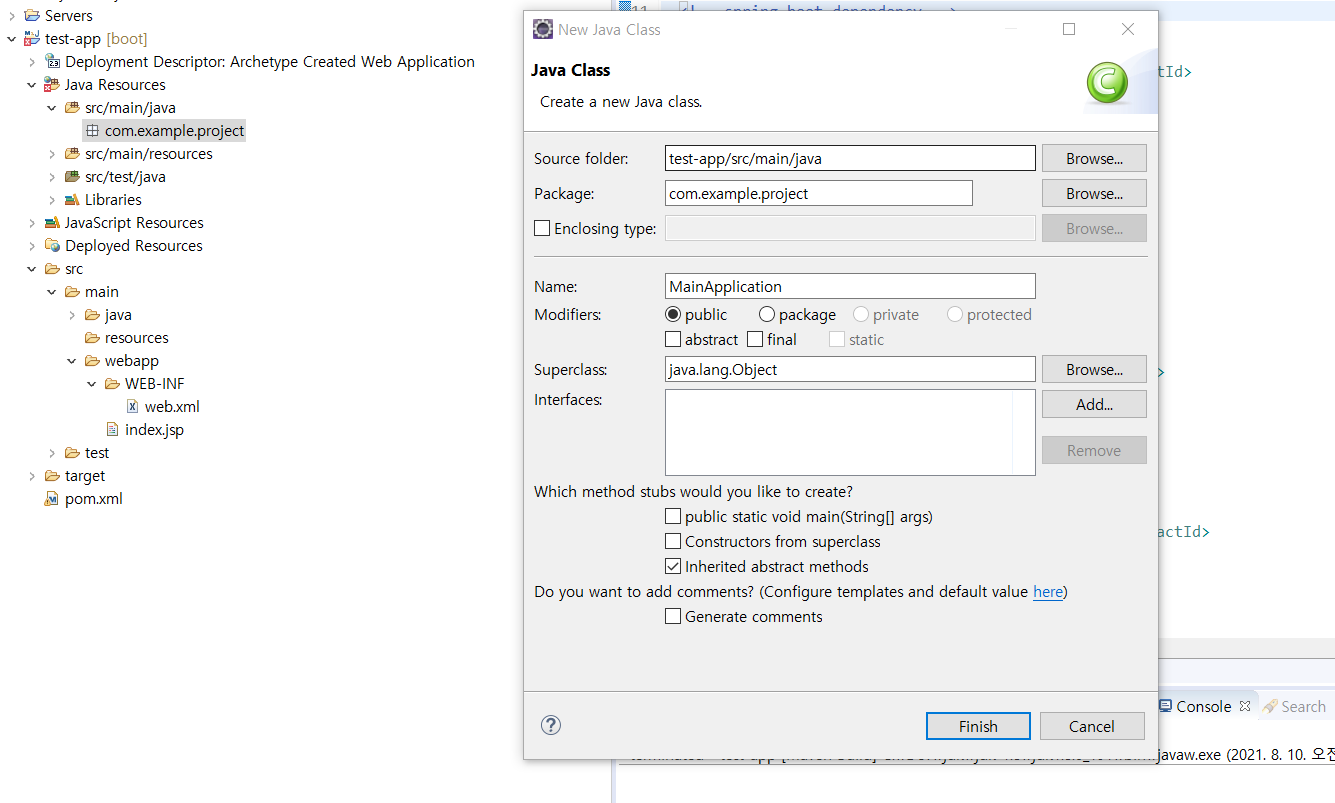
<dependency> … spring-boot-starter-web </dependency>

* spring-web, spring-mvc 등 웹 관련 의존성 패키지 입니다.

<plugin> … spring-boot-maven-plugin </plugin>

* SpringBoot에서 독립적으로 실행 가능한 .jar 파일을 만들기 위한 plugin 입니다.

1. SpringBoot 프로젝트 실행을 위해 가장 기본 패키지를 만들고 Main 역할의 Controller를 생성합니다.



1. Main Controller에 아래의 코드를 작성합니다.

@SpringBootApplication

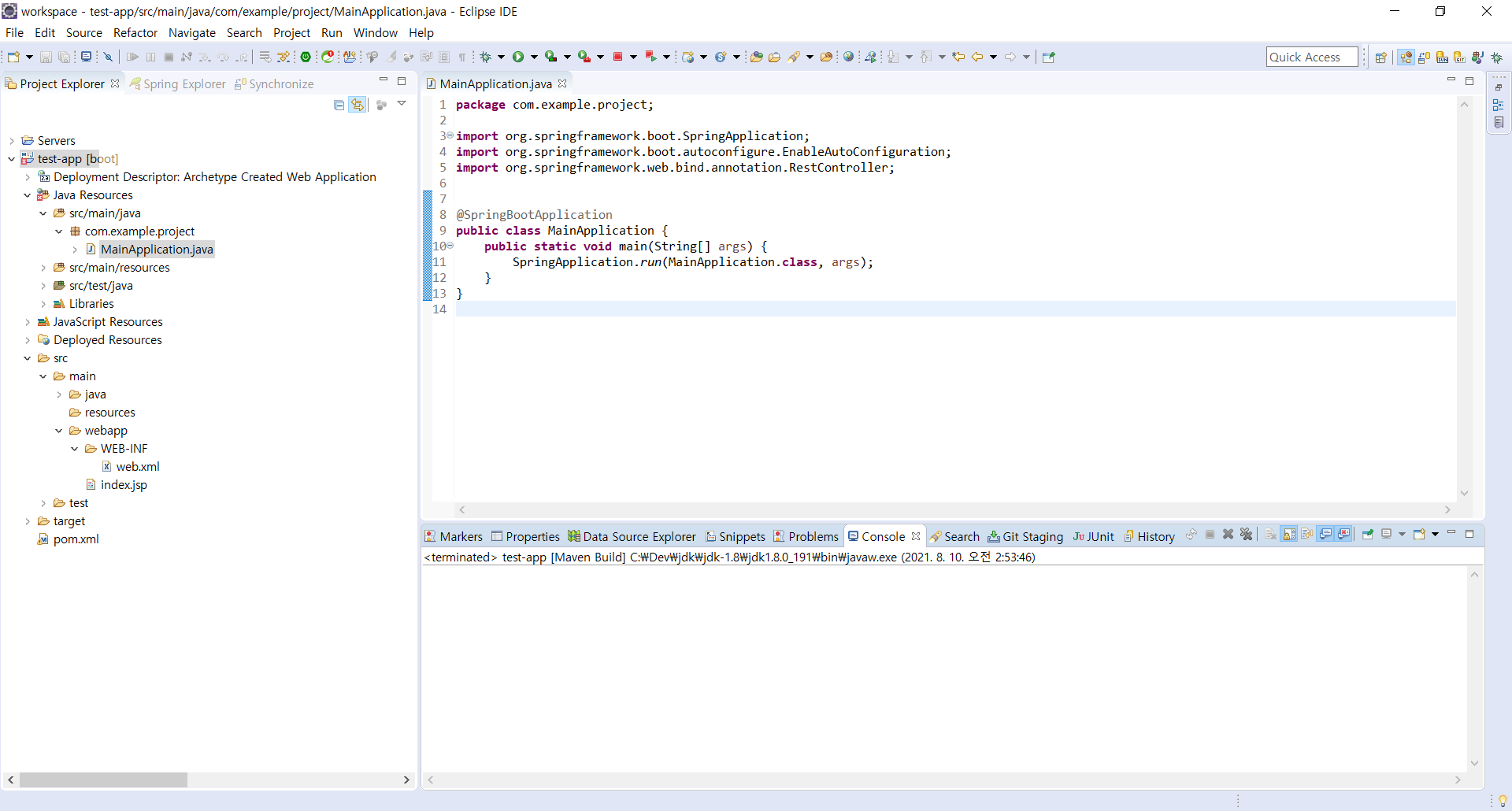
public class MainApplication {

public static void main(String[] args) {

SpringApplication.run(해당Controller파일명.class, args);

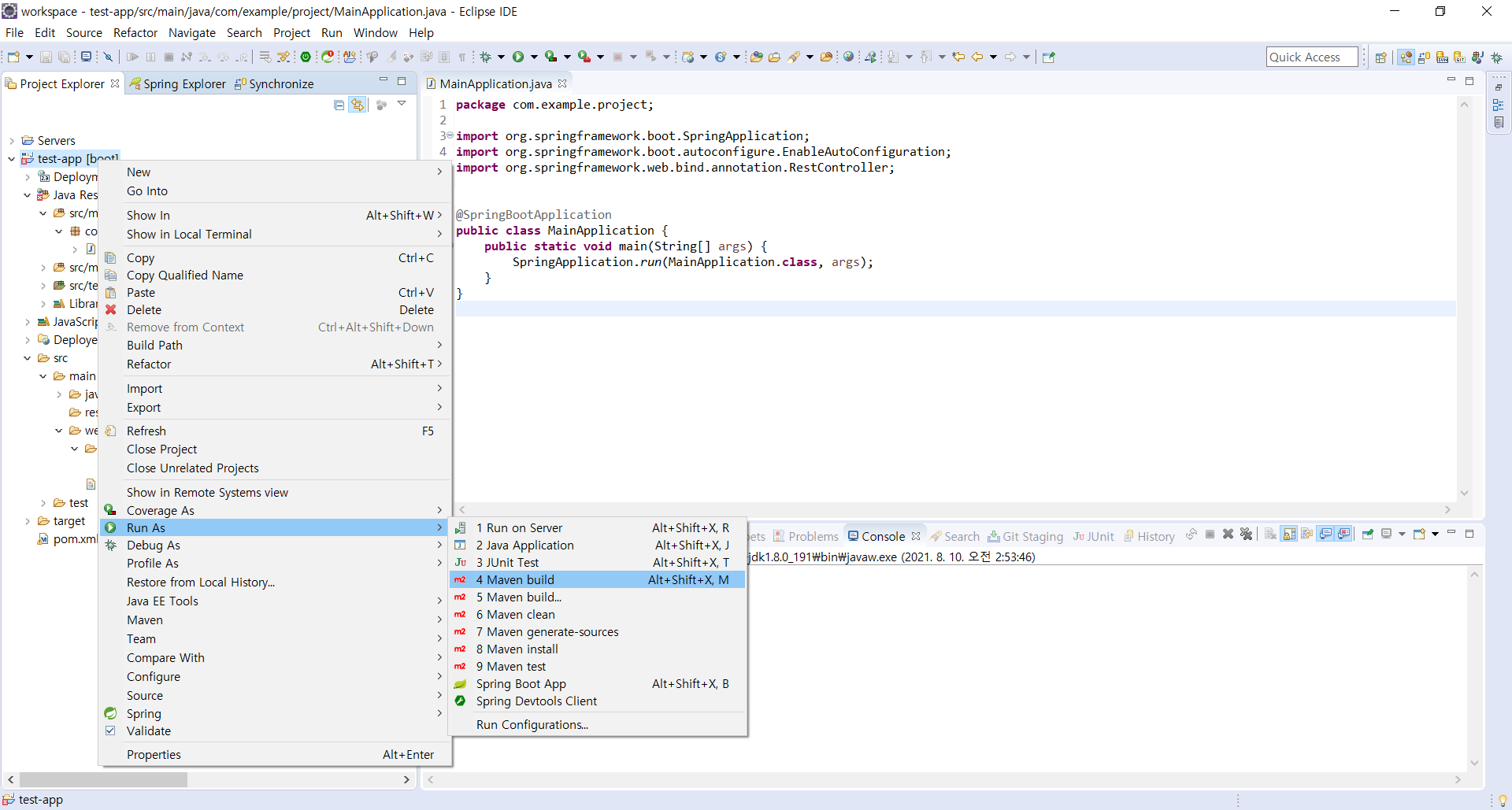
}

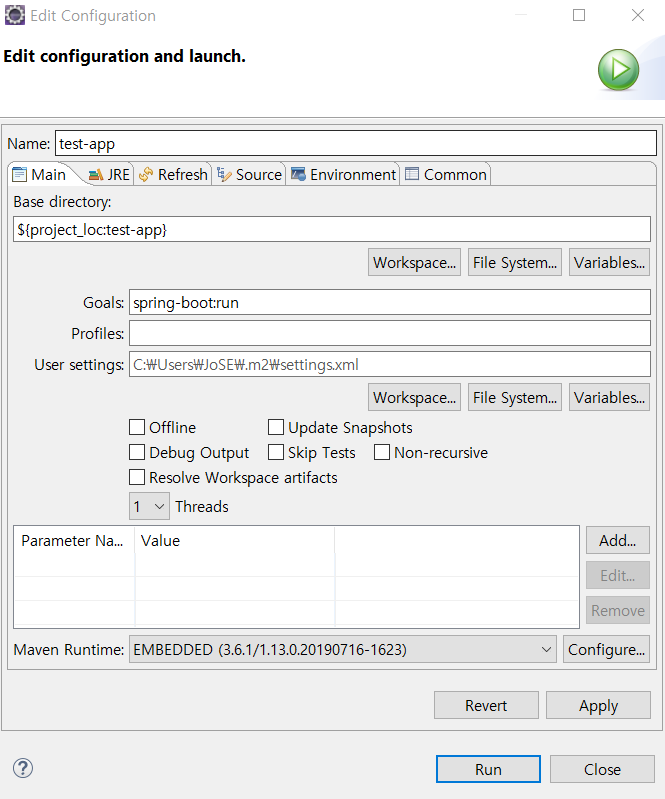
}



1. 프로젝트 실행을 위해 Maven build 설정을 추가합니다.

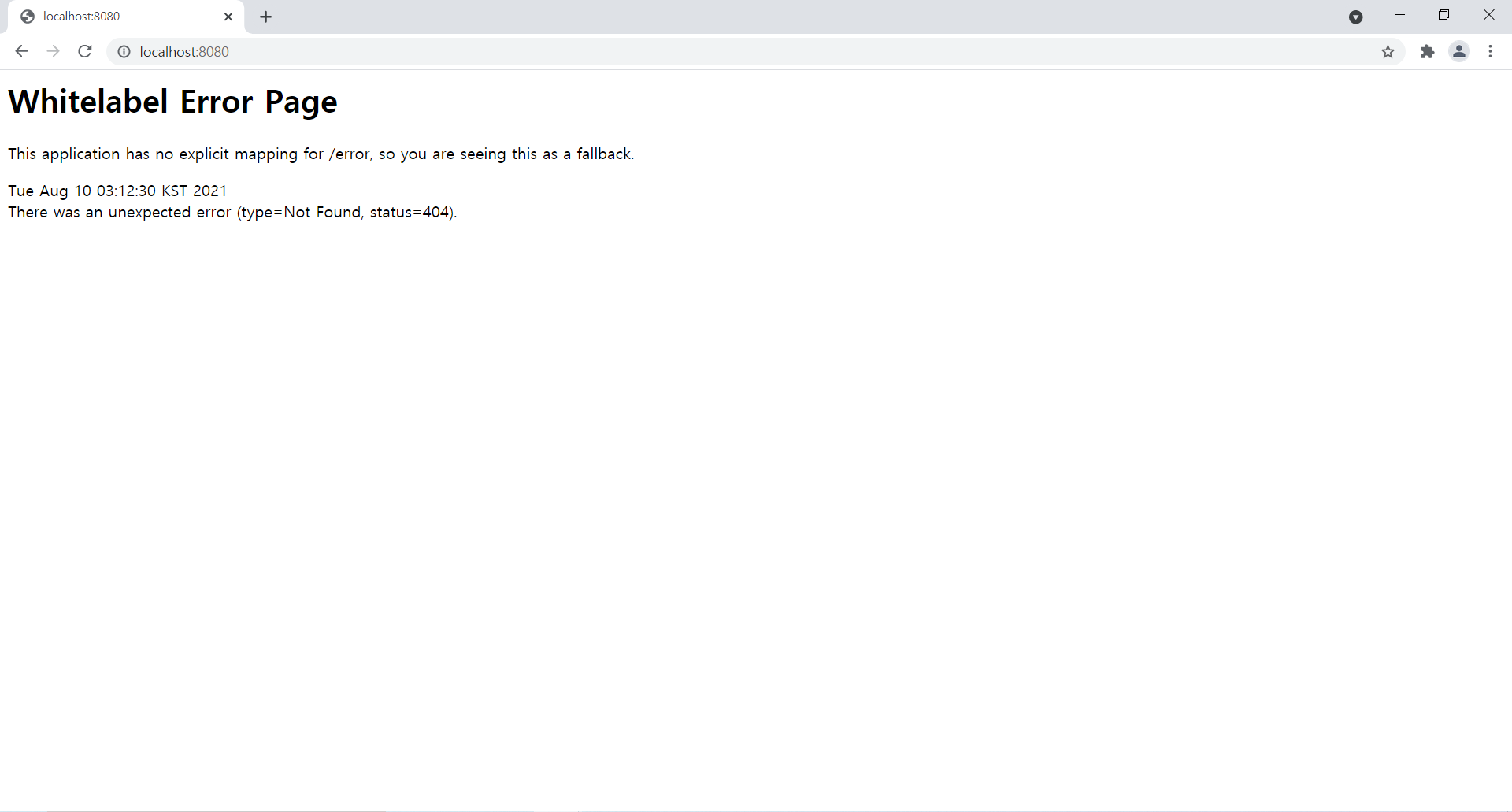
프로젝트 우클릭 🡪 Run As 🡪 Run Configurations … 🡪 spring-boot:run 입력 🡪 Run





이후부터는 ‘Run As 🡪 Maven Build’로 실행합니다.

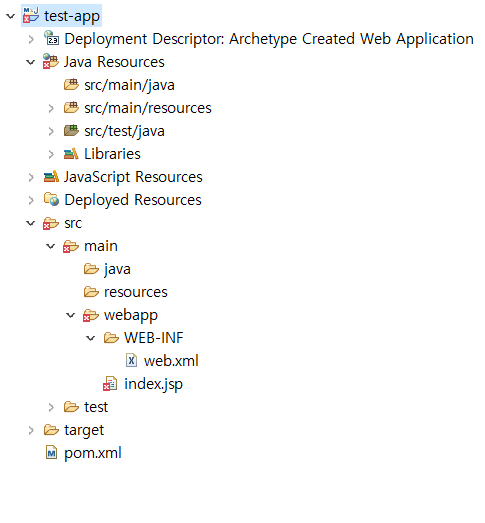
1. 프로젝트 실행 및 접속 여부를 확인합니다. 정상적으로 실행된 경우 아래의 화면이 표시됩니다.



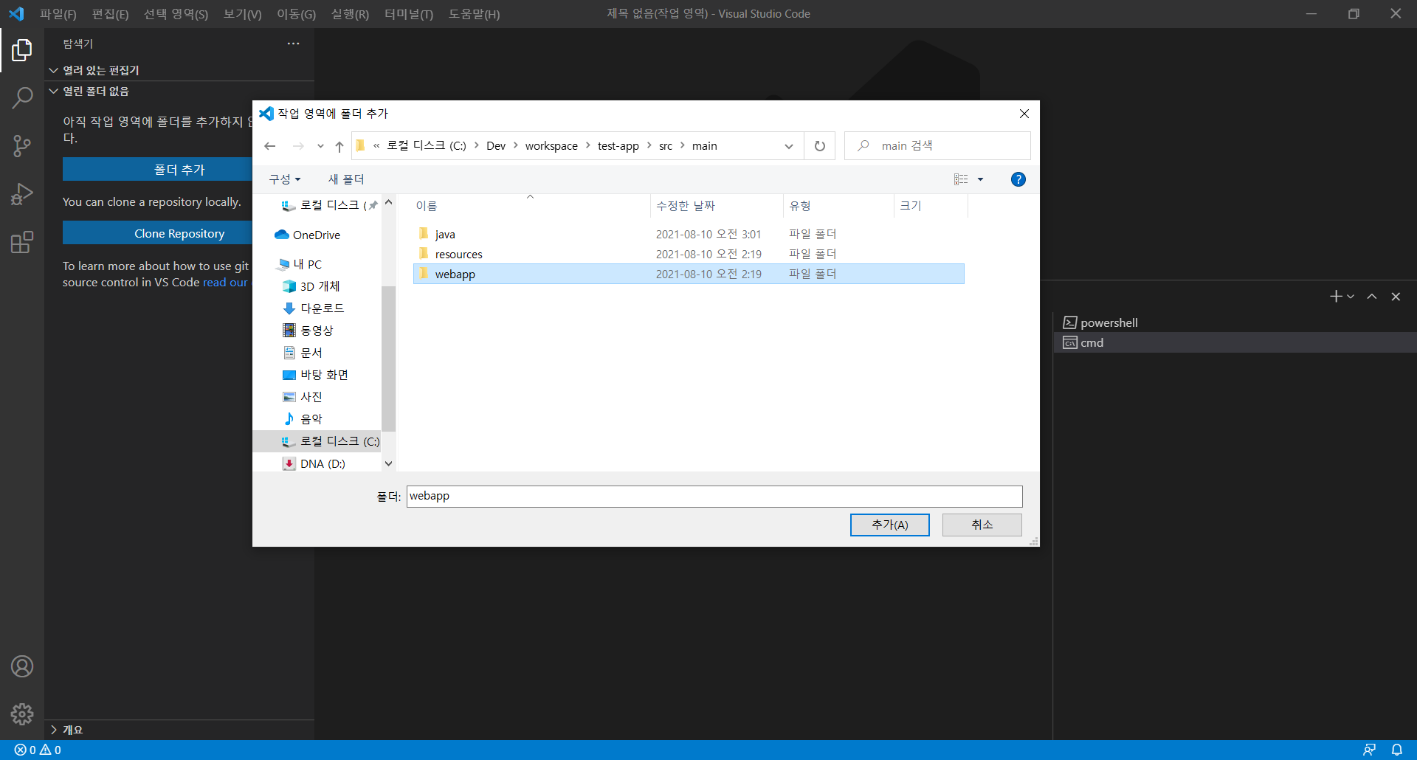
4-2. 프론트엔드(Vue.js) 프로젝트 구성

1. VS Code의 작업영역에서 ‘폴더 추가’를 클릭하여 4-1에서 생성했던 Java 프로젝트를 추가합니다. Java 파일을 열어볼 일이 없다면, Vue 프로젝트를 생성할 위치로 추가합니다.

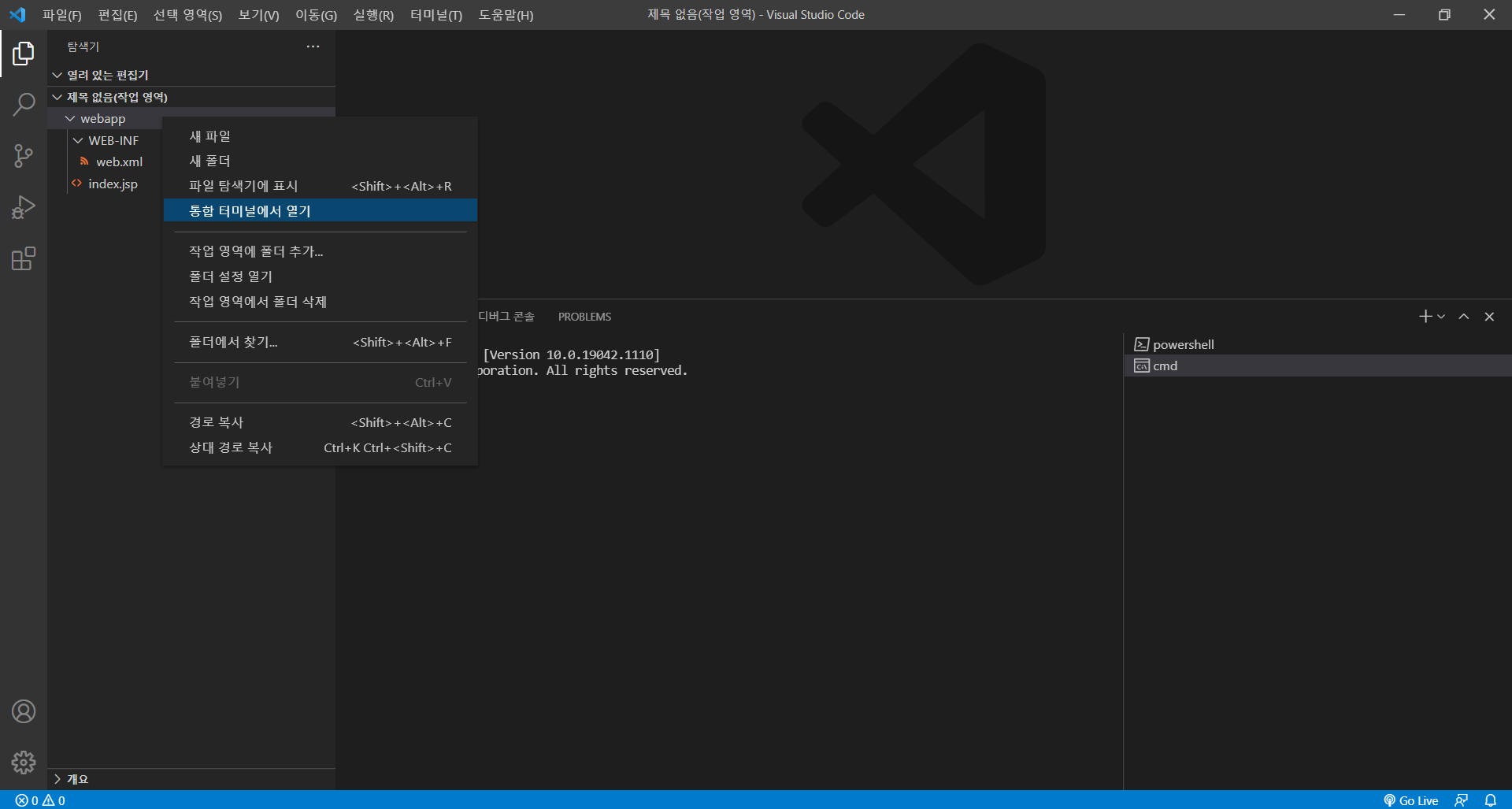
[4-1 에서 만들었던 백엔드 프로젝트 기본 구조]



* 본 가이드 에서는 webapp 아래에 Vue 프로젝트를 생성합니다.



1. 작업 영역에서 Vue 프로젝트를 생성할 위치에서 ‘우클릭 🡪 통합 터미널에서 열기’를 해당 위치의 터미널을 열어줍니다.



1. 해당 위치의 터미널에서 다음의 명령어를 사용하여 Vue 프로젝트를 생성합니다.

**vue init webpack <project-name>**

가이드 예시: vue init webpack vue-front

Vue 프로젝트 생성 시 프로젝트 정보 및 기본 설치 모듈에 대한 설정을 선택합니다.

* ***Project name***

: 앞서 입력했던 ‘vue init webpack 프로젝트명’의 ‘프로젝트명’이 기본값으로 설정됩니다. 변경을 원하지 않으면 Enter 키를 눌러 넘어갑니다.

* ***Project Description***

: 프로젝트에 대한 설명을 입력합니다. 기본값은 ‘A Vue.js project’ 입니다. 변경을 원하지 않으면 Enter 키를 눌러 넘어갑니다.

* ***Author***

: 작성자를 입력합니다. 기본값은 없습니다. 입력을 원하지 않으면 Enter 키를 눌러 넘어갑니다.

* ***Vue Build***

: Vue Build를 선택합니다. Runtime+Compiler/Runtime-only 중 선택하며, 대부분의 사용자에게 권장되고 있는 것은 Runtime+Compiler 입니다. 빌드와 관련된 보다 자세한 내용은 Vue.js 공식 문서에서 확인할 수 있습니다. (https://kr.vuejs.org/v2/guide/installation.html#각-다른-빌드간-차이점)

* ***Install vue-router? (Y/n)***

: vue-router의 설치 여부를 선택합니다. Vue 프로젝트에서의 페이지 이동을 위한 필수 요소이므로 Y를 선택합니다.

* ***Use ESLint to lint your code? (Y/n)***

: 코드를 분석하여 문법 오류 등을 표시하는 Lint도구인 ESLint의 사용 여부를 선택합니다.

* ***Pick an ESLint preset***

: 위의 ESLint 사용(Y)을 선택했을 때 설정할 수 있는 것JavaScript 스타일 규칙입니다. Standard/Airbnb/none(configure it yourself) 중 선택합니다. 기본값은 Standard입니다.

* ***Set up unit tests (Y/n)***

: 단위 테스트의 사용 여부를 선택합니다.

* ***Pick a test runner***

: 테스트 도구를 선택합니다. jest/Karma and Mocha/none(configure yourself) 중 선택합니다.

* ***Set up e2e tests with Nightwatch? (Y/n)***

: e2e(End-to-End) 테스트 도구인 Nightwatch의 사용 여부를 선택합니다.

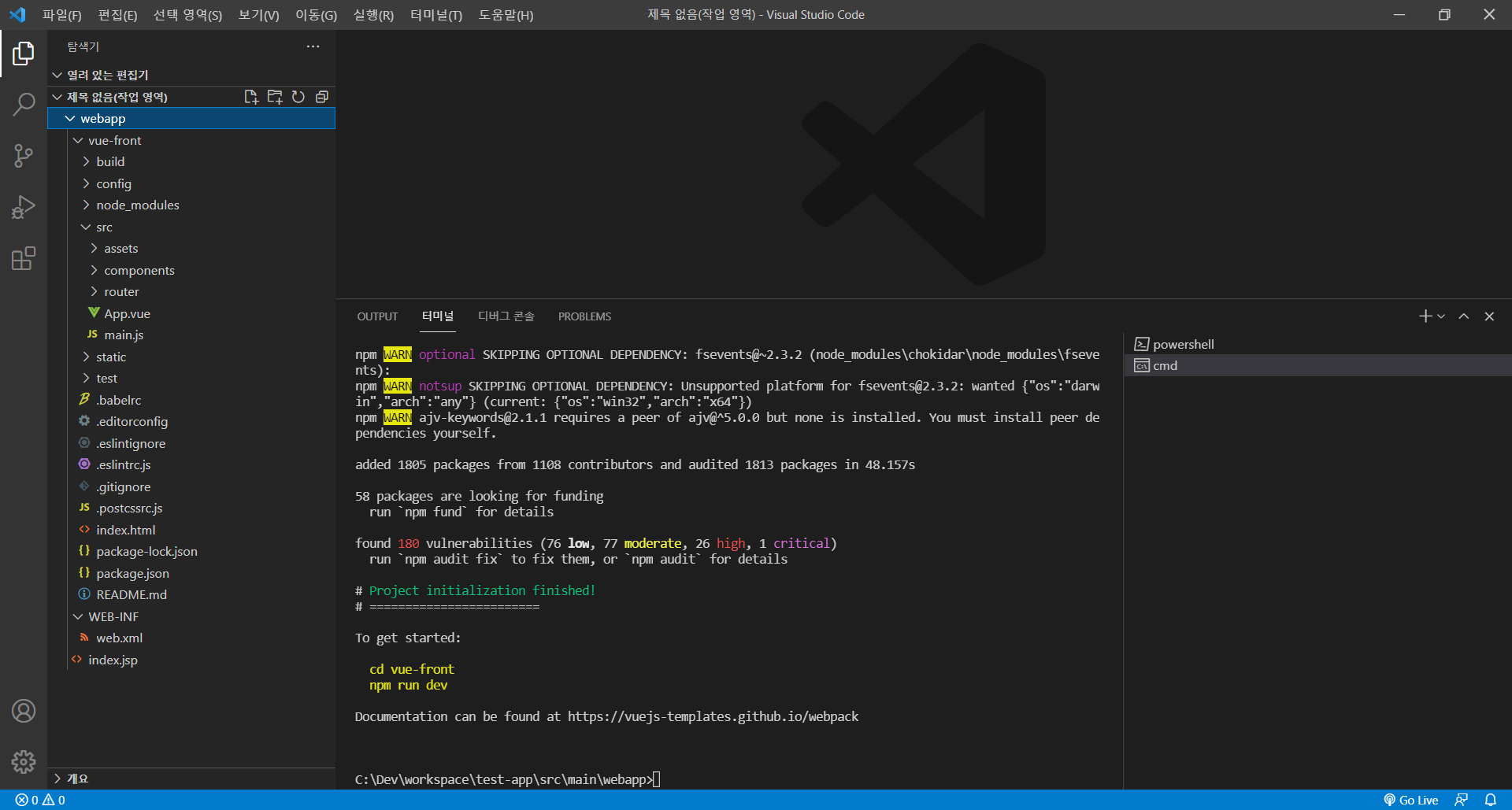
\*e2e테스트: 사용자가 사용하는 상황을 가정하여 처음부터 끝까지 전체 시스템을 테스트하는 것.

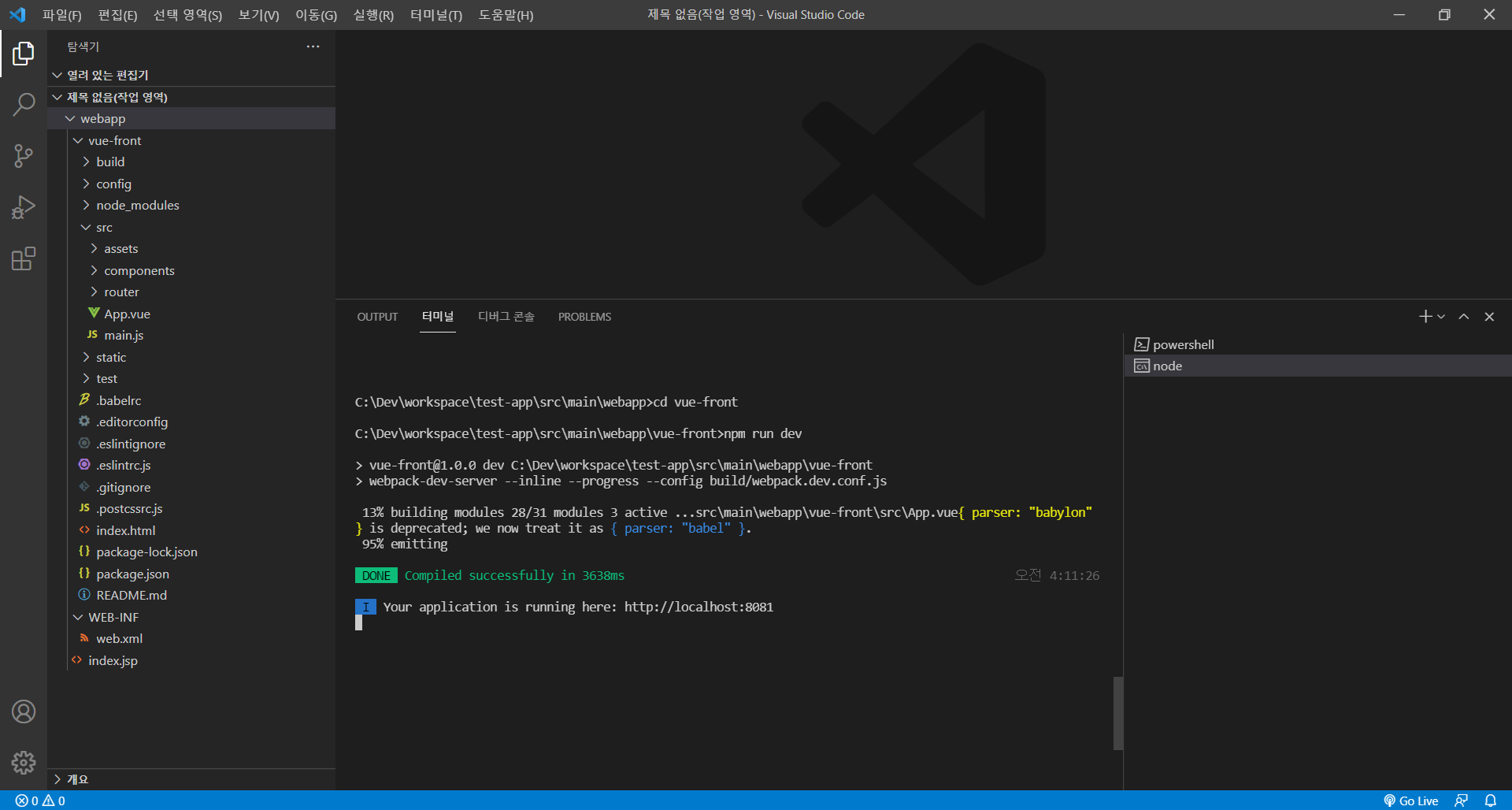
* ***Should we run ‘npm install’ for you after the project has been created? (recommended) (Use arrow keys)***

: 프로젝트 생성 후 ‘npm install’을 실행하여 node\_modules에 dependency 설치 여부를 선택합니다. Yes, use NPM/Yes, use Yarn/No, I will handle that myself 중 선택합니다.

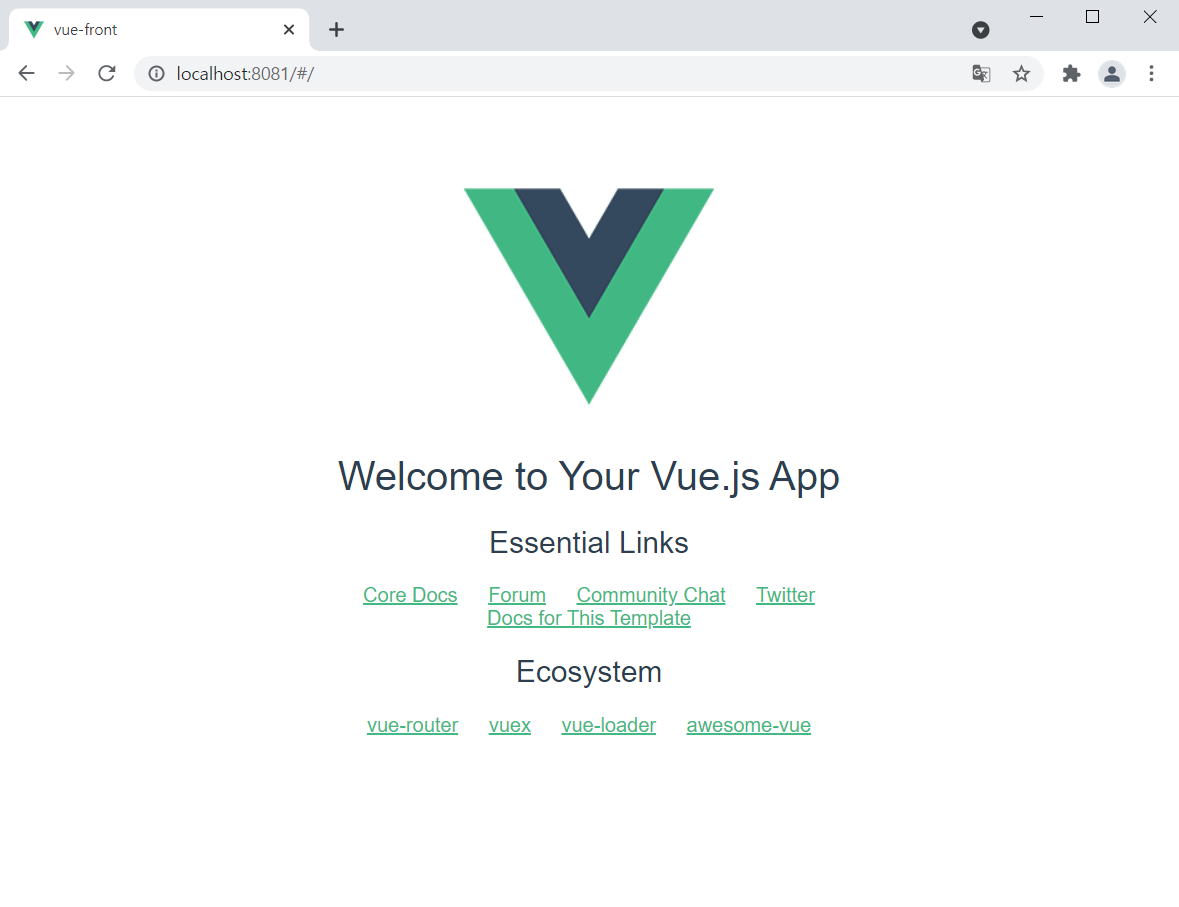
1. Vue 프로젝트 생성 후 안내되는 명령어 ‘**cd 프로젝트명’**에 따라 해당 위치로 이동하고 **npm run dev** 명령어로 Vue 프로젝트를 실행합니다.

이후 Vue 프로젝트 개발 시 작업에 필요한 npm 모듈 설치, 실행 등 명령어는 모두 현재 이동한 위치에서 실행하게 됩니다.





1. 프로젝트 실행 및 접속 여부를 확인합니다. 정상적으로 실행된 경우 아래의 화면이 표시됩니다.



1. 프로젝트 추가 설정

5-1. 서버 포트 변경

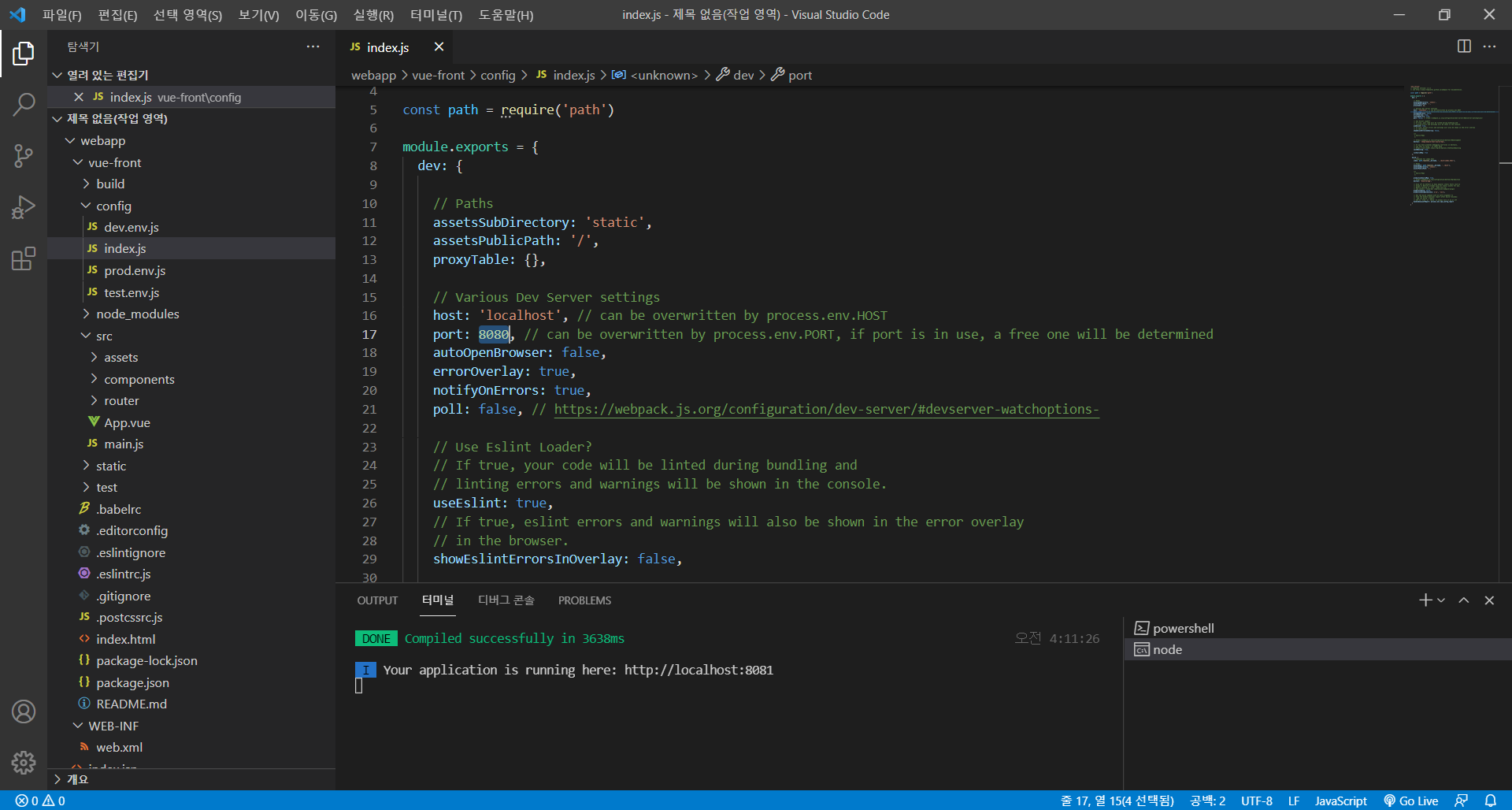
포트 설정을 변경하지 않았다면 기본적으로 백엔드와 프론트엔드 모두 8080 포트가 기본 설정입니다. 백엔드 서버를 먼저 실행하고, 프론트엔드를 나중에 실행할 경우 프론트엔드 측 서버 포트번호가 자동으로 8081 등 다른 번호로 변경되기도 합니다.

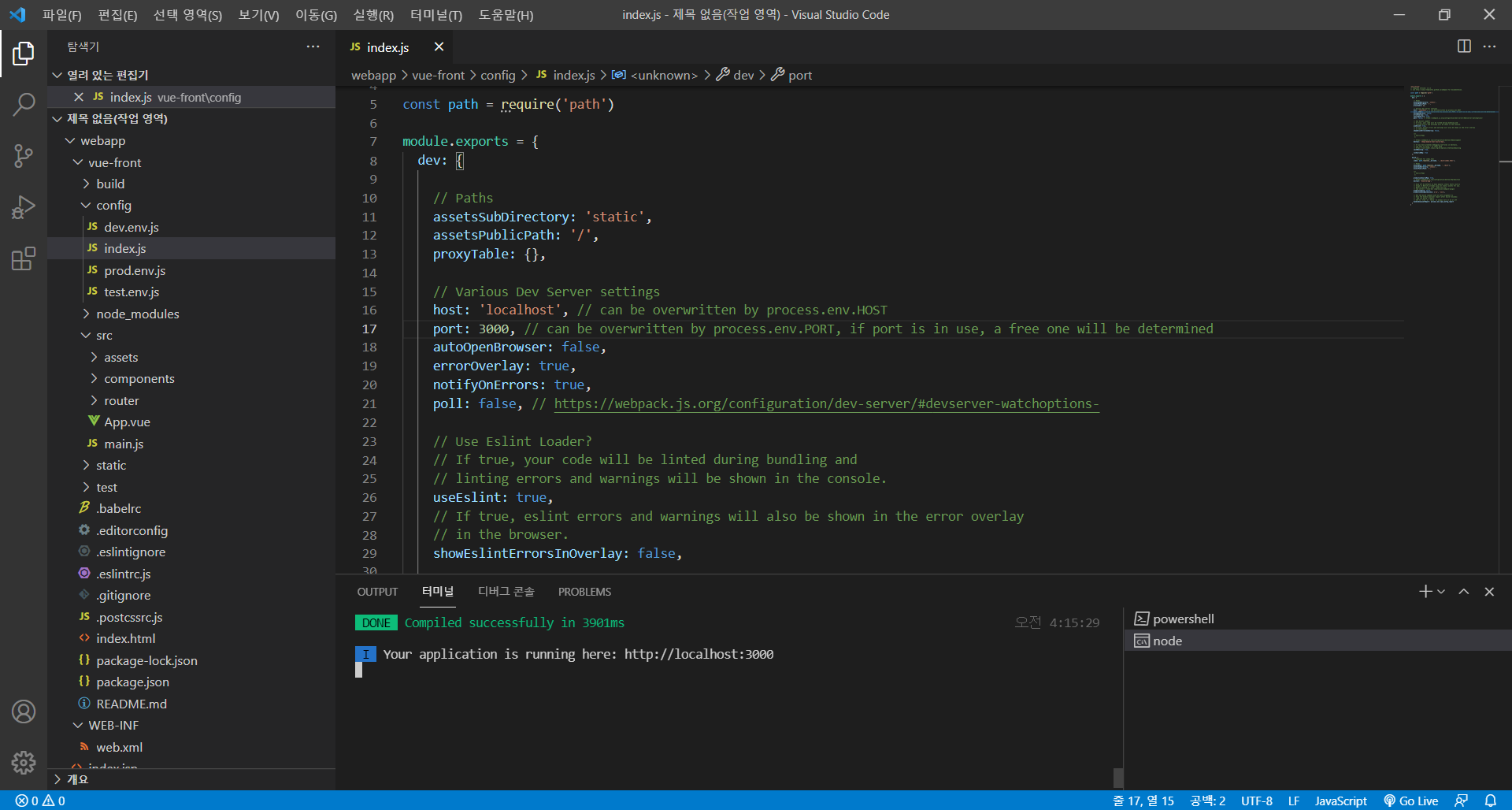
그러나 기본적인 설정이 8080이므로 포트 충돌을 방지하기 위해 백엔드 또는 프론트엔드의 포트를 변경해주어야 합니다.

본 가이드에서는 프론트엔드 측의 포트를 변경합니다.

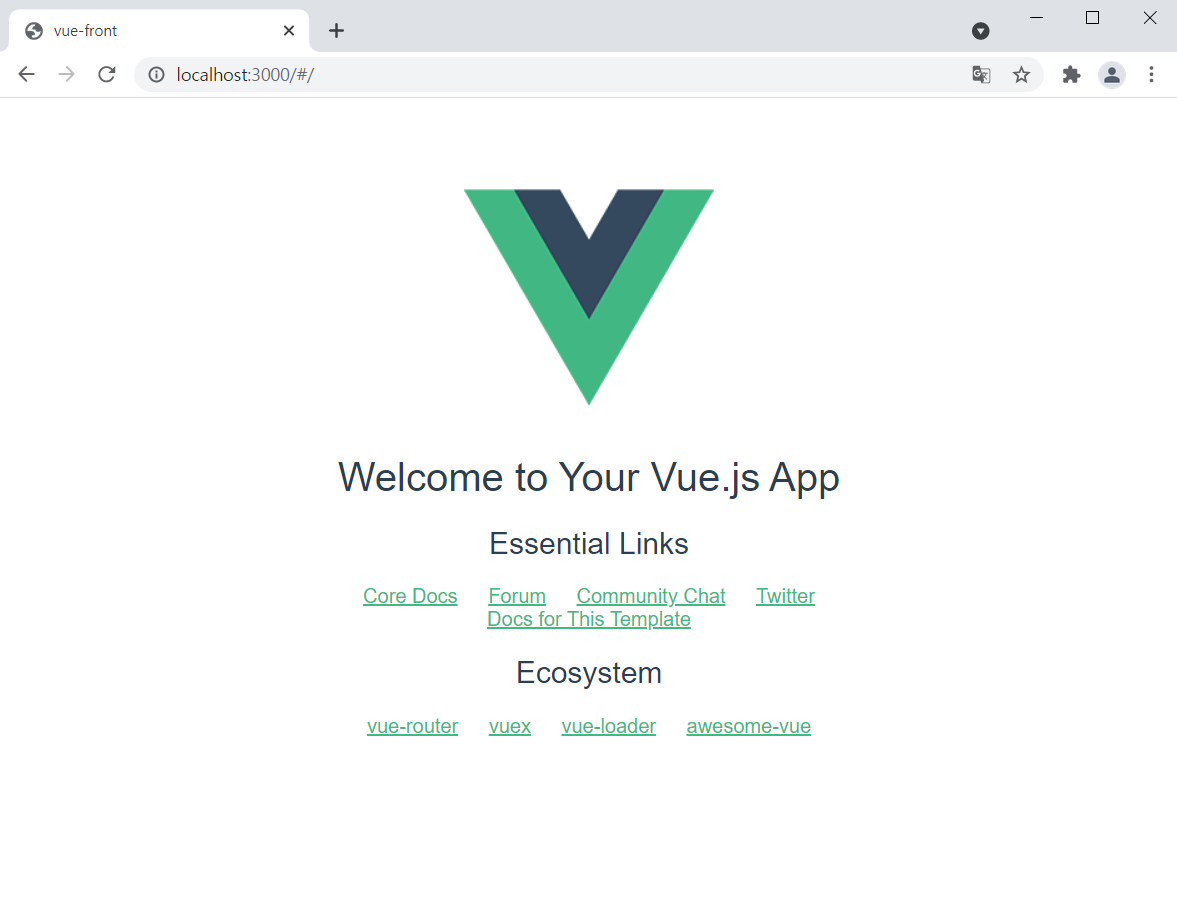
1. Vue 프로젝트 🡪 config 🡪 index.js

Vue 프로젝트의 설정 파일입니다. dev 🡪 port 부분을 수정합니다.





1. 포트 수정 후 **npm run dev** 명령어를 사용하여 Vue 프로젝트를 재실행하고 변경된 포트로 실행 및 접속 여부를 확인합니다. 정상적으로 실행된 경우 아래의 화면이 표시됩니다.



5-2. 추가 설치 모듈(라이브러리)

Vue 프로젝트 개발 시 필수처럼 여겨지는 모듈이 몇 가지 있습니다. Vue 프로젝트 구성 후 추가로 설치합니다.

1. Vue Router

Vue 프로젝트 내에서 컴포넌트에 주소를 매핑하는 라이브러리입니다. 주소 값에 따라 매핑된 Vue 컴포넌트 페이지로 이동합니다.

Vue CLI를 통해 Vue 프로젝트를 구성한 경우 설정에 따라 함께 설치되기도 합니다.

**npm install vue-router**

1. Axios

Vue 프로젝트에서 가장 대중적으로 사용되는 비동기 통신 라이브러리입니다. 실제 데이터와 통신하는 백엔드 서버 주소에 연결하기 위해 사용합니다.

**npm install axios**

1. Vuex

‘상태 관리 패턴’ 라이브러리라고 합니다. 여기서 상태란 컴포넌트 사이에서 주고 받는 데이터를 의미합니다. Vuex는 애플리케이션의 모든 컴포넌트에 대한 중앙 저장소 역할을 합니다. 중대형 이상의 프로젝트, 여러 컴포넌트가 동일한 데이터를 공유하는 경우 해당 라이브러리를 사용하도록 권고하고 있습니다.

**npm install vuex**