

3. eslint

1) eslint란?

오타 같은 실수에서 컴파일 에러가 발생해야 버그 찾기가 수월하다.

그런데 javascript는 인터프리터 방식의 언어라서 컴파일을 하지 않는다.
컴파일을 하지 않기 때문에, 컴파일 에러가 발생하지도 않는다.

소스코드에 오타가 있어도, 컴파일 에러가 발생하지 않고,
실행할 때 런타임 에러(runtime error)가 발생한다.

에러를 찾기 위해, 모든 기능을 다 실행해 보는 것은 힘들기 때문에,
컴파일 방식 언어보다 인터프리터 방식 언어는 버그 찾기가 힘들다.

그래서 인터프리터 언어는(스크립트 언어는) 큰 프로젝트에 적당하지 않고,
간단한 기능을 빠르게 개발하기에 적당하다.

이 불편함을 보완하는 도구가 eslint 이다.

eslint는 컴파일러처럼 소스코드를 분석해서 오타와 같은 문법 오류를 찾아준다.
바람직하지 않은 코딩 스타일도 지적해 준다.

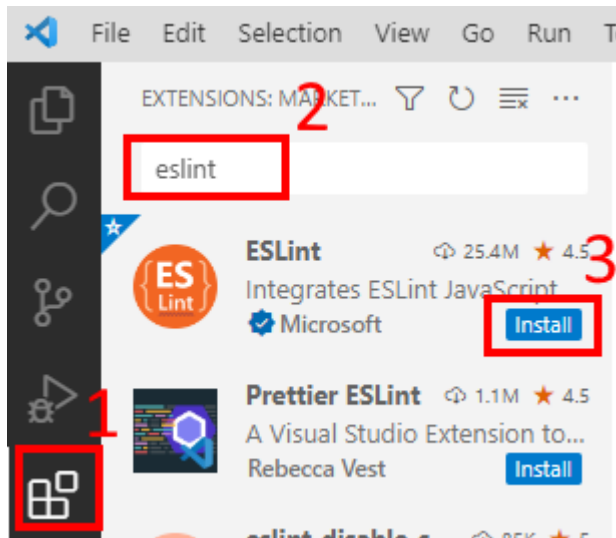
2) eslint 설치

eslint 설치 명령

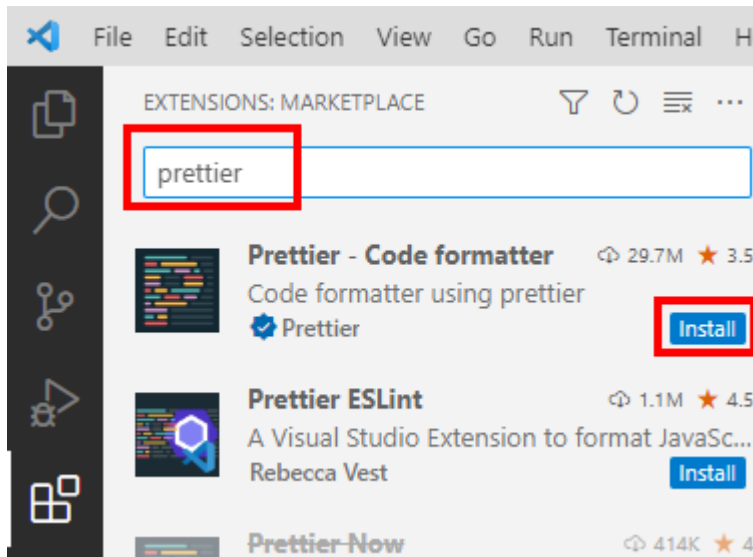
```
npm install -g eslint
```

3) visual studio code 확장 설치

visual studio code에 eslint 확장 설치



Prettier - Code formatter 확장 설치



node.js 프로젝트 설정 파일

4) package.json 파일 생성

node.js 프로젝트의 설정 파일은 package.json 파일이다.
이 파일이 프로젝트 폴더에 있어야 한다.

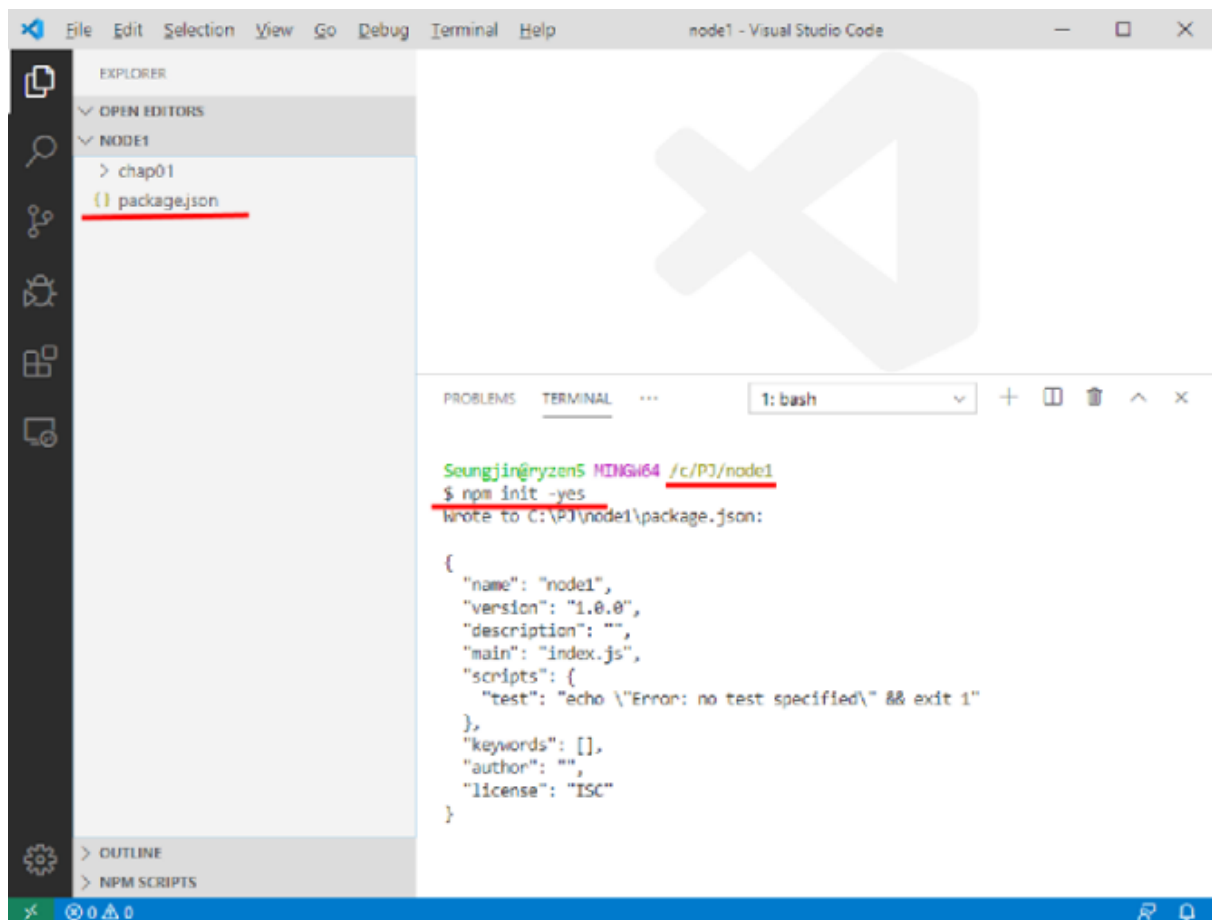
프로젝트 디렉토리에 package.json 파일을 생성하자.

package.json 생성

프로젝트 디렉토리에서 아래 명령을 실행해야 한다.

```
npm init -yes
```

node1 프로젝트 폴더에 package.json 파일이 생성된다.



5) eslint 설정 파일 생성

프로젝트 폴더 에 eslint 설정 파일을 추가하자.
eslint 설정 파일은 .eslintrc 이다.

프로젝트에 eslint 설정 파일(.eslintrc) 추가 명령

```
# 주의: node1 프로젝트 폴더에서 아래 명령을 실행해야 한다.  
npm init @eslint/config
```

y 엔터

```
user@DESKTOP-CKJ18T9 MINGW64 /d/mk/node1  
○ $ npm init @eslint/config  
Need to install the following packages:  
  @eslint/create-config  
Ok to proceed? (y) █
```

위 아래 화살표 키를 눌러서 'To check syntax and find problems'
항목을 선택하고 엔터키 입력

```
? How would you like to use ESLint? ...  
  To check syntax only  
> To check syntax and find problems  
  To check syntax, find problems, and enforce code style
```

- check syntax : 문법 오류를 검사한다
- find problems : 흔한 실수를 검사한다
- enforce code style: 코드 스타일을 검사한다

Javascript modules (import/export) 항목을 선택하고 엔터키 입력

```
? What type of modules does your project use? ...  
> JavaScript modules (import/export)  
CommonJS (require/exports)  
None of these
```

None of these 항목을 선택하고 엔터키 입력

```
? Which framework does your project use? ...  
React  
Vue.js  
> None of these
```

Yes 항목을 선택하고 엔터키 입력

```
? Does your project use TypeScript? » No / Yes
```

Node 항목을 선택하고 엔터키 입력

```
? Where does your code run? ... (Press <space> to select, <a> to toggle all, <i> to invert selection)  
✓ Browser  
✓ Node
```

Browser : 이 프로젝트의 코드가 웹브라우저에서 실행될 것이라면 선택

Node : 이 프로젝트의 코드가 node.js에서 실행될 것이라면 선택

Yes 항목을 선택하고 엔터키 입력

```
eslint@latest  
? Would you like to install them now? » No / Yes
```

npm 항목을 선택하고 엔터키 입력

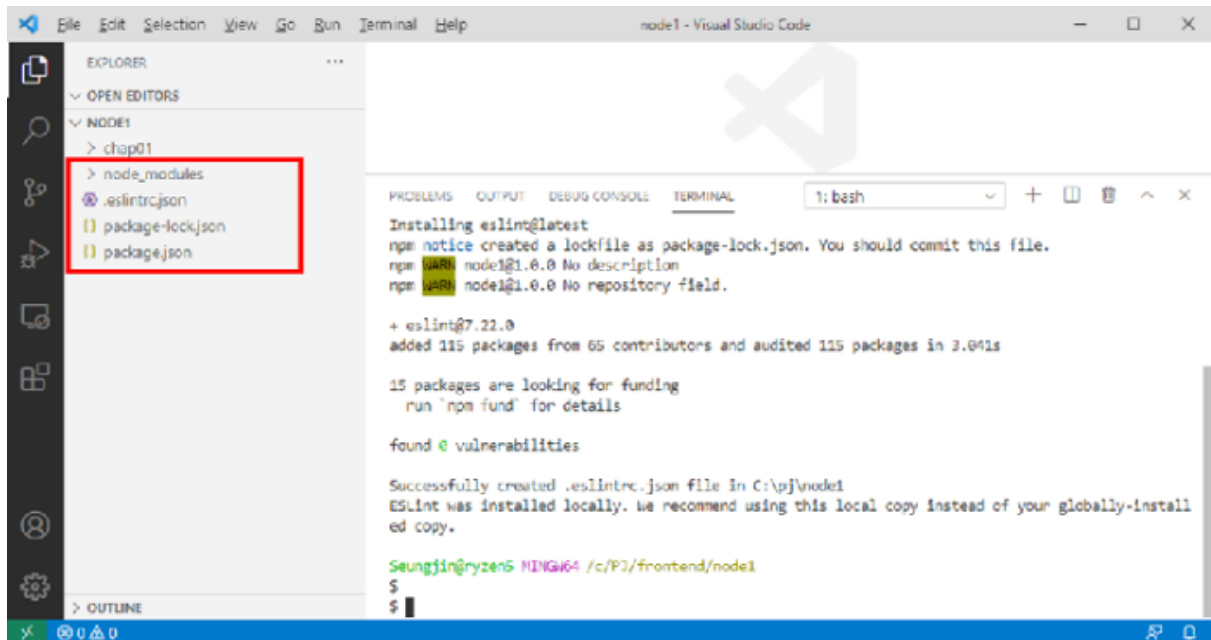
```
? Which package manager do you want to use? ...  
> npm  
  yarn  
  pnpm
```

위 명령에 의해서, eslint의 설정 파일인 .eslintrc.json 파일 생성된다.

6) 프로젝트 폴더

node1 프로젝트 폴더의 내용

- node_modules 폴더
- package.json 파일
- .eslintrc.json 파일



package.json 파일의 내용

node.js 프로젝트 설정 파일

```
{
  "name": "node1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "eslint": "^8.35.0"
  }
}
```

| 항목 | 설명 |
|-----------------|---------------------------|
| name | 프로젝트 이름 |
| version | 프로젝트 버전 |
| description | 프로젝트 설명 |
| main | 프로젝트의 실행이 시작될 첫 파일 |
| scripts | 프로젝트를 빌드, 테스트, 실행하기 위한 명령 |
| keywords | 프로젝트의 검색 키워드 |
| author | 작성자 |
| license | 라이선스 정보 |
| devDependencies | 프로젝트에 필요한 라이브러리들 목록 |

node_modules 폴더

프로젝트에 필요한 라이브러리들이 자동으로 저장되는 폴더

.eslintrc.json 파일

eslint 설정 파일

package-lock.json

프로젝트에 필요한 라이브러리들의 정확한 버전이 자동으로 등록되는 파일.
이 파일은 자동으로 생성된다.
지금은 이 파일을 무시하자.