Prettier

Prettier란?

Prettier는 정해진 규칙에 따라 자동으로 코드 스타일을 정리해 주는 도구입니다.

Prettier는 다음을 지원합니다.

- JavaScript, including ES2017
- JSX
- Angular
- Vue
- Flow
- TypeScript
- CSS, Less, and SCSS
- HTML
- JSON
- GraphQL
- Markdown, including GFM and MDX
- YAML

Installation

yarn

```
yarn add prettier --dev --exact
# or globally
yarn global add prettier
```

npm

```
npm install --save-dev --save-exact prettier
# or globally
npm install --global prettier
```

ESLint + Prettier

ESLint와 Prettier를 함께 사용하려면 ESLint에서 Prettier와 충돌되는 규칙을 비활성화 하고, Prettier의 규칙을 ESLint에 추가 하여야 합니다.

따로 설정을 하여도 되지만

ESLint 비활성

eslint-config-prettier Prettier와 충돌하는 규칙을 비활성화하는 구성입니다. devDependencies에 추가 한 다음 .eslintrc구성 내에서 확장하여 사용. extends 배열에서 마지막에 놓아야 다른 구성을 무시할 수 있습니다.

```
yarn add --dev eslint-config-prettier

npm install --save-dev eslint-config-prettier
```

.eslintrc:

```
{
  "extends": ["prettier"]
}
```

ESLint를 사용하여 Prettier 실행

eslint-plugin-prettier Prettier를 사용하여 컨텐츠를 형식화하는 규칙을 추가하는 플러그인입니다. devDependencies에 추가한 다음 플러그인과 규칙을 사용하도록 설정하십시오.

```
yarn add --dev eslint-plugin-prettier

npm install --save-dev eslint-plugin-prettier
```

.eslintrc:

```
{
  "plugins": ["prettier"],
  "rules": {
    "prettier/prettier": "error"
  }
}
```

권장 구성

eslint-config-prettier와 eslint-plugin-prettier를 함께 구성 합니다.

```
yarn add --dev eslint-config-prettier eslint-plugin-prettier

npm install --save-dev eslint-config-prettier eslint-plugin-prettier
```

.eslintrc:

```
{
  "extends": ["plugin:prettier/recommended"]
}
```

Prettier Ignoring Code

특정부분의 코드에 prettier를 적용하지 않을경우에는 ignore code를 사용하면 되며, 특정 파일 혹은 폴더 자체를 ignore할 경우 프로젝트 Root 폴더에 .prettierignore 파일로 지정할 수 있습니다.

.prettierignore

```
node_modules/
package.json
package-lock.json
src/route/index.js
tsconfig.json
.eslintrc.js
.eslintignore
.prettierrc
babel.config.js
```

```
postcss.config.js
vue.config.js
```

JavaScript

```
matrix(
  1, 0, 0,
  0, 1, 0,
  0, 0, 1
)

// prettier-ignore
matrix(
  1, 0, 0,
  0, 1, 0,
  0, 0, 1
)
```

JSX

```
<div>
  {/* prettier-ignore */}
  <span     ugly  format=''   />
</div>
```

HTML

```
<!-- prettier-ignore -->
<div         class="x"       >hello world</div           >

<!-- prettier-ignore-attribute -->
<div
  (mousedown)="       onStart    (    )         "
  (mouseup)="         onEnd      (    )         "
></div>

<!-- prettier-ignore-attribute (mouseup) -->
<div
  (mousedown)="onStart()"
  (mouseup)="         onEnd      (    )         "
></div>
```

CSS

```
/* prettier-ignore */
.my    ugly rule
{

}
```

Prettier Option

프로젝트 Root 폴더에 .prettierrc 파일로 Prettier의 옵션을 설정 할 수 있습니다.

.prettierrc:

```
{
  "printWidth": 400,
  "singleQuote": true,
  "trailingComma": "all",
  "arrowParens": "always",
  "Bracket Spacing": true
}
```

printWidth (default: 80)

한 줄 내에서 코드를 맞춥니다. 한 줄이 이 글자수를 넘게 되면 줄바꿈되어 코드가 정리됩니다.

tabWidth (default: 2)

텝을 눌렀을 때 몇칸이 띄어지는지를 설정합니다.

singleQuote (default: false)

홑따옴표를 사용 할건지 설정합니다.

trailingComma (default: 'none')

객체, 배열, 함수 등의 후행에 쉼표를 찍을지 제어합니다.

- `none` - 쉼표를 붙히지 않음
- `es5` - ES5에서 유효한 후행 쉼표 붙힘 (객체, 배열 등)
- `all` - 가능하면 후행 쉼표를 붙힘 (함수 인수)

bracketSpacing (default: true)

객체 리터럴 내부의 공백을 제어합니다.

- `true` - Example: `{ foo: bar }`.
- `false` - Example: `{foo: bar}`.

jsxBracketSameLine (default: false)

jsx 요소의 마지막 다음 줄에 닫기 `>`를 표시합니다.

- `true` - Example:

```
<button
  className="prettier-class"
  id="prettier-id"
  onClick={this.handleClick}>
  Click Here
</button>
```

- `false` - Example:

```
<button
  className="prettier-class"
  id="prettier-id"
  onClick={this.handleClick}
>
  Click Here
</button>
```

semi (default: true)

문장 뒤에 세미콜론을 붙힐지 뺄지를 설정합니다.

- `true`
- `false`

useTabs (default: false)

true면 탭이있는 줄을 들여 쓰기 합니다.

- `true`
- `false`

arrowParens (default: 'avoid')

단독 화살표 함수의 매개 변수 주위에 괄호를 자동으로 붙힘니다.

- `"avoid"` - 가능하면 괄호 생략. Example: `x => x`
- `"always"` - 항상 괄호 포함. E*xample:* `(x) => x`

jsxSingleQuote (default: false)

JSX에서는 큰 따옴표 대신 작은 따옴표를 사용하십시오.

- `true`
- `false`

parser (default: 'babylon') - JavaScript only

javascript에서 사용되며 파서를 설정합니다.

Prettier는 입력 파일 경로에서 파서를 자동으로 유추하므로이 설정을 변경할 필요가 없습니다.

`babel`과 `flow`파서는 자바 스크립트의 동일한 세트 (흐름 유형 약어 포함) 기능을 지원합니다. 경우에 따라 다를 수 있으므로 `babel`과 `flow`중 선택하여 사용 합니다.

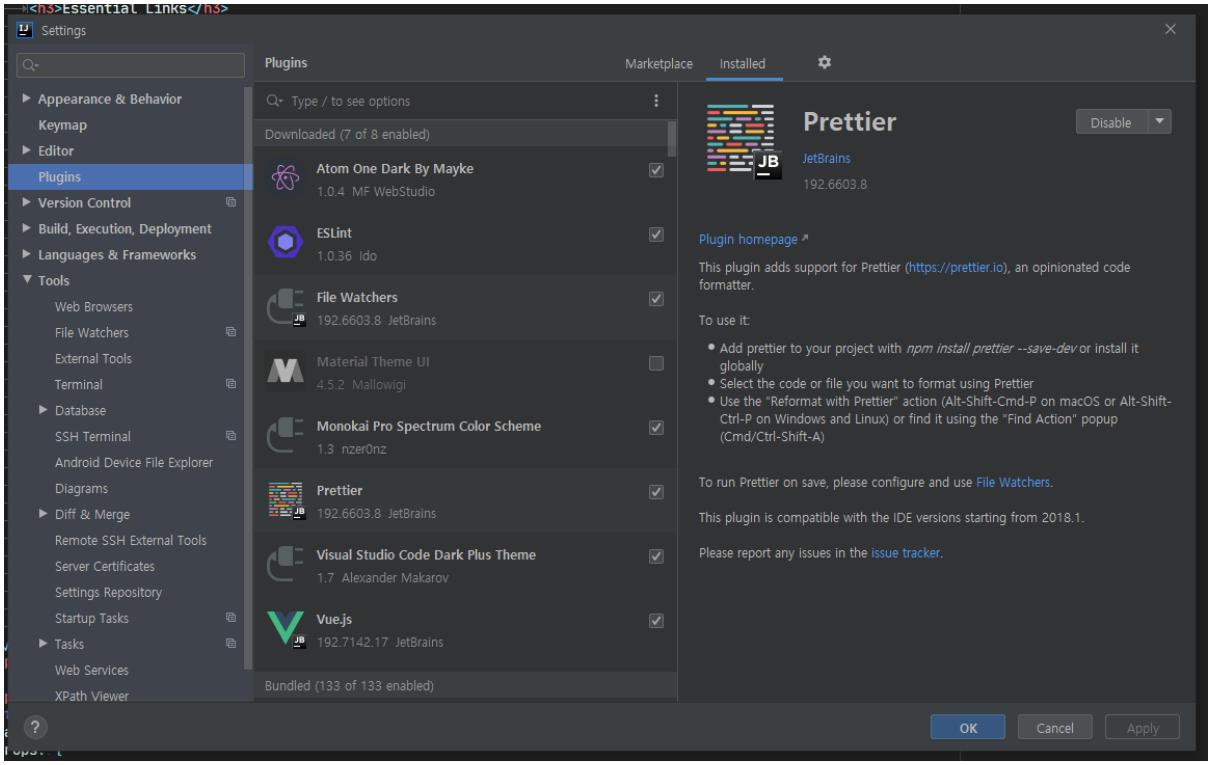
- `"babel"` (via @babel/parser) *Named* `"babylon"` *until v1.16.0*
- `"babel-flow"` (Same as `"babel"` but enables Flow parsing explicitly to avoid ambiguity) *First available in v1.16.0*

- `"flow"` (via flow-parser)
- `"typescript"` (via @typescript-eslint/typescript-estree) *First available in v1.4.0*
- `"css"` (via postcss-scss and postcss-less, autodetects which to use) *First available in v1.7.1*
- `"scss"` (same parsers as `"css"`, prefers postcss-scss) *First available in v1.7.1*
- `"less"` (same parsers as `"css"`, prefers postcss-less) *First available in v1.7.1*
- `"json"` (via @babel/parser parseExpression) *First available in v1.5.0*
- `"json5"` (same parser as `"json"`, but outputs as json5) *First available in v1.13.0*
- `"json-stringify"` (same parser as `"json"`, but outputs like `JSON.stringify`) *First available in v1.13.0*
- `"graphql"` (via graphql/language) *First available in v1.5.0*
- `"markdown"` (via remark-parse) *First available in v1.8.0*
- `"mdx"` (via remark-parse and @mdx-js/mdx) *First available in v1.15.0*
- `"html"` (via angular-html-parser) *First available in 1.15.0*
- `"vue"` (same parser as `"html"`, but also formats vue-specific syntax) *First available in 1.10.0*
- `"angular"` (same parser as `"html"`, but also formats angular-specific syntax via angular-estree-parser) *First available in 1.15.0*
- `"lwc"` (same parser as `"html"`, but also formats LWC-specific syntax for unquoted template attributes) *First available in 1.17.0*
- `"yaml"` (via yaml and yaml-unist-parser) *First available in 1.14.0*

IntelliJ Setting

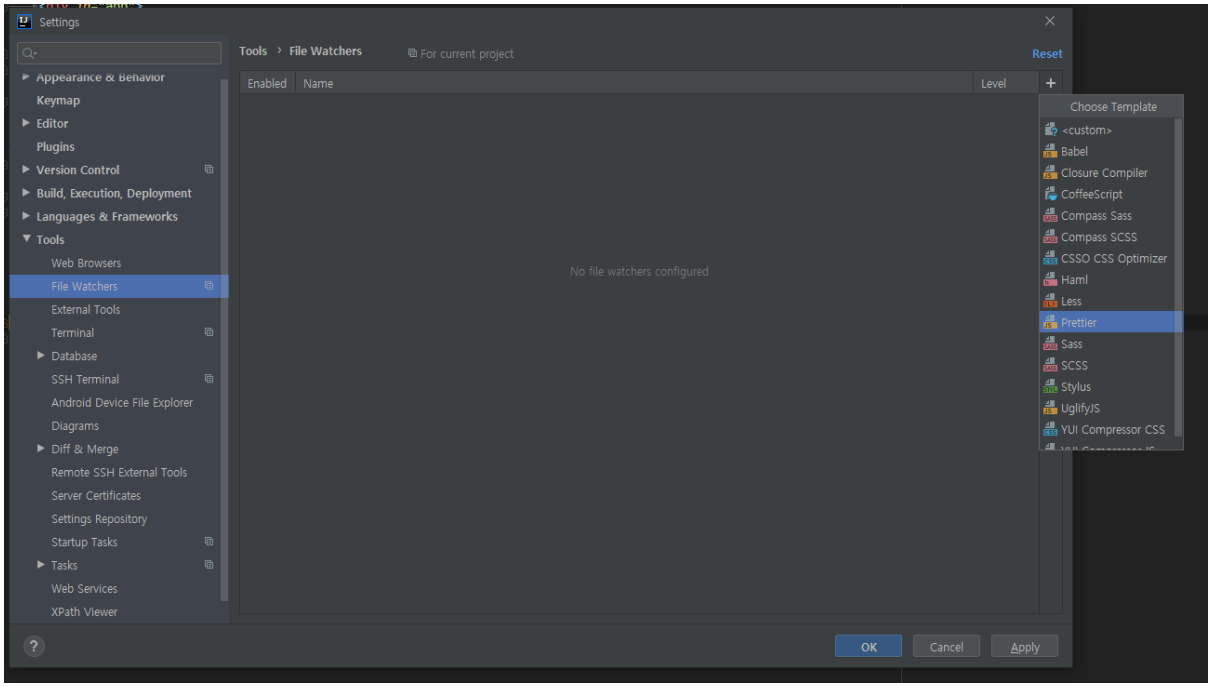
Prettier Plugin Install

File > Settings > Plugins > Marketplace 에서 prettier 검색 후 Prettier Plugin 설치가 필수로 필요합니다.

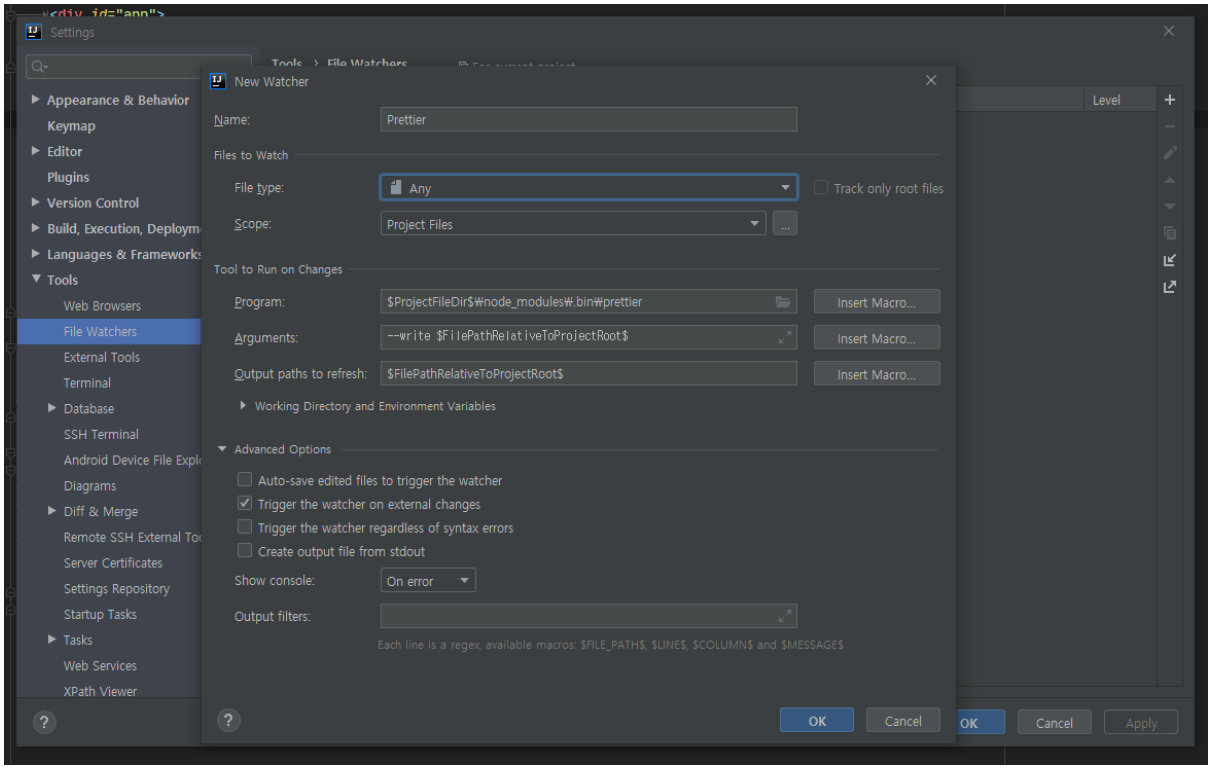


Prettier 실행

File > Settings > Tools > File Watcher > 오른쪽 + 버튼 > Prettier 선택.



File type을 Javascript(*.js 파일만)에서 Any(모든 파일)로 변경 후 저장(적용하고 싶은 확장자만 구성 할 수 있음).

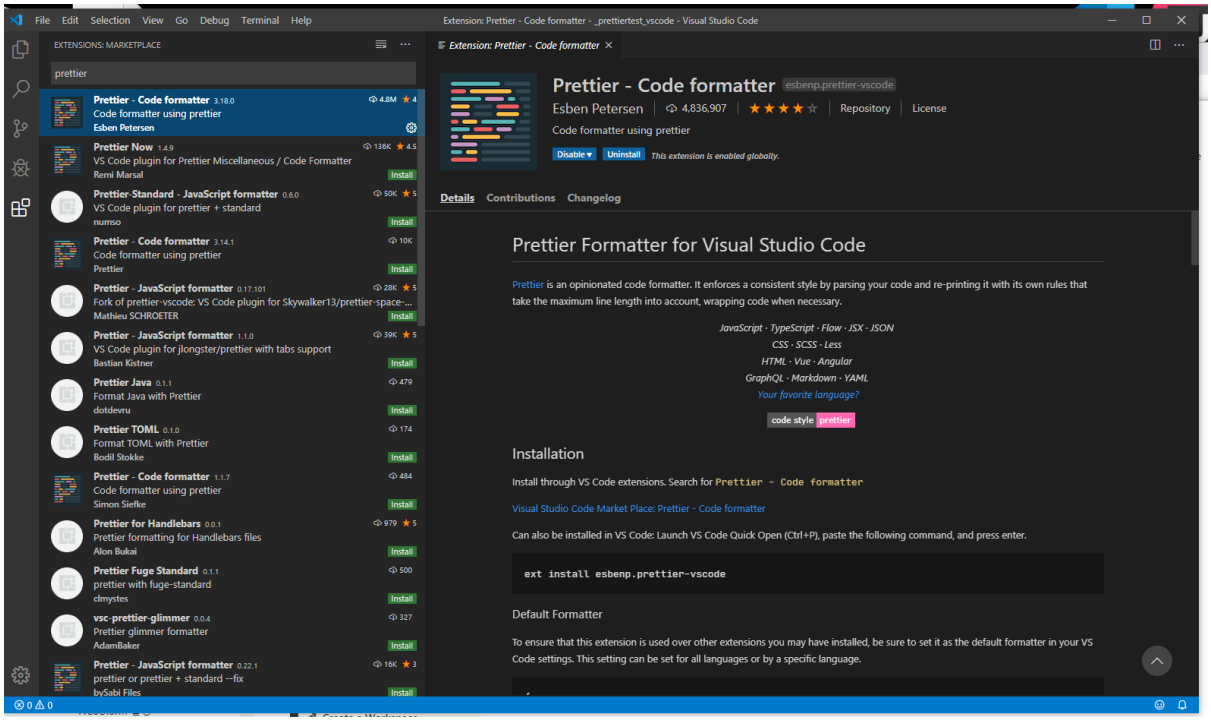


파일을 저장 하게 되면 Prettier가 실행되어 Code를 교정해 줍니다.

VS Code Setting

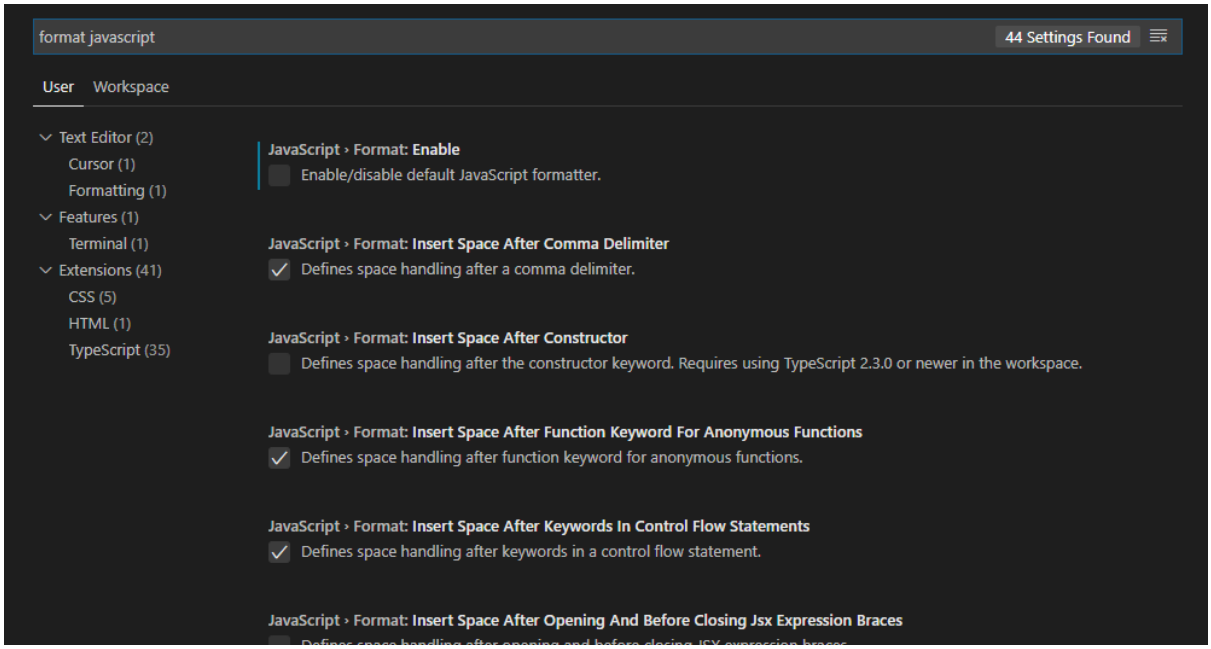
Prettier Plugin Install

File > Preperences > Plugins > Extensions 에서 prettier 검색 후 Prettier - Code fomatter 설치가 필요합니다.

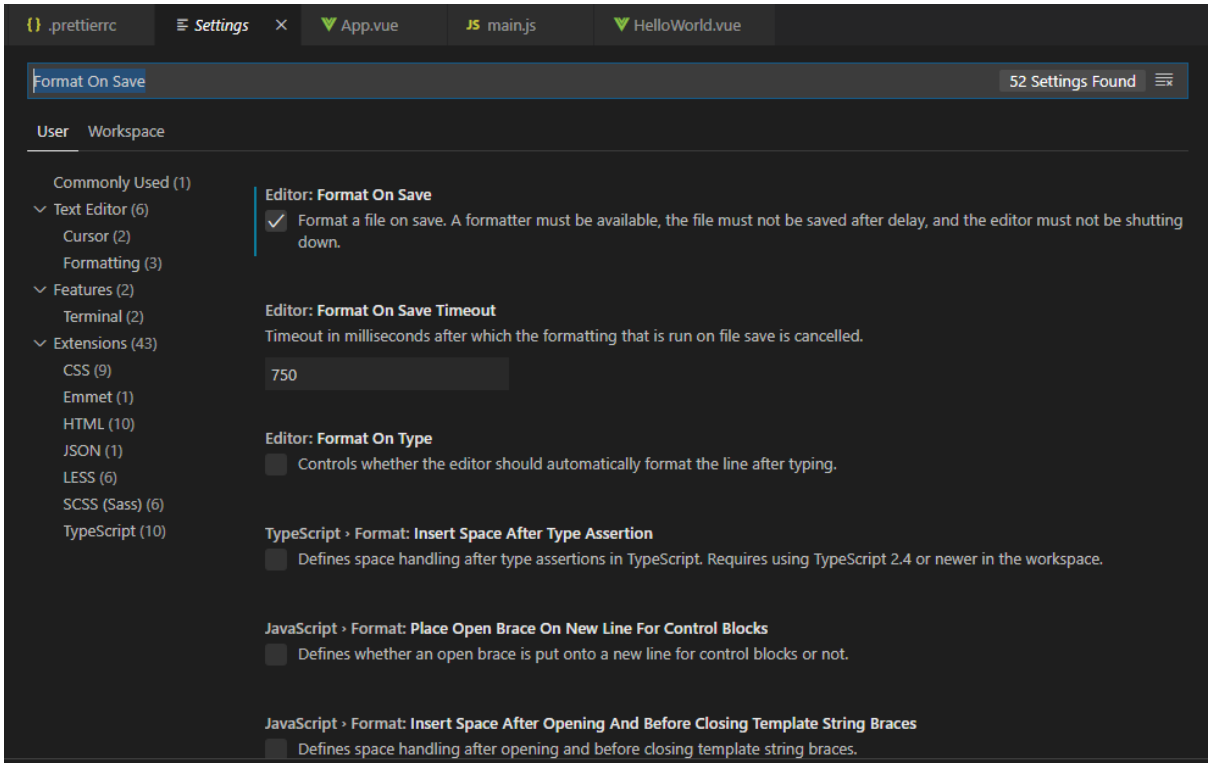


VS Code Setting - Global Setting

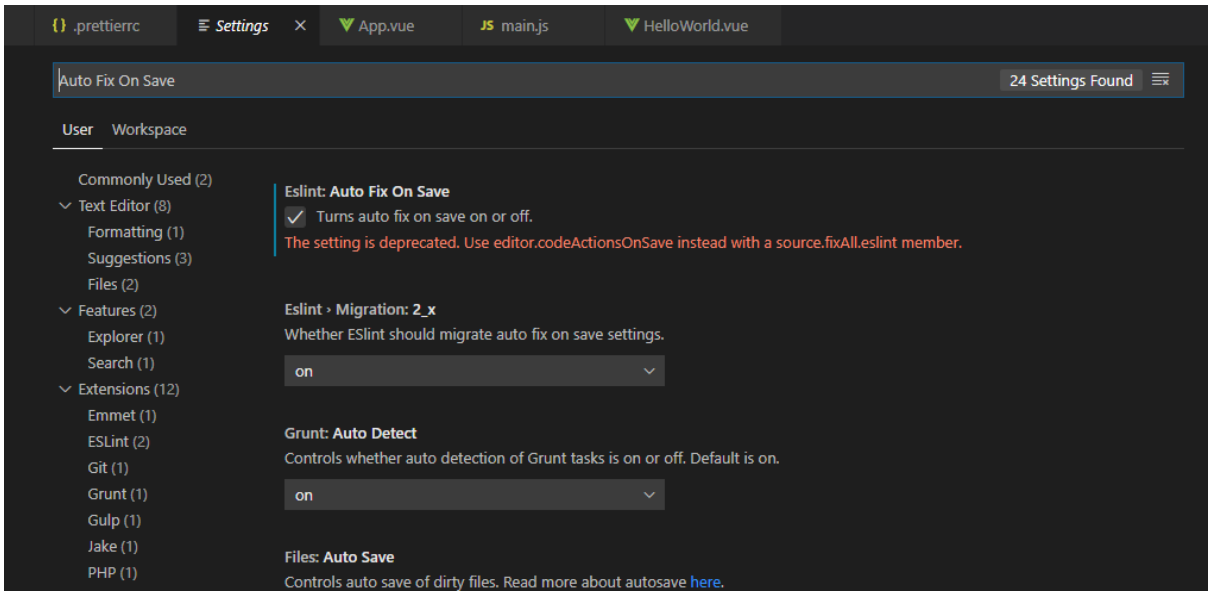
File > Preperences > Settings > format javascript 검색 후 비활성.



File > Preperences > Settings > Format On Save 검색 후 활성화.



File > Preperences > Settings > Auto Fix On Save 검색 후 활성화.

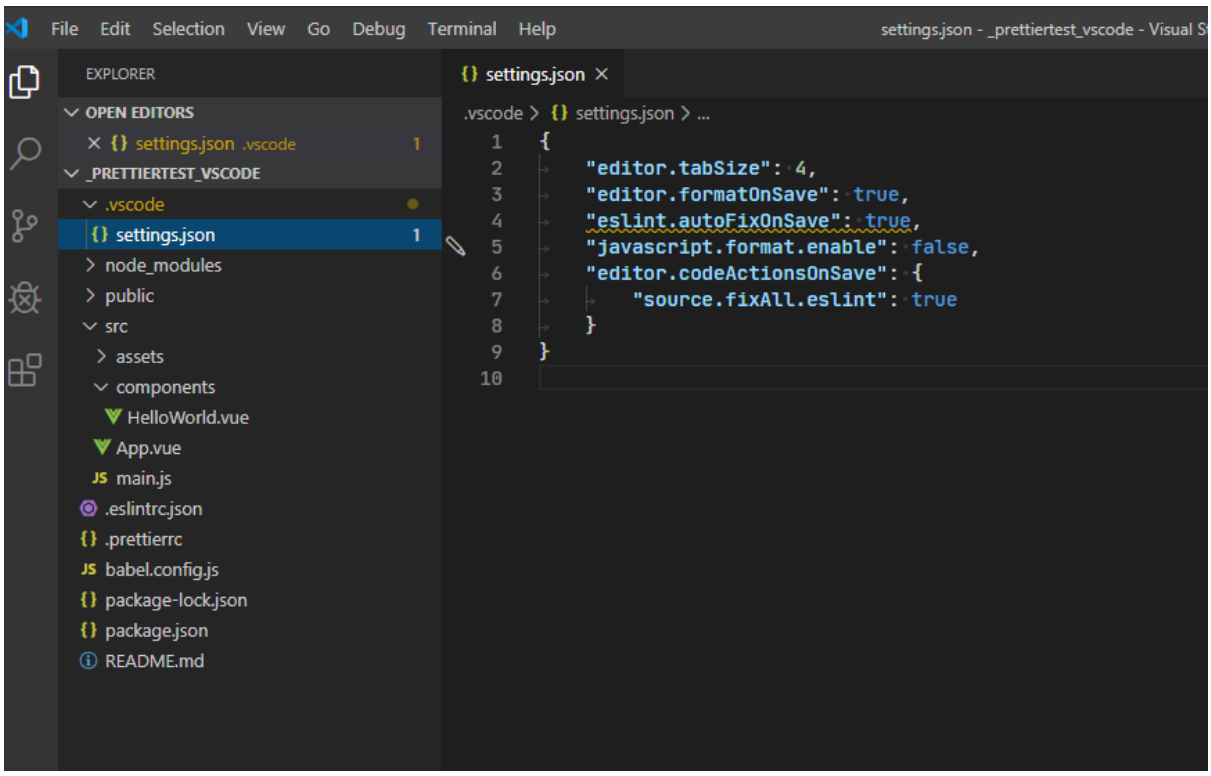


파일을 저장 하게 되면 Prettier가 실행되어 Code를 교정해 줍니다.

VS Code Setting - Project 별도 Setting

Project Root에 .vscode 폴더를 생성하여 해당 프로젝트에만 적용되는 Setting 파일을 생성합니다. 이는 VS Code의 Global Setting 보다 우선하여 적용 됩니다.

Prettier관련 Setting은 물론 기타 Setting또한 가능 합니다.




```
{
  "editor.tabSize": 4,
  "editor.formatOnSave": true,
  "eslint.autoFixOnSave": true,
  "javascript.format.enable": false,
  "editor.codeActionsOnSave": {
    "source.fixAll.eslint": true
  }
}
```