

ESLint 와 Formatter, Vue3에 사용하기

2022.08.04.Thu

Author : Young

Contents

1. ESLint/Prettier란?

2. Vue3 ESLint 세팅

3. ESLint Rules

4. ESLint - Prettier 적용

5. ESLint Settins.json 수정하기

추가 참고사항

동작화면

ESLint/Prettier란?

협업 등에 있어서 코드 컨벤션은 굉장히 중요함.

- **Lint** : 보푸라기
에러가 있는 곳에 표시해두는 모양을 의미
- **ESLint** (EcmaScript + Lint)
코드를 분석해 문법적인 오류나 안티패턴을 찾아준다.
- **Prettier** **란?** Formatter 의 일종
- **Formatter****란?** 저장 시 코드의 구조를 맞춰주는 역할

우리가 할 일 : ESLint 의 Rule을 Prettier에 적용하여 저장 시에 알아서 규칙을 적용할 수 있게 해주겠다는 것임!

Vue3 ESLint 세팅

터미널에 아래와 같이 입력

```
$ vue init
```

실행화면은 아래와 같다.

```
Vue.js - The Progressive JavaScript Framework

✓ Project name: ... learn-vue3
✓ Add TypeScript? ... No / Yes
✓ Add JSX Support? ... No / Yes
✓ Add Vue Router for Single Page Application development? ... No / Yes
✓ Add Pinia for state management? ... No / Yes
✓ Add Vitest for Unit Testing? ... No / Yes
✓ Add Cypress for both Unit and End-to-End testing? ... No / Yes
✓ Add ESLint for code quality? ... No / Yes
✓ Add Prettier for code formatting? ... No / Yes
```

- ESLint / Yes
- Prettier / Yes

위와 같이 세팅

ESLint, Prettier 설정

eslint.rc.cjs 파일

```
/* eslint-env node */
require("@rushstack/eslint-patch/modern-module-resolution");

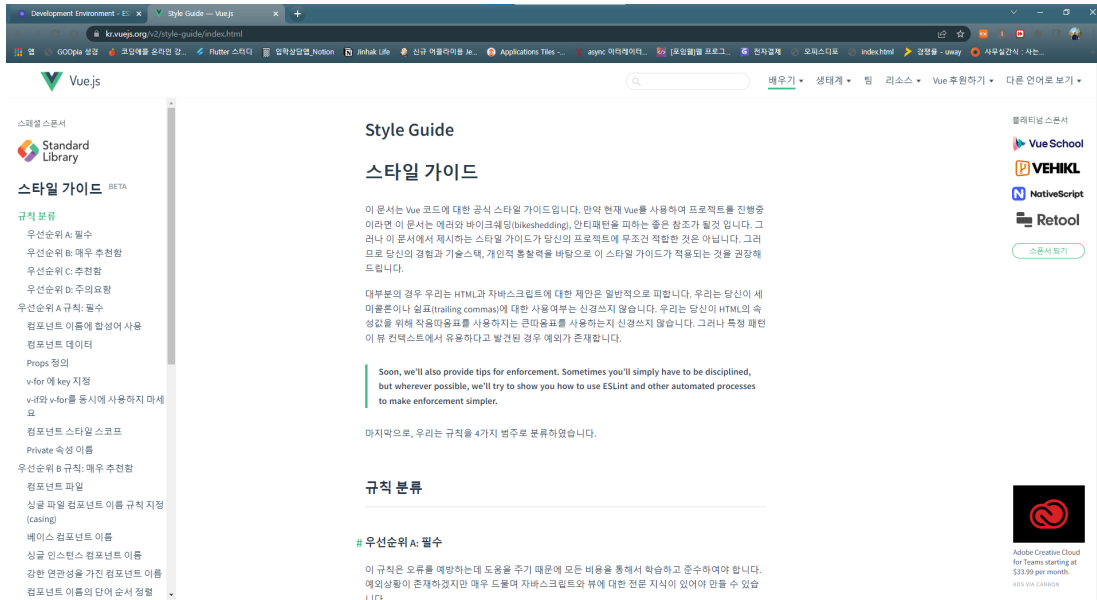
module.exports = {
  "root": true,
  "extends": [
    "plugin:vue/vue3-essential",
    "eslint:recommended",
    "@vue/eslint-config-prettier"
  ]
}
```

- "plugin:vue/vue3-essential", : Vue 스타일가이드
- "eslint:recommended", : eslint의 추천항목을 검사하라
- "@vue/eslint-config-prettier" : 충돌 방지를 위해 필요함.

Vue Style-Guide

"plugin:vue/vue3-essential",

vue에는 style-guide가 있음 특정 패턴으로 개발하면 굉장히 유용하다.



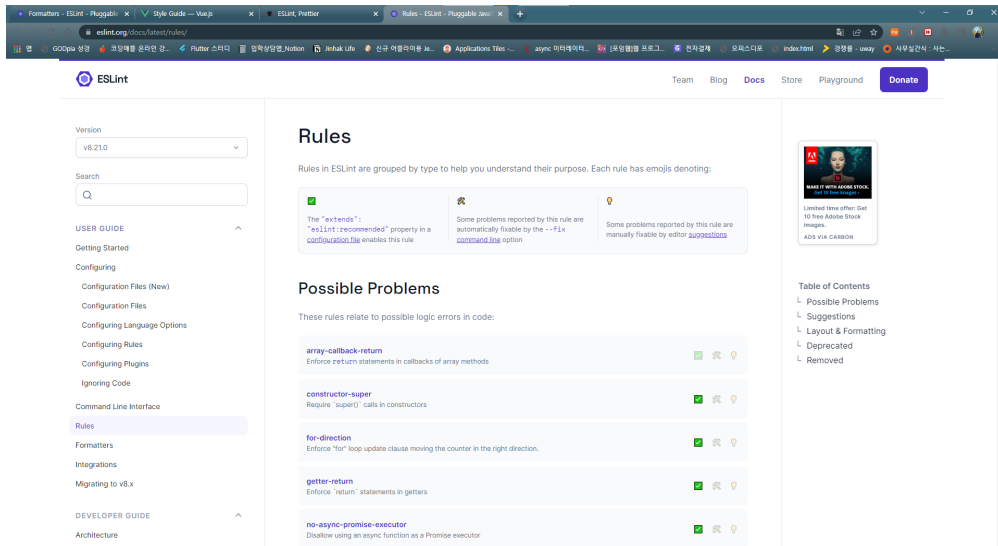
참조링크

1. 컴포넌트 이름을 사용할 땐 합성어로 사용하라.
2. props를 설정할 때도 배열로 하기보다 객체로 만드는것이 상세하다.
.. 등등의 내용이 있음 이를 가져와서 사용해라.라는 뜻임

ESLint Rules

"eslint:recommended",

ESLint의 추천 항목을 자동으로 검사해라.



ESLint Rules 관련 정보

"@vue/eslint-config-prettier",

충돌 방지 옵션임.

ESLint Rules

위 페이지에 가면 다양한 규칙이 있고
아래와 같이 사용 가능하다.

```
// eslint.rc.cjs 파일
rules: {
  "no-console": "error",
  "camelcase": process.env.NODE_ENV === "production" ? "error" : "off",
},
```

위와 같이 만들 수 있는데
규칙 위반 시의 경우를 아래와 같이 설정할 수 있다.

- error (에러를 뜨게)
- warn (경고창만 띄움)
- off (사용 안함)

ESLint - Prettier 적용

```
rules: {  
  "no-console": process.env.NODE_ENV === "production" ? "error" : "off",  
  "prettier/prettier" : ["error", {  
    //프리티어 룰들 적용  
  }]  
}
```

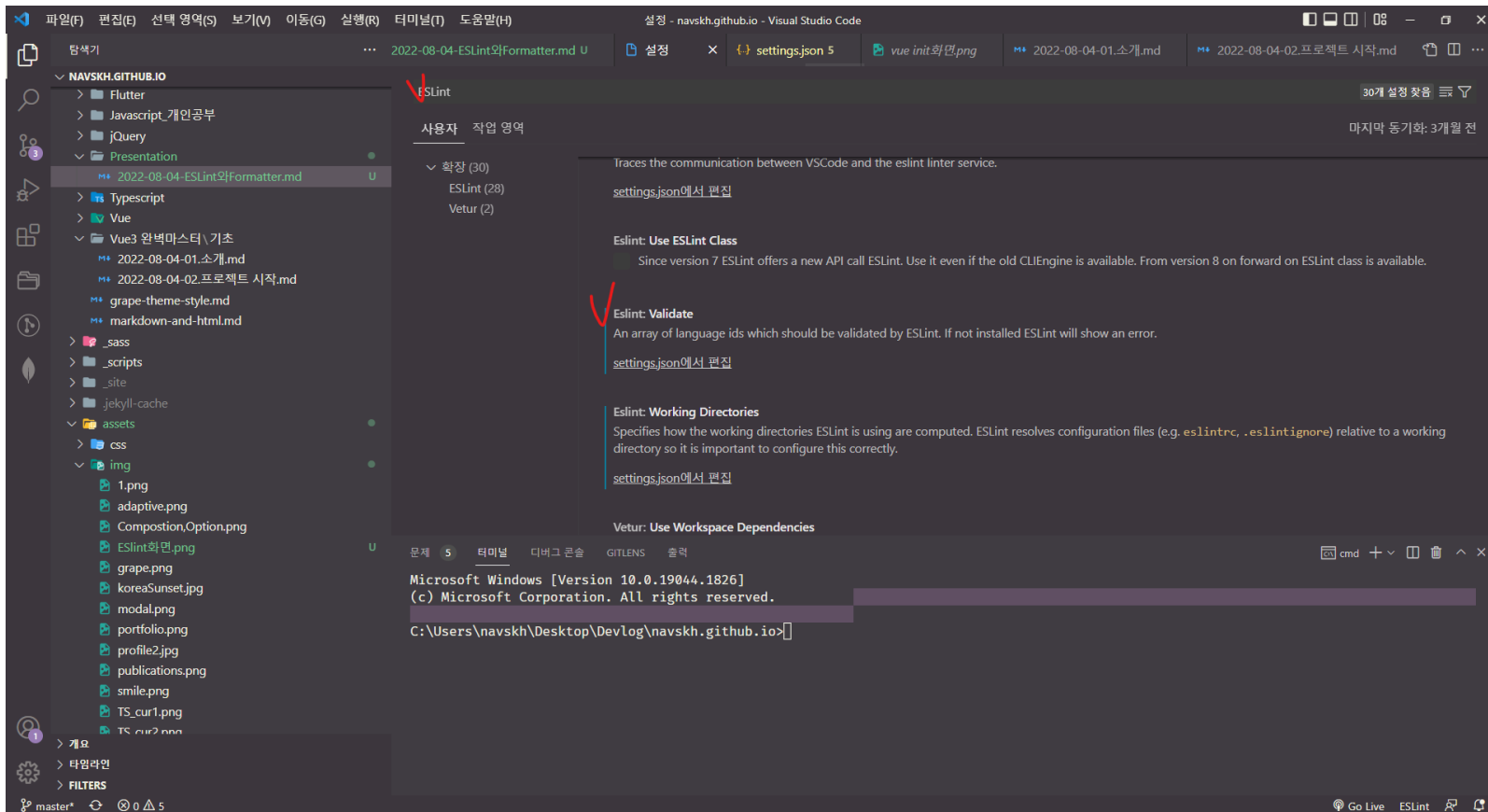
```
"prettier/prettier" : ["error", {}]
```

이 안에 Rule 들을 적용하면

Prettier에 의해 저장 시 알아서 Rule에 맞게 변환되어 저장될 수 있도록 만들 수 있음.

ESLint VSCode 에 설정하기

ctrl + , 를 누른 후
ESLint를 검색한 후 Validate를 찾아서 Settings.json을 클릭한다.



ESLint Settns.josn 수정하기

- `"eslint.validate"` : eslint 활성화 되는 언어 지정 가능

```
//settns.json
"eslint.validate": [
  "javascript",
  "typescript",
  "vue",
  "html",
  "markdown"
],
```

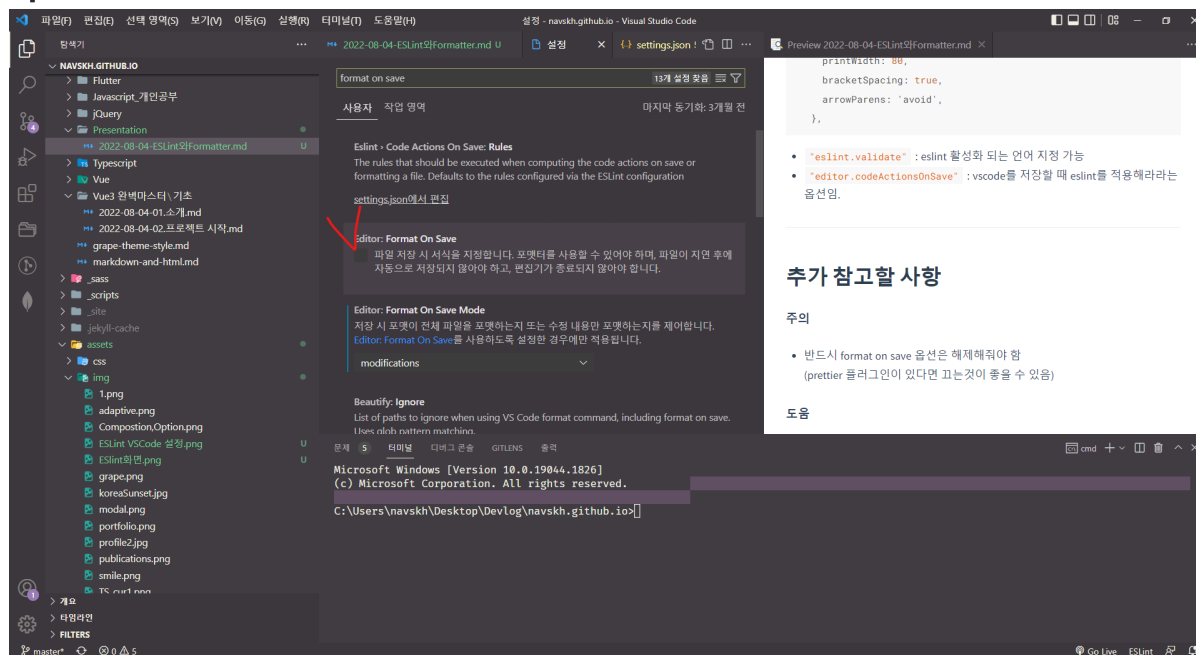
ESLint Settings.json 수정하기

- `"editor.codeActionsOnSave"` : vscode를 저장할 때 eslint를 적용해라라는 옵션임.

```
"editor.codeActionsOnSave": {  
  singleQuote: true,  
  semi: true,  
  useTabs: true,  
  tabWidth: 2,  
  trailingComma: 'all',  
  printWidth: 80,  
  bracketSpacing: true,  
  arrowParens: 'avoid',  
},
```

추가 참고할 사항

- 반드시 format on save 옵션은 해제해줘야 함 (prettier 플러그인이 있으면 돌게 됨)



- `npm run lint`를 통해 전체 파일에 lint 진행을 할 수 있다.

적용되는 화면 (영상 시연)

```
learn-vue3 > src > lectures > TestLint.vue > Vetur > {} "TestLint.vue" > template > div > button
1  <template>
2    <div>
3      <button v-on:click="increment">
4    </button>
5  </div>
6 </template>
7
8 <script>
9 import { onMounted, ref } from 'vue';
10 export default {
11   setup() {
12     const counter = ref(0);
13     const increment = () => counter.value++;
14
15     onMounted(() => {console.log('마운트!');});
16     return {counter, increment};
17   },
18 };
19 </script>
20
21 <style>
22 </style>
23
```

