

# Dev Guide for Vue

---

이 문서는 사내 Vue 프레임워크 프로젝트 레이아웃을 다루며,  
개발자가 소스 코드 작성 시에 따라야 할 규칙과 그에대한 가이드를 기술합니다.  
Vue의 패키지 버전은 2가 아닌 3이상을 사용 해야합니다.(작성일 기준 3.3.2)  
Typescript를 사용 하십시오.  
이 프로젝트는 사내 인원들의 노력으로 품질을 향상 시킬 수 있습니다.  
새로운 혹은 업데이트가 필요한 패턴이 있다면 이슈를 열어주세요.

다른 코드 구조 추천과 네이밍과 패키지 구성에 대해 더 알아보기:

- <https://vuejs.org/style-guide/>
- <https://vue-dev-guide.netlify.app/guide>

## 목차

- [Dev Guide for Vue](#)
  - [다른 코드 구조 추천과 네이밍과 패키지 구성에 대해 더 알아보기:](#)
  - [목차](#)
- [세팅\(Setup\)](#)
  - [신규 프로젝트 생성시](#)
- [파일/폴더 구조](#)
  - [파일 작성 방법](#)
  - [폴더 구성 방법](#)
    - [/locales/](#)
    - [/public/](#)
    - [/src](#)
      - [/src/components](#)
      - [/src/composables](#)
      - [/src/{service}](#)
      - [/src/layouts](#)
      - [/src/modules](#)
      - [/src/pages](#)
      - [/src/store](#)
      - [/src/styles](#)
      - [/src/types](#)
    - [/test/](#)
- [Vue API/Pkg 이용](#)
  - [Pinia\(Store\)](#)
    - [BAD](#)
    - [GOOD](#)
  - [TODO](#)

---

## 세팅(Setup)


---

1. vscode(IDE)가 설치 되었는지 확인 하십시오.
2. 다음 vscode Extensions 목록을 설치 하십시오.
  1. [Vue Language Features \(Volar\)](#)
  2. [Typescript Vue Plugin](#)

## 신규 프로젝트 생성시

1. 회사 공식 setup baseline 프로젝트를 Clone 합니다.

```
$ git clone https://github.com/socketbear/vue-dev-guide.git
```

2. [Typescript takeover to Volar 활성화](#) 기본적으로 vscode는 builtin 익스텐션으로 TS(Typescript)를 지원합니다. 하지만, Volar는 Vue 와 TS 둘 다 지원 합니다. 이는 1프로젝트 당 2개의 Ts지원 익스텐션이 활성화 됨을 의미하며, 큰 프로젝트에서 성능저하를 야기합니다. 하단 이미지와 같이 Bottom 윈도우에서 takeover을 확인 하십시오 

## 파일/폴더 구조

---

### 파일 작성 방법

일관성, 가독성 향상을 위해 일정 형식과 순서에 따라 작성되어야 합니다. [example](#)

- 모든 vue 파일은 composition-api(options가 아님에 유의)을 이용하여 작성합니다.

### 폴더 구성 방법

[/locales/](#)

[/public/](#)

[/src](#)

source 의 약어로 프로젝트를 구성하는 소스코드가 저장될 폴더 입니다.

[/src/components](#)

[/src/composables](#)

[/src/{service}](#)

[/src/layouts](#)

[/src/modules](#)

[/src/pages](#)

프로젝트에서 사용중인 패키지 [vite-plugin-pages](#)는  
위 폴더의 \*.vue/js 파일에서 export 된 vue 컴포넌트를 페이지로 구성합니다.

- 각 파일명은 kebab-case를 준수하여 작성 합니다.
- 각 파일명은 URL 과 1대1 매칭이 되어야합니다.

### /src/store

특정 {service}가 아닌 전역적으로 사용되는 [Pinia](#)모듈로 구성된 폴더입니다.

- 각 파일명은 camelCase를 준수합니다.

### /src/styles

### /src/types

### /test/

## Vue API/Pkg 이용

---

[vue api](#) 목록을 프로젝트에 적용 하였을때의 가이드, 주의사항으로 구성 되어있습니다.

### Pinia(Store)

Vue(공식) 어플리케이션의 상태관리(State Management) 라이브러리로서,  
App 내 모든 컴포넌트에 대한 중앙 집중식 저장소(store) 역할을 담당합니다. "자사"는 "Options API" 패턴을 사  
용 하지 않으며 반드시 "Composition-API" 패턴을 사용 해주시기 바랍니다.

#### BAD

```
export const useCounterStore = defineStore('counter', {
  state: () => ({
    count: 0,
  }),
  getters: {
    doubleCount(state) {
      return state.count * 2
    },
  },
})
```

#### GOOD

```
export const useCounterStore = defineStore('counter', () => {
  const count = ref(0)
  const name = ref('Eduardo')
  const doubleCount = computed(() => count.value * 2)
```

```
function increment() {  
  count.value++  
}  
  
return { count, name, doubleCount, increment }  
})
```

## TODO

- ☒ Option B
- ☐ [https://drive.google.com/drive/folders/1o2szWphReMZ3Z\\_jzz9FCJA8EEERG91S8?usp=share\\_link](https://drive.google.com/drive/folders/1o2szWphReMZ3Z_jzz9FCJA8EEERG91S8?usp=share_link)
- ☐ <https://vue-dev-guide.netlify.app/guide>