**VS Code Settings / Workspace**

Extensions Dependencies

1. Vue – Official (Base)
2. Vetur (Vue Add On)
3. ESLint (Vue Error Formatter)
4. Prettier – Code Formatter (Code Formatter)

Suggested Extensions

1. Pretty Typescript Errors – Helps understand TS errors

**Vue 2**

Getting Started

1. Install Vue CLI Globally – *One time installation*

npm install -g @vue/cli

or clone repository from Github link

* Create new empty folder
* Run cmd in the created folder
* gh repo clone repo\_name (replace repo\_name with actual repository name)
* Right click inside cloned repo, “Open with VS Code”

npm install // to install node\_modules (dependencies to run)

1. Create a new project

* Run cmd in project folder directory

vue create my-vue-app

* Replace “my-vue-app” with project name

1. Select manual features:

* Choose **‘*Manually select features’*.**
* Select ***Babel, TypeScript, Router, Vuex, Linter / Formatter***.
* Choose ***2.x*** for Vue version.
* Set up ESLint and choose ***ESLint + Prettier***.
* Use class-style component syntax – ***Yes***
* Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? – ***Yes***
* Use history mode for router? (For routing to pages) – ***Yes***
* Pick a linter / formatter config: ***Prettier***
* Pick additional lint features: ***Lint on save***
* Choose where to store config (***Dedicated for large project, package.json for small project***)

1. General Layout

**Folders in project directory**

1. src – Source, all coding files in here
2. assets – Photos and other assets are put here
3. components – Most pages / components are put here
4. router – Router / Page settings file here (Manages paging routing methods)
5. store – Vuex settings file here. (Store state management etc)
6. views – App.vue for the application overlay, main.ts for vue initialization
7. types – Datatypes are handled here. (d.ts files)
8. services – API services are handled here (.ts files – api.ts, …service.ts)
9. General commands

**Vue 2**

|  |  |
| --- | --- |
| Code | Description |
| npm install | Installs project dependencies specified in package.json. |
| npm list | Lists installed packages and their versions. |
| npm run lint | Runs ESLint to check and fix code style issues. |
| npm run serve | Compiles and hot-reloads the Vue app for development. |
| npm run build | Compiles and minifies the Vue app for production. |
| npm run test | Runs unit tests for the Vue app. |
| npm install vue-router | Installs Vue Router for client-side routing. |
| npm install vuex | Installs Vuex for state management. |
| npm install -g @vue/cli | Installs Vue CLI globally for project scaffolding and management. |
| vue create my-vue-project | Creates a new Vue project using Vue CLI. |
| vue add <plugin> | Adds a plugin to the Vue project (e.g., Vue Router, Vuex). |

**Git**

|  |  |
| --- | --- |
| Code | Description |
| git status | Shows the current status of the repository. |
| git add -A | Adds all changes in the working directory to the staging area. |
| git commit -m "commit notes" | Commits staged changes with a descriptive message. |
| git push | Uploads local repository content to a remote repository. |
| git pull | Fetches and integrates changes from a remote repository to the local one. |
| git remote -v | Lists all remote repositories and their URLs. |
| git branch | Lists all branches in the repository. |
| git checkout <branch> | Switches to the specified branch. |
| git merge <branch> | Merges changes from <branch> into the current branch. |
| git log | Displays commit history. |

## Frequenly Asked Questions (FAQ’s)

**VS Code Settings / Workspace**

1. I need to run “npm run lint” every time I edit a line of code.

**Reason**

* This occurs when lint doesn’t automatically run on save.

**Solution**

* Open VS Code settings [ File > Preferences > Settings ]
* Open Settings.json

A screenshot of a computer

Description automatically generated

* Paste and save the following

{

“editor.codeActionsOnSave”:{

“source.fixAll.eslint”:true

}

"editor.formatOnSave": true,

}

1. V-Tooltip errors in web console.

**Reason**

* V-Tooltips not installed and configured properly

**Solution**

* Add these codes in your main.ts

import VTooltip from "v-tooltip"; //Add this line

Vue.config.producionTip = false;

Vue.use(Tooltip); // Add this line

* Create types file v-tooltip.d.ts and add this line

Declare module “v-tooltip”;

1. I accidently upgraded to Vue 3, how to revert the upgrade?

**Reason**

* You ran npm install which sometimes upgrades Vue 2 to Vue 3, causing lots of package dependency errors.

**Solution**

* Uninstall Vue 3 or bugged Vue versions

npm uninstall vue

* Install Vue 2

npm install vue@2

* Check installed packages and dependencies

npm list

1. Coding’s encountering problems regarding prettier/prettier formatting.

**Reason**

* Prettier not configured properly in package.json or extension settings.

**Solution**

* Try running this command

npx prettier --write . // the dot included (writes all files)

* Add these codes in your package.json, rules section if still doesn’t work

"prettier/prettier": [

"error",

{

"endOfLine": "lf"

}

]