1. 集成TS的方式 { 1) 使用 Vue CLI 创建并支持 TS 的项目
2) 在已有项目中配置.

1) vue create xxx

① Use Class-style component syntax: 是否使用类风格定义组件.

② Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)?

a): 是否TS只负责TS, 其它都由babel来处理.

b) 虽然TS也可以指定兼容的版本号

c): 对于polyfill这些功能, 如果不选, 则需要TS和babel都转换一次.

③ lint 时机可多选

2. vue-class-component、vue-property-decorator: 用类的方式写 Vue 组件.

3. .eslintrc.js 通过 extends 和 { @vue/typescript/recommended 支持vue和ts校验.
@vue/standard: 支持vue校验 }
parserOptions: 指定校验js的版本

4. 定义组件的方式:

1) 普通: export default {...}

2) Options API: export default Vue.extend({

...

})

优点: 编辑可以在写代码时check语法. 推荐 (decorator还不稳定)

3) class API: export default class Foo extends Vue {...}

① beforeRouterEnter 这些第三方钩子, 需要注册才可以使用!

4) class API + vue-property-decorator 非官方, 基于vue-class-component.

5. ts识别别名在 tsconfig.json 中 paths 字段.

6. readonly.

7. 版本号: 13.5.6
   主版本 次版本 小版本

① 13.5.6: 固定13.5.6版本
② ~13.5.6: 锁定次版本号后面最新版本 → 13.5.6~x
③ ^13.5.6: 锁定主版本号后最新版本 → 13.5~x.x
④ latest: 最新版

8. 使用eslint作为vscode默认格式化器.

step1: 安装 eslint
step2: vscode 引出提示, 使用 node_modules/eslint 作为配置 → 右键 ✓
step3: setting → 搜索 eslint → 勾选 Eslint > Format: Enable.
step4: 选择默认格式化工具为 eslint
Ps: 修改设置相关的系统时, 有时需要重启 vscode!!!