淘票票项目文档

项目目录

├─api 生成mock模拟数据，使用mockjs生成

├─components 公共组件目录，主要存放项目中公共的组件，不同页面都用到的组件，或者是将来可能会成为公共的组件

├─directive 自定义指令，实现一些效果类功能

├─filters 过滤器，对于一些公用的输出格式化使用过滤器，对于不共用的输出格式可以使用计算属性来完成

├─mock 生成模拟接口，使用axios-mock-adapter配合axios完成onGet，onPost模拟接口

├─plugin 插件，vue中的插件，像一些全局使用功能，插件使用install创建，use注册

├─router 项目路由，路由的配置项，和路由拦截（钩子函数，生命周期）的配置

├─static 静态资源目录，静态css，静态image，静态插件

│ ├─css

│ ├─images

│ └─lib

└─views 页面目录，每个页面目录下包含一个index.vue页面的主要入口和一个components文件夹，页面中的组件

├─Login

├─Movie

│ └─components

├─MovieMain

└─Zoom

使用命令

Npm run dev 启动开发环境

Npm run build 打包项目，生成上线项目

Vue实例

el 挂载目标，可以是css选择器，也可是HTMLElement实例，el的替代方案是$mount()

data:页面需要使用的所有数据，但是有一个区别在vue实例中data是对象，在组件中data是function

data: {

}

Data: function () {

Return {

}

}

Methods：实例中所有要使用到的方法

Components：注册组件，一个组件的使用必须，先引入，后注册，然后使用

Computed：计算属性

Watch： 一个对象，键是需要观察的表达式，值是对应回调函数，用于监听对应值的变化，还可以监听路由的变化

Watch，computed，methods 的区别

Watch是对单一的某一个值的变化进行监听，

Computed，methods，当所依赖的值发生改变，方法就会自动执行一遍

Computed如果依赖的值没有发生改变，多次调用只会执行一次，但是methods就不一样了调用就会执行

Computed会缓存计算的结果，使用闭包原理，如果值未变，直接返回结果不进行计算

声明周期

一个实例或者一个组件的，创建到销毁的过程，这个过程也称为回调，也叫钩子函数

默认情况下

初始化数据阶段：created 一般初始化的数据在这里进行操作，ajax初始化数据的请求

Dom挂载阶段：mounted dom元素可见$el出现，在这里我们可以做和dom相关的操作

更新回调：updated 当数据的改变引起了dom的重新渲染，才会触发

销毁destroyed 当组件销毁的时候，一般在这里主要是做一些垃圾回收，像清除定时器，清除浏览器事件。。。。。。

在上面每一个声明周期都会有一个 xxxx之前， beforCreate beforeUpdate beforeDestroy

Keep-alive开启的时候（组件缓存）

Activated 组件启用的时候

Deactivated 组件停用的时候

当keep-alive开启后，

destroyed这部分钩子函数将不会执行，因为组件是缓存了，并没有销毁

create，mounted永远都只执行一次，就是当网站第一次打开的时候，而是执行 activated

插件：

插件通常会为Vue添加全局功能。使用install方法创建，使用use方法注册

插件的概念其实就是给vue的原型添加了一个方法