

vue 个人使用记录

1.使用命令vue create 项目名称(不可以使用驼峰命名可以使用 my-project)

2.个人使用vue-cli搭建项目以及添加的一些自定义配置

3.项目一般需要使用到 vue-router vuex axios

--其中vue-router 负责路由跳转、vuex负责状态管理、axios负责请求接口数据

my-project

----public

-----src

----api // 存放axios通用配置 以及部分公用报错信息

--api.js // 此文件编写请求接口

--http.js // 此文件编写axios通用配置以及响应拦截等信息

----components // 组件目录

----router // 路由目录

--index.js // 编写每个组件路由

----config // 通用配置文件 比如 ip 端口号

--config.js

----util // 各种项目通用方法

--util.js

4.axios

响应拦截器:

```
axios.interceptors.response.use(
```

```
  response => {
```

```
    // 此处编写拦截后需要做的操作 比如拦截token 判断用户是否登录等
```

```
  },
```

```
  error => {
```

```
    // 这里处理异常信息
```

```
  }
```

```
);
```

post get等方法封装(下边的是post方法封装其余的类似):

```
export function post(url, params = {}) {
```

```
  return new Promise((resolve, reject) => {
```

```
    axios.post(url, params)
```

```
      .then(res => {
```

```
        resolve(res.data)
```

```
      })
```

```
      .catch(err => {
```

```
        reject(err.data)
```

```
      })
```

```
  })
```

5.配置文件vue.config.js

outputdir: build打包后的目录

productionSourceMap: // 是否在构建生产包时生成sourcdeMap

```
const resolve = function (dir) {
```

```
  return path.join(__dirname, dir);
```

```
}
```

chainwebpack: config => {} // 可以配置简写目录 需要搭配resolve

函数

6.vue-router

hash模式： 使用 URL 的 hash 来模拟一个完整的 URL，于是当 URL 改变时，页面不会重新加载

路由拦截： 写在main.js中

```
router.beforeEach((to, from, next) => {  
  // 处理路由拦截后的逻辑 比如根据token是否存在或者账号是否  
  有当前页面权限 进行拦截  
})
```

7.vuex (vue状态管理模式)

```
state:{name:'helloVueX'} // 存放的键值对就是所要管理的状态  
mutations:{fn(state){}} // state成员操作 组件中使用  
this.$store.commit('fn', 值)  
getters: // 加工state成员给外界  
actions: // 异步操作 异步执行完后执行commit方法 相当于统一处理以免  
异步导致数据返回顺序有误  
modules: 模块化状态管理
```

8.组件使用

在components中创建组件文件xxx.vue 然后再需要使用到的组件中使用
import 引入 在组件的components 方法中注册

9.暗号来了哦

男枪吃大油条