

一. SSR

<https://ssr.vuejs.org/zh/guide/#%E5%AE%89%E8%A3%85>

二. Nuxt.js

1. Nuxt.js介绍与安装

<https://zh.nuxtjs.org/guide>

`npx create-nuxt-app <项目名>`

服务端渲染， 解决首屏加载速度， 和 seo问题

2. Nuxt.js的配置

<https://zh.nuxtjs.org/guide/configuration>

3. 路由

Nuxt.js 依据 pages 目录结构自动生成 vue-router 模块的路由配置。

(1) 要在页面之间使用路由，我们建议使用<nuxt-link> 标签。支持activeClass,tag

(2)嵌套路由

创建内嵌子路由，你需要添加一个 Vue 文件，同时添加一个与该文件同名的目录用来存放子视图组件。

```
1 pages/  
2 --| film/  
3 -----| nowplaying.vue  
4 -----| comingsoon.vue  
5 --| film.vue
```

(3) 动态路由

必须加下划线 (文件夹也可以加下划线(多级嵌套)， 文件也可以加下划线)

```
1 pages/  
2 --| detail/  
3 -----| _id.vue  
4
```

(4) 获取动态路由参数

```
1 asyncData({params}){  
2   console.log(params.id);  
3 }
```

4. 视图

在layout 里面 写好default.vue 可以认为这是根组件的模板了，

所有的组件都加在里面， 但是有些页面 可能不一样，就可以使用 个性化定制页面。

5. 异步数据与资源文件

(1) 如果组件的数据不需要异步获取或处理，可以直接返回指定的字面对象作为组件的数据。

```
1 export default {  
2   data () {  
3     return { foo: 'bar' }  
4   }  
5 }
```

(2)反向代理的配置（重启服务器）

```
1 npm i @nuxtjs/proxy -D  
2 在 nuxt.config.js 配置文件中添加对应的模块，并设置代理  
3  
4 modules: [  
5   '@nuxtjs/axios', //添加axios  
6   '@nuxtjs/proxy' //添加proxy模块  
7 ],  
8 axios: {  
9   proxy: true  
10 },  
11 proxy: {  
12   '/api': {  
13     target: 'http://example.com',  
14     pathRewrite: {  
15       '^/api' : '/'  
16     }  
17   }  
18 }
```

6. vuex状态树（注意：重启服务器）

(1) 需要添加 store/index.js 文件，并对外暴露一个 Vuex.Store 新的实例
每次访问都要返回一个实例，防止交叉请求状态污染

```
1 import Vue from 'vue'
2 import Vuex from 'vuex'
3
4 Vue.use(Vuex)
5
6 const store = () => new Vuex.Store({
7
8   state: {
9     counter: 0
10   },
11   mutations: {
12     increment (state) {
13       state.counter++
14     }
15   }
16 })
17
18 export default store
```

(2) fetch 方法用于在渲染页面前填充应用的状态树 (store) 数据，
与 asyncData 方法类似，不同的是它不会设置组件的数据。

如果页面组件设置了 fetch 方法，它会在组件每次加载前被调用（在服务端或切换至目标路由之前）。

```
1 export default {
2   async fetch ({ store, params }) {
3     let { data } = await axios.get('http://my-api/stars')
4     store.commit('setStars', data)
5   }
6 }
7
```