正课:

1. SPA

2. Vue-router组件

3. 脚手架

1. SPA: Single Page Application

什么是SPA: 整个应用程序只有一个完整的HTML页面和多个保存不同"页面"内容的组件片段。

SPA应用侦测地址栏中相对路径的变化，选择匹配的组件片段替换唯一HTML文件中的相应区域。

vs 多页面:

请求次数:

单页面应用只在首次加载时请求一次，将一个完整的HTML页面和多个片段都获取到客户端保存。每次跳转时，不再需要向服务端发送请求。——只有一次

多页面应用，每跳转一次页面，都重新向服务器发送请求——多次

页面跳转:

单页面应用，不需要替换整棵DOM树，只需要替换DOM树中局部节点即可——效率高

多页面应用，每次跳转，需要删除并重建整棵DOM树——效率低

相同资源的请求: Bootstrap

单页面应用因为页面没有改变，所以无需重复请求资源

多页面应用，每加载一次页面，都重新请求所需的资源

2. Vue-router组件:

什么是: 在Vue中专门实现单页面应用的组件

何时: 如果用Vue框架实现单页面引用时

如何:

1. 下载并引入vue.js和vue-router.js

2. 定义基本的根组件

<div id="app">

</div>

<script>

new Vue({

el:"#app",

data:{}

})

3. 在根组件内容中，预留可能被其他子组件替换的区域

<router-view/> 为其他子组件的片段占位——路由插槽

何时: 只要页面中可能变化的片段区域，都要用router-view占位

4. 定义多个子组件, 每个组件2部分组成:<template>和<script>

5. 定义路由: 3步:

1. 定义路由字典:

路由字典: 包含相对路径和组件间对应关系的集合

var routes=[

{path:"/index", component: Index },

{path:"/login", component: Login }

];

2. 创建路由器对象，装载路由字典

var router=new VueRouter({ routes })

3. 将路由器装入根组件new Vue中

new Vue({

...,

router

})