正课:

1. 脚手架:

1. 脚手架:

什么是: 具有核心功能的半成品代码

我们只要向其中添加个性化自定义内容即可。

为什么: 简化和标准化项目结构

何时: 今后只要开发Vue项目，都是用脚手架生成

如何:

创建项目:

1. 下载Vue-CLI脚手架命令行工具

Vue-CLI 专门用于通过命令生成脚手架代码

先安装命令行工具: npm i -g @vue/cli

2. 用命令行工具创建项目代码:

vue create 项目名

选Manually select features 手动选择功能

选 Babel: 专门自动将ES6代码转化为ES5代码

比如: ES6 中 抛出模块: export default{ ... }

Router: Vue-router组件

Vuex: 客户端状态管理——下一个老师讲

Use history mode for router 选Y

采用不带#的路由地址

选 In package.json 将所有子工具的配置，都集中放在一个配置文件中(package.json)

Save this as a preset for future projects? 选N

是否将当前项目配置保存为默认项目配置

Use https://registry.npm.taobao.org for faster installation?

是否使用国内的淘宝镜像作为包管理工具，选n

3. 运行测试项目:

cd 项目文件夹

项目文件夹下> npm run serve 启动临时开发服务器，编译并宿主Vue生成后的文件。

4. 安装必要组件: axios, bootstrap, jquery

npm i -save axios

npm i -save querystring

脚手架项目结构:

1. public：专门保存不需要编译的静态资源的文件夹

比如可复制学子商城项目的img文件夹到public下

还保存着唯一的完整静态页面: index.html

共用的css和共用的js

比如:

|public

|--css

|--bootstrap.css

|--base.css

|--img

|--js

|--jquery-3.2.1.js

|--bootstrap.js

|--qs.min.js

2. src: 包含项目所有要编译的源代码文件

2.1 assets: 需要编译打包的静态资源

2.2 views: 所有直接出现在路由字典中的组件

2.3 components: 所有全局组件

2.4 app.vue: 整个项目中唯一根组件

2.5 main.js: 是根组件app.vue的主程序

2.6 router.js: 整个项目的路由器和路由字典

3. 每个组件的内容:

每个组件都是一个.vue文件

包含三部分内容:

<template>

<div>

原template中的HTML片段

</div>

</template>

<script>

export default {

原组件对象中的属性: data, mounted(), methods...

}

</script>

<style scoped>

//当前组件自用的CSS样式

</style>

axios的使用:

1. 在main.js中引入axios模块，并将axios添加到Vue类型的原型对象中:

//ES6引入模块

import axios from 'axios'

//相当于const axios=require("axios")

Vue.prototype.axios=axios;

2. 坑: axios请求默认不带cookie！导致无法使用服务器端session。

解决: 配置axios，让其携带cookie:

axios.defaults.withCredentials=true

3. 坑: 如果服务端采用CORS跨域，默认无法接受验证信息的

服务端:

安装cors模块：npm i -save cors

express主模块app.js中，配置cors中间件，允许接受身份认证信息：

app.use(cors({

origin:'http://localhost:8080',

credentials:true

}))

4. 调用axios发送请求:

在任意.vue中:

this.axios.get/post( ... )