**今天学习的内容**

**1:复习上周知识重点**

**2:脚手架项目运行流程**

**-vue cli 运行简版**

**index.html --唯一项目访问页面(#app)**

**vue\_oo/public/index.html {位置可以修改}**

**<div id="app">全所项目内容</div>**

**-main.js --程序入口文件**

**(加载所有组件/资源/将所有组件渲染 #app)**

**App.vue --根组件**

**router.js --路由文件**

**new Vue({**

**router, --路由对象**

**render: h => h(App)**

**}).$mount('#app')**

**(1) -render渲染 h 参数相当**

**(2) document.createElement();**

**(3)将根组件显示到 index.html**

**(4) <div id="app">根组件</div>**

**#所有项目完成,使用 webpack打包工具**

**#所有资源图片js/css/组件 app.js 文件**

**#用户看 index.html app.js**

**-脚手架(难度)**

**(1)package.json 配置文件**

**"scripts" 启动;编译;脚手架命令**

**“dev”:"webpack-dev-server --progress --config"**

**脚本名 启动脚手命令 显示编译进度条 读取默认配置文件**

**dependencies 在运行时依赖第三方组件(生产环境)**

**devDependencies 在编译时依赖第三方组件(开发环境)**

**-vue-style-loader 加载处理样式文件**

**-css-loader 加载样式文件**

**-file-loader 加载图片..**

**"engines": {**

**"node": ">= 6.0.0", 项目运行环境node.js 大于 6.0**

**}**

**browserslist**

**"not ie <= 8" 项目运行在浏览器要求 > IE 8**

**注意: npm i mint-ui --save-dev**

**将mint-ui下载并且将内容配开发环境**

**npm i mint-ui --D**

**3:复习HTTP协议-(调错看不懂出错信息) 背下来**

**HTTP作用:传输网页资源(html;css;js;img;avi;) 网页内容搬运工**

**工作方式:请求(前)和响应(后)**

**请求消息三部分:请求启始行;请求头;请示主体**

**响应消息三部分:响应启始行;响应头;响应主体**

**GET /**

**GET:请求方式[GET;POST;PUT;HEAD;DELETE;CONNECT;TRACE;OPTION]**

**/ 请示地址**

**GET 获取服务器资源**

**POST 发送数据服务器**

**区别:语义;数据量;安全;中文**

**PUT 更新服务器上数据**

**DELETE 删除服务器上数据**

**HEAD 获取服务器资源[只要响应头 不要响应主体]**

**响应消息:响应状态码**

**100-199**

**200-299 响应成功**

**300-399 重定向 http://www.sun.com**

**https://www.oracle.com/sun/index.html**

**400-499 客户端出错**

**500-599 服务器出错**

**5:今天的目标**

**5.1:学子商城--vue ui-购物车(computed)**

**-服务器发送模拟数据: id;price;count;title**

**-接收服务器显示组件,样式**

**-为商品数量按钮绑定点击事件**

**建议:为按钮绑定点事件传递参数商品编号**

**依据编号修改数组对应数量**

**-如何计算合计**

**(1)函数调用**

**getSubTotal(){.... return }**

**{{ getSubTotal }}**

**(2)计算属性:效率更好**

**computed:{**

**getSubTotal(){**

**... retrun...**

**}**

**}**

**{{ getSubTotal }}**

**常见错误**

**1:端口3000 被占有 Error: listen EADDRINUSE :::3000**

**解决:关闭以前控制台程序->终端{终止终端}**

**2: this.getCartList is not a function**

**解决:拼写错误**

**5.2:学子商城--vue ui-商品详情-加入购物车(vuex 重点)**

**-参数传递**

**-将商品编号和数据发送服务器(程序)保存**

**服务器程序:编号数据存储(数据库)**

**(1)实例: INSERT INTO xz\_cart VALUES(null,pid,count,ctime);**

**(2)获取参数:返回添加成功**

**脚手架**

**-将App.vue 组件购物数量+1**

**遇到问题多个组件共享一个数据 购物车中商品数量**

**CartInfo.vue App.vue ...**

**如果使用以前技术:父子组件参数传太复杂**

**5.3: 加入购物车:解决方案 Vuex**

**Vuex 是一个仓库,Vuex核心对象store就是容器**

**此容器包含应用中共享数据**

**store保存共享数据,监听数据状态一但发生改变，通知所有**

**相关组件，更新数据状态**

**保证所有组件使用共享数据状态统一**

**5.4: 购物车数量:解决方案 Vuex如何使用**

**1:创建 Vuex对象**

**1.1 state 属性:容器存储共享数据**

**1.2 mutations属性:操作(修改)容器中存储共享数据**

**1.3 getters 属性:操作(获取)容器中存储共享数据**

**2:将Vuex对象注册Vue实例**

**3:调用Vuex数据**

**(1)mutations 属性 this.$store.commit("方法名称")**

**(2)getters 属性 {{$store.getters.方法名}}**

**4: mutations 方法可以增加一个参数**

**increment(state,c){**

**state.count+=c;**

**}**

**this.$store.commit("increment",3)**

**#全局状态Vuex 刷新页面丢失数据,如果需永久保存**

**#html5 localStorage**

**常见错误**

**(1) property or method "$Store" is not defined**

**解决问题:大小写问题**

**{{$Store.getters.optCount}}**

**{{$store.getters.optCount}}**

**new Vue({**

**store:store**

**})**

**5.5:学子商城--vue ui-商品图片列表**

**-修改连接home "支付" -> 图片列表**

**-创建图片列表组件 photo/PhotoContainer.vue**

**/home/photo**

**-mui 卡片**

**-mint-ui lazy-load**

**-服务器 /imagelist**

**遇到问题: lazy-load 没有出现加载图片旋转图片**

**解决方案:**

**(1)mint-ui有一个BUG如果按需引入组件不显示**

**完整引入**

**import MintUI from "mint-ui";**

**Vue.use(MintUI)**

**5.6:学子商城--vue ui-用户登录**

**服务器:**

**-创建用户表 xz\_user1(id;uname,upwd)**

**id INT**

**uname VARCHAR(25)**

**upwd VARCHAR(32)**

**#所用户密码需要加密后保存**

**id 1001 uname fafafa upwd 4c904kjd843 bank:25**

**-服务器程序**

**脚手架:**

**作业1:创建组件/homg/LoginContainer.vue**

**/home/login**

**作业 2:表单**

**5.x:Element-UI**