程涛 taonetwork2018

# 1:第四阶段课程安排

vue移端动端组件库和项目 6day

html5新特性(视频/绘图) 7day

微信公众平台-小程序和项目 6day

混编 1day

## 2:vue ui组件库

vue 当前很火前端框架vue用户群非常多,

vue 开始针对PC端设备而不针对移动设备开发

pc端与移动端区别:

-屏幕宽度: <=750px

-操作方式:pc 鼠标 移动 手指

国内团队:饿了么:基于vue开发移动端组件库 mint-ui

mui 移动组件库

## 3:vue ui使用组件库

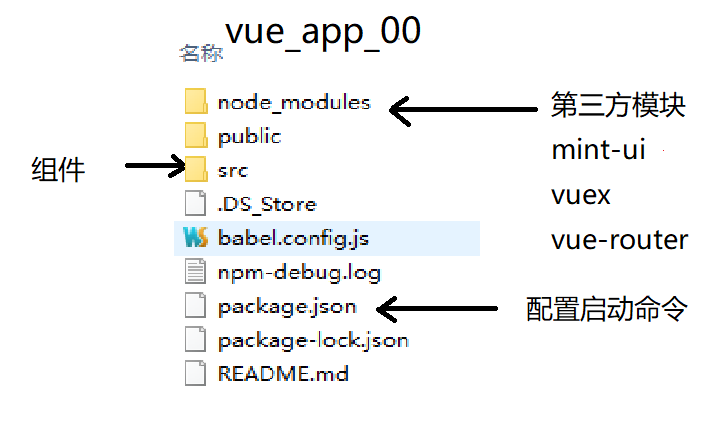
-下载组件 js;css;fonts 在项目中即可使用

-使用脚手架 下载配置使用

\*脚手架管理非常多第三方模块

\*webpack将项目打包

## 4:vue 脚手架



-node\_modules 第三方模块

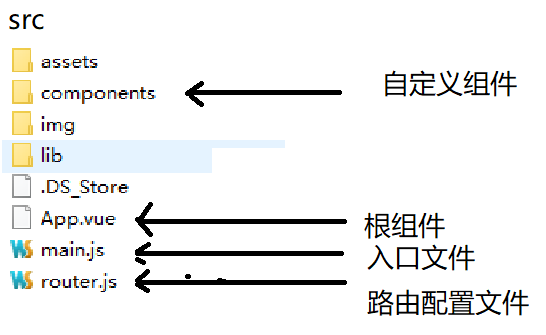
mint-ui

-package.json

"script":{

"serve":"...."

}



## 5:vue 脚手创建新组件

-创建空组件

src/components/Test.vue

-为空组件配置访问路径 router.js

/Test

-测试

http://127.0.0.1:8080/#/Test

练习:创建组件List.vue

位置:src/componets/List.vue

访问路径: /List

在data中保存数组数据其中3用户对象 {id:1,name:"tom"}

并且将用户对象值显示模板中

常见错误:

1: ERROR Failed to compile with 1 errors

error in ./src/router.js

This relative module was not found:

./components/Test1.vue

错误: compile 编译出错(写书代码出错)

2: ERROR Failed to compile with 1 errors

tag <template> has no matching end tag

原因:<template></template>

## 6:为脚手架添加组件库:

移动端组件库(mint-ui)-顶部导航条

官方网站: http://mint-ui.github.io/#!/zh-cn

(1)下载mint-ui组件库

npm i mint-ui -S

(2)在main.js 引入组件(按需加载)

import {Header} from "mint-ui"

(3)在main.js 注册组件

Vue.component(Header.name,Header);

(4)在指定组件使用组件 List.vue

<template>

<mt-header title="学子商城"></mt-header>

</template>

#注意:css文件单独引入

## 7:在脚手架中引入Button组件

(1)在main.js 引入Button

import {Header,Button} from "mint-ui"

(2)在main.js 注册 Button

Vue.component(Button.name,Button);

(3)在指定组件中使用Button List.vue

<mt-button>登录按钮</mt-button>

## 8:vue ui学子商城--用户登录

(1)创建表 xz\_login 用户登录表

(2)创建服务器端程序 app\_server\_00/app.js

(3)脚手架创建 Login.vue

### 8.1 vue ui学子商城--创建表app\_server\_00/db.sql

xz\_login

id INT PRIMARY KEY AUTO\_INCREMENT

uname VARCHAR(25)

upwd VARCHAR(32)

#mysql函数 md5('123')=>8sxj23jksd89xcjkw2

1 tom 202cb962ac59075b964b07152d234b70

#用户登录判断

--用户输入 tom 123

--程序 md5(123) 202cb962ac59075b964b07152d234b70

### 8.2创建服务器端程序 app\_server\_00/app.js

功能一:用户登录

-客户请求方法GET 客户请求路径 /login

-获取客户参数 用户名/密码 uname;upwd

-sql

SELECT id FROM xz\_login

WHERE uname = ? AND upwd = md5(?)

-返回数据格式 json

{code:1,msg:"登录成功"}

{code:-1,msg:"用户名或密码有误"}

常见错误

1: 该网页无法正常运作

node.js 程序己经停止工作出错

解决问题:查看node.js 窗口出现信息

res.send(status, body):: Invalid status code

res.send()中间只能字符串;数组;js对象

2: Error: Can't set headers after they are sent.

原因:你在程序中一个流程执行二次 res.send();

res.send(); 功能:发送数据+再也不发数据停止工作

res.send()

解决问题:将前面send注释

3: Error: listen EADDRINUSE :::3000

原因: 有下程序正在运行占用3000端口

再启动程序占用3000 报错地址冲突

解决:关闭原先程序 vs code 点击回收站

### 8.3脚手架创建 Login.vue

1:打开脚手架创建空组件

src/components/tabbar/Login.vue

2:配置路径 /Login

作业:学子商城--用户登录

1:复习昨天重点

# 2:今天的目标

## 2.1:学子商城--用户登录

-为登录按钮绑定点击事件 @click="btnClick"

为用户和密码双向绑定

-btnClick

获取用户名和密码

发送ajax

this.axios.get();

## 2.2:提示用户信息 登录成功 用户名格式不正确

mint-ui Toast("提示消息");

-由于此功能不是每个组件都使用

-在main.js 引入注册全局组件,所有自定义组件都要使用

-局部组件使用Toast

(1)在当前组件 import {Toast} from "mint-ui"

(2)Toast("消息")

## 2.3:发送ajax 使用axios第三方模块

-下载第三方模块 npm i axios

-在main.js 引入模块

import axios from "axios"

-在main.js 配置跨域访问保存session值选项

axios.defaults.withCredentials=true

-在main.js 将axios配置vue实例中

Vue.prototype.axios = axios

-在当前组件库中使用axios

this.axios.get(url).then(result=>{})

常见错误:

- Access-Control-Allow-Origin 跨域访问

脚手架8080 --> node.js 3000 端口不同

解决问题

-node.js下载第三方模块 cors

-app.js 引入模块

const cors = require("cors");

-在服务器配置允许来自8080所有请求

app.use(cors({

origin:["http://127.0.0.1:8080","http://localhost:8080"],

credentials:true

}))

origin:允许访问列表

credentials:每次访问验证

常见错误:

1:用户名无论正确或者错误,服务器都是错误

-分析原因

-解析:F12 network控制面板

## 2.4:学子商城--商品列表

功能:用户选中商品列表组件显示商品列表(一页)

点击[加载更多]显示下一页...

完成上述功能

-数据库表

-node.js 程序

-脚手架

## 2.5:学子商城--商品列表-数据库表[多表查询]

1:xz\_laptop lid[商品编号]/lname[商品名称]/price[价格]

2:xz\_laptop\_pic laptop\_id ,md[商品中图]

多表查询(套路)

-将表名保存from关键字后面

from xz\_laptop,xz\_laptop\_pic

-为每张表起一个别名

from xz\_laptop l,xz\_laptop\_pic p

-查找两张表关系(二张表某一列 相等 大于 小于)

WHERE l.lid = p.laptop\_id

最终结果

SELECT l.lid,l.lname,l.price,p.md

from xz\_laptop l,xz\_laptop\_pic p

WHERE l.lid = p.laptop\_id

GROUP BY l.lid

LIMIT 0,7;

小结:多表查询方法使用三步套路

多行相同数据只取第一个 GROUP BY

分页 LIMIT

-node.js 程序学子商城--商品列表

获取脚手架发送二个数据 第几页 一页几行

查询数据库内容并且将结果返回

-参数 pno[第几页] pageSize[一页几行]

默认值 pno=1 pageSize=4

-sql 现成

-json {code:1,data:[..]}

常见错误:

1: 无法访问此网站

原因:node.js 出错停止工作

解决:看node.js 窗口出错信息

SQL syntax sql语法出错 在附近 LIMIT

## 2.6:学子商城--商品列表-脚手架

-分析设计外观

最外层父元素弹性布局

子元素弹性布局

子元素宽度 49%

-创建空组件 src/componet/tabbar/GoodList.vue

-指定组件访问路径 /GoodList

#项目很多图片:商品图片,广告..

#所有图片都保存服务器 node.js

作业1:商品列表完成

1:复习昨天的重点

# 3:今天的目标

## 3.1:完成学子商城--商品列表(mint-ui)

发送ajax请求获取商品列表

1:什么发送请求 [打开组件时候 1页][点击按钮 2 3]

2:在哪实现功能 当前组件 created()

通用方法:

-先创建方法 loadMore() 加载更多

-created(){this.loadMore()}

-点击加载更多按钮 this.loadMore();

## 3.2:完成学子商城--商品列表-添加按钮(mint-ui)

mint-ui button

-main.js 引入Button组件

-main.js 注册 Button组件

-<mt-button>加载更多</mt-button>

-为button组件绑定点击事件,完成加载下一页功能

#解决问题：加载一页完成追加功能

#数组拼接操作

[一页 二页]

var rows = this.list.concat(result.data.data);

this.list = rows;

## 3.3:完成学子商城--首页(mint-ui/mui)

#主要完成功能 脚手架

-创建首页组件 src/componets/tabbar/Home.vue

-指定首页路径 /Home

(1)顶部导航条

(2)轮播图

(3)九宫格

(4)tabbar

## 3.4:完成学子商城--首页-顶部导航条

-mint-ui组件库 header 组件

-main.js 引入组件 import {Header} from "mint-ui"

-注册组件 Vue.component(Header.name,Header)

-在模板中调用 <mt-header></mt-header>

属性:

title="显示文字"

fixed 固定在顶部

## 3.5:完成学子商城--首页-轮播图

-引入组件 main.js

import {Swipe,SwipeItem} from "mint-ui"

-注册组件 main.js

Vue.component(Swipe.name,Swipe);

Vue.component(SwipeItem.name,SwipeItem);

-在模板调用 Home.vue

<mt-swipe>

<mt-swipe-item> 1 img..</mt-swipe-item>

...

</mt-swipe>

#属性

:auto="4000" 自动播放的时间间隔

#注意:默认轮播图高度为 0

## 3.6:完成学子商城--首页-九宫格

-九宫格组件库 MUI 独立库与VUE关系不是绑定

-下载对应文件 mui [css;js;fonts]

-src/lib/mui/....

-main.js 直接加载css文件即可

import './lib/mui/css/mui.css'

import './lib/mui/css/icons-extra.css'

-在对应组件添加标签和样式

## 3.7: 扩展知识-图片宽度

公司前端设计师：[PS]设计效果图

前端开发工程师: html/css/js/vue...

物理像素:设计效果图中像素 120px

逻辑像素:程序中使用中像素 60PX

#智能手机:iphone4 技术标准

#手机系列:苹果/安卓

#公司工作:效果图片 iphone6

#iphone6 逻辑像素与物理 2x DPR

## 3.8:完成学子商城--首页-tabbar

mui 库

复制 tabbar.html 标签到 Home.vue组件

## 3.9:学子商城项目小结

-用户登录

表 xz\_login[id,uname,upwd] md5()

1 tom 202s03kd93827899b07

app.js app.get("/login",(req,res));

\*参数 uname;upwd

\*sql SELECT id FROM xz\_login

WHERE uname = ? AND upwd = md5(?)

\*json {code:1,msg:"登录成功"}

{code:-1,msg:"用户名或密码有误"}

Login.vue

<form>

<input type="text" v-model="uname" />

..

<input type="button" value="" @click="loginBtn" />

</form>

Toast("提示")

-商品列表

表:xz\_laptop(商品表) xz\_laptop\_pic(商品图片表)

app.js:

\*参数:pno 页码 pageSize 页大小(一页几行)

\*sql:SELECT l.lid,l.lname,l.price,p.md

FROM xz\_laptop l,xz\_laptop\_pic p

WHERE l.lid = p.laptop\_id

LIMIT ?,?

?:offset = (pno-1)\*pageSize;

?:pageSize = parseInt(pageSize)

\*json:{code:1,data:[]}

GoodList.vue

<div class="item">

<img src="" />

<span>999.00</span>

</div>

-loadMore 加载下一页数据

\*显示第一页第二页数据.. concat 数组拼接

\*点击下载更多按钮

\*created()

-pno:0 list:[]

-首页

-表

-app.js

app.get("/imglist") 轮播图

app.get("/grid") 九宫格

-Home.vue

\*顶部导航栏 mint-ui Header

\*轮播图 mint-ui Swipe/SwipeItem

\*九宫格 mui/grid\_default.html

\*tabbar mui/tabbar.html

## 3.10:学子商城项目--新闻列表

表:xz\_news

id INT

img\_url VARCHAR(255)

title VARCHAR(225)

ctime DATETIME

point INT

content VARCHAR(20000)

作业:学子商城新闻列表

1:复习昨天重点

# 4:今天学习目标

## 4.1:学子商城--新闻列表

分析功能:创建新闻列表组件 NewsList.vue

-创建新组件 src/component/home/NewsList.vue

-为组件指定路径 /NewsList

-修改组件外观 mui/media-list.html

-组件创建成功后发送第一页请求

-点击组件"加载更多"按钮 发送下一页请求

## 4.2:学子商城--新闻列表-日期格式错误

网页通常日期格式货币格式不满足要求使用过滤器

解决问题

格式:

\*main.js创建全局过滤器其它任意组件都可以直接调用

Vue.filter("过滤器名称",function(原始数据格式){

对原始数据格式处理

返回字符串

})

\*在模板中直接调用过滤器

{{item.ctime | 过滤器名称}}

## 4.3:组件之间跳转与数据传递

(1)组件之间跳转

方式一:标签

<router-link to="组件路径">文字</router-link>

示例:Exam01 -> Exam02

<router-link to="/Exam02">跳转组件</router-link>

方式二:编程

this.$router.push("组件路径");

示例:

<img src="1.jpg" @click="jump"/>

function jump(){

this.$router.push("组件路径")

}

(2)数据传递

发送数据:

方式一:<router-link :to="'/Exam02?nid='+item.id">...

方式二:this.$router.push("/Exam02?nid=5")

接收数据: this.$route.query.nid

#最早接收数 created()

## 4.4:学子商城--新闻详细--评论列表(子组件)

-创建空组件 src/component/home/NewsInfo.vue

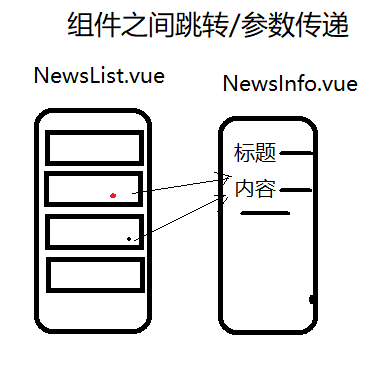
-为此组件绑定路径 /NewsInfo

-显示新闻详细信息 mui/card.html

-当用户点击新闻列表中某条新闻

跳转NewsInfo.vue 组件nid=3

-将当前新闻编号传递NewsInfo.vue组件



# 5:今天的目标

## 5.1:学子商城--新闻详细

技巧1:SQL

var sql = "INSERT INTO xz\_comment VALUES(null,?,?,now())";

1年项目升级,在表xz\_comment添加3列

var sql = "INSERT INTO xz\_comment(id,nid,content,ctime)";

sql+="VALUES(null,?,?,now())";

技巧2:post请求 vue 使用ajax库 axios

#原因 axios 处理post请求参数有一个内部问题.

#第三方式 qs 将js对象转换字符串

1:在main.js 引入qs 模块,将模块注册Vue实例

import qs from "qs"

Vue.prototype.qs = qs;

2:发送post数据将数据先创建js对象再用qs转换

字符串.

var postData = this.qs.stringify({

nid:3,

content:"..."

});

this.axios.post(url,postData).then(...)

## 5.2:学子商城--购物车(重点)

购物车:功能保存用户希望购买商品，需要长期保存

数据库;cookie;...

分析购物车外观

(1)mui card.html

(2)mui media-list.html

开始流程:

-先创建组件 ShopCart.vue /ShopCart

## 5.3:学子商城--购物车(重点)--删除指定商品

-数据库表:

xz\_cart[id, uid,pid,pname, price]

-node.js 程序:

/removeItem

参数:id购物车id

sql:DELETE FROM xz\_cart WHERE id = ?

-脚手架组件:

[button] onclick ajax

脚手架删除当前商品

-常见错误

(1)node.js NaN id=undefined

原因:数据从组件没有发送node.js

post请求

解决:F12->NETWORK->Header->response

Header标签

FormData post请求发送数据格式与内容

id:3

作业1:

-全选复选框[全选或者清空商品复选项];

-单独选中某个商品

-删除选中商品

-清空购物车

1:复习昨天重点

1.1:用户登录

分析功能:判断当前操作是否是网页正式用户,

浏览用户

用户输入用户和密码在系统数据库判断

用户输入是否正确

-数据库表 xz\_login[id,uname,upwd]

-app.js

\*参数 uname;upwd

\*sql SELECT id FROM xz\_login WHERE uname=?

AND upwd = md5(?)

如果判断查询是否成功 result.length==0

\*json {code:1,msg:"登录成功.."}

-Login.vue

获取用户参数[v-model]双向绑定

发送ajax请求 ??

1.2:商品分页显示

分析功能:用户浏览商品信息,尽可能多的显示商品

分页

-数据库表 xz\_laptop[商品表] xz\_laptop\_pic[商品图片]

SELECT l.lid,l.price,p.md

FROM xz..

LIMIT ?,? //?起始记录数 0 1 2

//?每页有几条 5

-app.js

\*参数 pno[页码] pageSize[页大小]

\*设置默认 pno=1 pageSize=7

\*sql

\*json {code:1,data:result}

-GoodList.vue

pno:0

loadMore(){this.pno++;ajax}

1.3:发表评论

分析功能:新闻详情组件显示当前新闻内容

用户可以对新闻发表评论..

-数据库表 xz\_comment[id,nid,content,ctime]

-app.js

app.post("/addcomment",(req,res))

\*参数:nid[新闻编号];content[评论内容]

req.body.nid req.body.content

\*INSERT INTO xz\_comment VALUES(null,?,?,now());

INSERT INTO xz\_comment(id,nid,content,ctime)

VALUES(null,?,?,now())

\*判断条件执行是否成功

result.affectedRows 影响行数 0

\*{code:1,msg:"删除成功"}

-NewsInfo.vue axios组件发送post请求问题

main.js 引入qs

import qs from "qs"

Vue.prototype.qs = qs;

var postData = this.qs.stringify({

nid:1,

content:".."

});

this.axios.post(url,postData)...

1.4:购物车[列表/删除指定购物车中商品内容]

分析功能:临时保存用户喜欢商品

将用户商品保存购物车表中

-数据库表 xz\_cart[id,uid,pid,pname,price]

-app.js

\*参数 id 购物中编号

\*sql DELETE FROM xz\_cart WHERE id = ?

\*json {code:1,msg:"删除成功"}

-ShopCart.vue

axios.post(....)

#结论:如果有数据需要传递事件处理函数

#有数据在当前元素添加自定义属性

#<input type="text" :data-idx="item.id"

# :data-age="item.age"

#@click="btnHandle"

btnHandle(e){

e.target.dataset.age;

e.target.dataset.idx;

}

<input type="button" @click="removeItem"

value="删除"/>

removeItem(e){

e事件对象:浏览器自动创建对象(包含与事件相关)

target 触发事件对象

e.target.dataset.id

}

1.5:F12 打开浏览器控制台

Elements:程序在网页中(内存)运行结果

css/js 语法

Console:控制台 123

NetWork:网络

# 6:今天的目标

## 6.1:学子商城--购物车--(重点) 删除多个商品

-数据库:sql xz\_cart[id,uid,pid,pname,price]

-node.js app.js

\*参数 ids:"4,6,9"

\*sql DELETE FROM xz\_cart WHERE id IN (ids);

\*json {code:1,msg:"删除成功"}

http://127.0.0.1:3000/removeItem?ids=5,6,8

-ShopCart.vue 传递参数多

<input type="checkbox"

:checked="item.cb" :data-id="item.id" />

#分析功能删除选中多个商品

#this.list = [{cb:false,id:1,pname,price},

{cb:true,id:2,pname,price},];

[] mac 10000

[\*] 华为 11000

{删除选中商品}

函数结果拼接字符串 "1,3" 发送请求

解决问题一:商品选中状态

-cb当前商品选中状态:false 没选中 true选中

-商品加载时默认false

-loadMore 获取list 先循环result.data.data

对象添加cb属性值false,再保存list

#顺序(注意)

## 6.2:小结购物车组件

分析总体功能:

(1)要求用户登录成功后操作购物车

#用户登录成功,服务器保存一个值 uid=1

#购物车保存用户喜欢商品

#用户操作

-将一个商品添加至购物车

-显示购物车中商品列表

-删除购物车中某一个商品

-删除购物车中多个商品

-方便用户:全选按钮

## 6.3:小结购物车组件-列表

分析:当用户打开购物车页面显示他添加商品列表

表:xz\_cart[id,uid,pid,pname,price]

app.js

参数(没有): uid 保存服务器node.js

sql:SELECT id,uid,pid,pname,price FROM xz\_cart

json:{code:1,data:[]}

ShopCart.vue

data:{ list:[]}

methods:{loadMore(){.... this.list}}

什么时候调用

created(){this.loadMore()}

## 6.4:小结购物车组件-删除指定一件商品

表:xz\_card[id,uid,pid,pname,price]

app.js

参数:id

sql: DELETE FROM xz\_cart WHERE id = ?

json: {code:1,msg:"删除成功/失败"}

ShopCart.vue

<input type="button"

value="删除" :data-id="item.id" @click="removeItem"/>

removeItem(e){

1:获取当前购物车id e.target.dataset.id

2:发送ajax请求 this.axios.post(...)

}

## 6.5:小结购物车组件-删除多个商品

分析:用户选择某一些商品删除 [\*]

点击按钮[删除选中商品]

表:xz\_cart [id,uid,pid,pname,price]

app.js

参数:删除多个商品,获取多个商品id "1,3,5"

var ids = req.query.ids; //1,3,5

sql: DELETE FROM xz\_cart WHERE id IN("+ids+")

json:{code:1,msg:"删除成功"}

ShopCart.vue

修改loadMore 加载购物车列表

loadMore(){

#为购物车列表中对象添加属性 item.cb=false

#当前商品前 复选框状态

}

<input type="checkbox" :checked="item.cb" />

添加modifyItem

modifyItem(e){

#当用户选中某个商品,修改list数组对应对象

#item.cb=true

e.target.dataset.i //当前元素数组下标

}

<input type="checkbox" @click="modifyItem" :data-i="i"/>

添加delItem 删除选中商品按钮

delItem(){

this.list[{id:1,cb:true},{id:2,cb:false},{id:3,cb:true}]

"1,3,4,"

substring(起始下标,几个);

this.axios(url)

}

## 6.6:小结购物车组件-全选

--添加 selectAll 全选

<input type="checkbox" @click="selectAll" />全选

selectAll(e){

var rs = e.target.checked;

for(var item of this.list){

item.cb = rs;

}

}

--添加 selectAll 全选/清除判断

<input type="checkbox" @click="selectAll" :checked="allcb" />

data:{

allcb:false //希望全选状态和变量绑定

}

selectAll(e){

var rs = e.target.checked;//复选框状态

allcb = rs; //复选框状态变量同步

}

modifyItem(){

商品前复选框:需要计算操作

1:数组个数 14

2:数组中选中元素个数 cb==true 14

if(size1 == count){

this.allcb = true;

}else{

this.allcb =false;

}

}

作业1:

1:商品列表 GoodList.vue

2:商品详情 GoodInfo.vue

-显示当前商品信息 标题/价格/...

-点击"加入购物车"

3:html5新特性... 7天(小游戏)