**WEB第五阶段DAY08 — uni-app**

复习：

1、uni-app是一个基于Vue.js语法的MVVM框架，用于“需要多端适配的移动端项目”

2、单文件组件中模板（template）

1) 不允许使用HTML标签；可用组件分为三类：

①内置组件(高仿微信小程序)：视图(view)、滚动视图(scroll-view)、文本(text)、富文本(rich-text)、图片(image[mode])、输入框(input[type])、按钮(button[type])、导航器/超链接(navigator[open-type])、轮播广告(swiper)、数据拾取器(picker)、单选框组(radio/radio-group)、复选框组(checkbox/checkbox-group)

②扩展组件(uni-ui库)：评分(uni-rate)、图标(uni-icons)、宫格(uni-grid[cols]/uni-grid-item)、卡片(uni-card)、列表(uni-list)、导航条(uni-navbar)、弹出层(uni-popup)

③自定义组件(components)： 1）创建 3）使用

2) 数据绑定（高仿Vue.js）：

①内容绑定：{{}}

②属性绑定：v-bind ：

③样式绑定：:style :class

④事件绑定：v-on @

⑤双向数据绑定： v-model

⑥条件渲染：v-if/elseif/else v-show

⑦列表渲染：v-for

3、单文件组件中脚本（script）

脚本基础结构类似于Vue（export default { data/props/methods/computed }），但是可用的对象和方法类似于小程序：

①返回页面顶部：uni.pageScrollTo()

②页面跳转五种方法：uni.navigateTo() / uni.navigateBack() / uni.redirectTo() / uni.switchTab() / uni.reLaunch()

③异步请求服务器端接口：uni.request({url, method, header, data})

④四种对话框：uni.showToast() / uni.showModal() / uni.showLoading() / uni.showActionSheet()

⑤客户端存储：uni.setStorageSync() / uni.getStorageSync() / ...

⑥获取系统信息：uni.getSystemInfoSync()

⑦获取全局App对象：getApp()

4、uni-app中的生命周期方法 —— 面试题

第一组：整个应用级生命周期方法(App.vue)

onLaunch() / onShow() / onHide()

第二组：页面级生命周期方法(pages—pages.json中分配了访问地址的组件)

onLoad(data) / onShow() / onReady() / onHide() / onUnload()

onPageScroll() / onReachBottom() / onPullDownRefresh()

第三组：组件级生命周期方法(components)

beforeCreate() / created()

beforeMount() / mounted()

beforeUpdate() / updated()

beforeDestroy() / destroyed()

**一、uni-app中的模板**

1、view：最简单的容器组件，相当于HTML中的div

2、text：最简单的文本组件，相当于HTML中的span

3、image：用于展示图片的组件，类似于HTML中的img

①默认image的尺寸都是320\*240

②默认情况下，修改图片的宽度，高度不会自动等比例缩放

③可以指定mode="widthFix"，实现“宽度固定，高度自动等比例缩放”

4、input：单行文本输入域

<input type="键盘类型" password="是否显示密码"/>

5、button：按钮

6、navigator：导航器、超链接，用于实现页面跳转

相对地址： <navigator url="../findpwd/findpwd"> 不带.vue

绝对地址： <navigator url="/pages/findpwd/findpwd"> 不带.vue

7、picker：数据拾取器，类似于HTML中的下拉菜单，可以有五种选择：普通、多列、日期、时间、省市区(中国)

8、map：在应用中显示地图；uni-app仅提供该组件，但是不提供地图数据，底层还是要依赖第三方地图提供商——需要注意打包为不同的应用，可用选用的地图提供商

模板中实现数据绑定：

|  |
| --- |
| ①内容绑定：  <text>{{表达式}}</text> |
| ②属性绑定：  <any v-bind:属性名="表达式"/>  <any :属性名="表达式"/> |
| ③样式绑定：  <any :style="{样式属性: 表达式}" :class="{类名:布尔表达式}"/> |
| ④事件绑定：  <any v-on:click="处理方法"/>  <any @click="处理方法(实参列表)"/> |
| ⑤双向数据绑定：  <input v-model="模型变量名"/> |
| ⑥条件渲染：  <any v-show="表达式"/> —— 本质：display: none/block  <any v-if="表达式"/> <any v-elseif="表达式"/> <any v-else/> —— 本质：删除/添加节点 |
| ⑦列表渲染：  <any v-for="(item, idx) in 数组" :key="idx"/> |

**二、uni-app中的样式**

1、uni-app支持的尺寸单位：em、vw/vh、%、px、rpx

2、多个页面或组件共享的样式变量编写在 uni.scss；

3、多个页面或组件共享的样式编写在 App.vue；

4、一些特殊选择器： :not(not){ }、 page{ }；

5、组件中不支持标签选择器，但是页面中可以使用；

**三、uni-app中的脚本**

1、五种路由跳转方法

2、四种提示框

**①吐司对话框** —— 用于简单的提醒，非重要事件的提示

uni.showToast( )

**②模态对话框** —— 用于严重的提示，甚至必须作出选择的提示

uni.showModal( )

**③加载中对话框** —— 提示操作正在进行中

uni.showLoading( ) / uni.hideLoading()

**④动作清单对话框** —— 让用户选择要执行的动作

uni.showActionSheet( )

3、读写客户端数据存储

**在客户端存储数据** —— 类似于localStorage.setItem(k, v)：

uni.setStorage( key, value, success(){ } ) —— 异步保存

uni.setStorageSync( key, value ) —— 同步保存

**读取客户端存储的数据** —— 类似于 let v = localStorage.getItem(k)：

uni.getStorage( key, success(value){ } ) —— 异步读取

let value = uni.getStorageSync( key ) —— 同步读取

**删除客户端存储的一个数据** —— 类似于 localStorage.removeItem(k)：

uni.removeStorage( key, success(){ } ) —— 异步删除

uni.removeStorageSync( key ) —— 同步删除

**删除客户端存储的所有数据** —— 类似于localStorage.clear()：

uni.clearStorage( success( ){ } ) —— 异步删除

uni.clearStorageSync( ) —— 同步删除

4、getApp( )：返回App.vue创建的全局App实例对象

**四、uni-app应用中请求服务器端接口数据**

uni-app使用uni.request()可以实现服务器端数据API的异步请求，编译后会转换为各平台的专有技术。uni.request( )的使用方法：

|  |  |
| --- | --- |
| //没有请求主体的GET请求  let url = ""  let [err, res] = await uni.request({  url  })  if(err){ 处理请求错误 }  else { res.data就是响应消息主体 } | //有请求主体的POST请求  let url = ""  let [err, res] = await uni.request({  url,  method: 'POST', //请求方法  data: {k:v, ....} //请求主体  })  if(err){ 处理请求错误 }  else { res.data就是响应消息主体 } |

**五、重点面试题：uni-app中的生命周期方法**

**第一组：应用程序级生命周期方法**——App.vue：

onLaunch()：应用程序启动，每个应用此方法只能调用一次

onShow()：应用程序显示出来，可以多次调用

onHide()：应用程序隐藏起来，可以多次调用

**第二组：页面级生命周期方法**——pages

onLoad()：页面加载完成，类似于组件的mounted，每个组件只挂载一次；一般用于异步请求页面需要的数据

onShow()：页面显示出来，可以多次调用

onReady()：页面准备就绪，第一次挂载完成后，进场动画执行完成，可供用户使用

onHide()：页面隐藏起来，可以多次调用

onUnload()：页面卸载了，类似于组件的destroyed；一般用于释放页面所占用的资源，例如：定时器、WebSocket....

onPageScroll()：页面滚动了

onReachBottom()：页面滚动到底部了

onPullDownRefresh()：页面在顶部下拉刷新了

**第三组：组件级生命周期方法**——components

beforeCreate() / created()

beforeMount() / mounted()

beforeUpdate() / updated()

beforeDestroy() / destroyed()

**六、uni-app中的页面跳转传参**

|  |
| --- |
| //page1  使用“查询字符串/搜索参数”来传递数据  <navigator url="/pages/page2/page2?pno=5&kw=dell">  uni.navigateTo({ url: "/pages/page2/page2?pno=5&kw=dell" }) |
| //page2  使用onLoad()生命周期方法来读取路由参数  onLoad( data ){  //data： {pno: "5", kw: "dell"}  }  onLoad( {pno, kw} ){  //对上个页面传递来的查询参数进行解构  } |

**七、重要面试题：客户端请求的防抖**

bounce：弹跳

debounce：抑制弹跳，防止抖动

有些事件，可能在短时间内容多次触发，例如：输入框的内容改变、元素拖动、光标移动....

如果这类事件处理方法中需要调用服务器端接口，就可能发生“短时间内接口频繁请求”问题——服务器压力过大，其中很多请求还可能是无效请求。

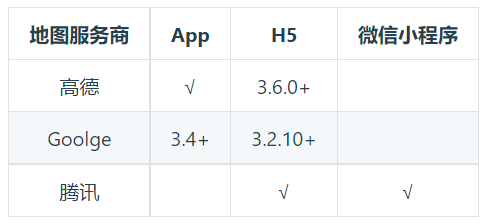
解决方案：前端页面一定要控制请求的频次间隔，例如：事件发生1s后才发起服务器端请求，如果这个1s还没到，再次触发同一个事件，那么重新等待1s... —— 防抖

|  |
| --- |
| let timer = null //全局定时器变量  function doEvent( ){ //事件处理方法 —— 可能被高频次的密集调用  if(timer){ 停止定时器，timer置为空 } //如果之前存在定时器任务，先取消  timer = setTimeout( ()=>{发起服务器端请求}, 1000) //再重新创建定时器任务  } |

**八、使用地图**

使用内置组件<map></map>在应用中显示地图；

uni-app仅提供该组件，但是不提供地图数据，底层还是要依赖第三方地图提供商（例如：高德、谷歌、腾讯）——需要注意打包为不同的应用，可用选用的地图提供商不同：



在应用中使用高德地图的步骤：

①注册高德地图开发者账号： https://lbs.amap.com

②登录高德开发者账户，创建一个高德应用（Web端应用），获得Key和SecurityCode

③在项目的清单文件中（manifest.json）指定启用高德地图，配置Key和SecurityCode

manifest.json > Web配置 > 定位和地图 > 启用“高德地图” > 输入“Key和SecurityCode”

④在页面中使用地图组件

|  |
| --- |
| <map longitude="地图中心点的经度" latitude="地图中心点的纬度" scale="地图缩放级别">  </map> |

课后任务：

① 阅读uni-app手册，掌握map中的markers（地图上的标记物）属性的使用方法，为服务器端返回的六个停车区渲染出六个地图标记物

② 阅读uni-app手册，掌握map中的markers（地图上的标记物）属性的使用方法，点击某个标记物，在其上方弹出说明文字： 天通苑二区停车区空余车位：49