



第三章 UNIAPP与VUE（二）

目录

一、解读 “About “page

1. template结构
2. 文本插值
3. 条件编译
4. this that
5. uni api uni.
6. 回调函数
7. 匿名函数与箭头函数
8. methods的两个方法

二 . tabBar组织页面

三 . subpackage的应用

四 . 路由跳转

— “About” from Hello uniapp project



1.1



扫码体验uni-app

uni-app 是一个使用 Vue.js 开发跨平台应用的前端框架。 1.2

本示例源码获取方式:

1. 下载 HBuilderX, 新建 uni-app 项目时选择 Hello uni-app 模板。
2. <https://github.com/dcloudio/hello-uniapp> 1.3.1

分享

1.3.2

{{version}}

当前版本: 13.8.12

2

<!-- #ifdef APP-PLUS -->

template

view.about

1 view.content

1.1 view.qrcode

image

text.tip

1.2 view.desc

text.code

text.code

1.3 view.source

button

2 view.version

文本插值

```
<!-- #ifdef APP-PLUS -->  
<view class="version">  
    当前版本: {{version}}  
</view>  
<!-- #endif -->
```

- 文本插值是最基本的数据绑定，使用“Mustache”语法（即双大括号）
- 例中，双大括号标签会被替换为相应组件实例中 version 属性的值。同时每次 version 属性更改时它也会同步更新。

```
data() {  
    return {  
        providerList: [],  
        version: ''  
    }  
},
```

关于“条件编译”

- 详见 <https://uniapp.dcloud.net.cn/tutorial/platform.html#preprocessor>
- 以 #ifdef 或 #ifndef 加 %PLATFORM% 开头，以 #endif 结尾。
- 利用注释实现，在不同语法里注释写法不一样，js/uts使用 // 注释、css 使用 /* 注释 */、vue/nvue/uvue 模板里使用 <!-- 注释 -->；
 - ◆ #ifdef: if defined 仅在某平台存在
 - ◆ #ifndef: if not defined 除了某平台均存在
 - ◆ %PLATFORM%: 平台名称

```
<!-- #ifdef APP-PLUS -->  
<view class="version">  
    当前版本: {{version}}  
</view>  
<!-- #endif -->
```

```
//#ifdef APP-PLUS || H5  
uni.chooseVideo({  
    success: (res) => {  
        console.log(res.result);  
    }  
});  
//#endif
```

```
//#ifndef H5  
uni.scanCode({  
    success: (res) => {  
        console.log(res.result);  
    }  
});  
//#endif
```

关键字this that

- this关键字是指当前组件的实例，使用this可以直接访问当前组件中的属性和方法。
- that是一个临时对象，可用于在函数内部保存this的值。在一些情况下，函数内部的this并不是指向当前组件的实例（uniapp中this的动态指向），此时可以使用that来保存this的值，以便在函数内部继续使用。

```
onLoad() {  
  // #ifdef APP-PLUS  
  this.version = plus.runtime.version;  
  uni.getProvider({
```

```
export default {  
  data() { return { items: [] } },  
  methods: {  
    loadData() {  
      let that = this; //保存当前组件的实例  
  
      uni.request({  
        url: 'http://localhost:8080/getData',  
        success(res) {  
          //此处使用that来访问items属性，而不是this  
          that.items = res.data;  
        }  
      })  
    }  
  }  
}
```


uni-app的 js API

```
44 | },
45 |   onLoad() {
46 |     // #ifdef APP-PLUS
47 |     this.version = plus.runtime.version;
48 |     uni.getProvider({
49 |       service: 'share',
50 |       success: (result) => {
51 |         const data = [];
```

API

- <https://uniapp.dcloud.net.cn/api/> ; 在HX中选中api按F1及时获取帮助
- uni-app的 js API 由标准 ECMAScript 的 js API 和 uni 扩展 API 这两部分组成。
- 标准 ECMAScript 的 js 仅是最基础的 js。浏览器基于它扩展了 window、document、navigator 等对象。小程序也基于标准 js 扩展了各种 wx.xx、my.xx、swan.xx 的 API。node 也扩展了 fs 等模块。
- uni-app 基于 ECMAScript 扩展了 uni 对象，并且 API 命名与小程序保持兼容

API 列表

基础

网络

媒体

文件

数据缓存

位置

设备

界面

路由

键盘

第三方服务

广告

平台扩展

其他

- uni.on 开头的 API 是监听某个事件发生的 API 接口，接受一个 CALLBACK 函数作为参数。当该事件触发时，会调用 CALLBACK 函数。
- 如未特殊约定，其他 API 接口都接受一个 OBJECT 作为参数。
- 以uni.getProvider(OBJECT)为例如下

```
uni.getProvider({
  service: 'share',
  success: (result) => {
    const data = [];
    for (let i = 0; i < result.provider.length; i++) {
      switch (result.provider[i]) {
        case 'weixin':
```

OBJECT 参数说明

参数名	类型	必填	说明
service	String	是	服务类型，可取值见下面说明。
success	Function	否	接口调用成功的回调
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

JS的回调函数Callback

- JS函数可以作为一个参数在另一个函数中被调用。
- JS在定时器、建立网络连接、读取网络流数据、向文件写入数据、Ajax提交、请求数据库服务等情况下，实行异步操作，这就是说JS代码的执行顺序并不是从上至下按部就班完成的。
- 在大多数编程语言中，函数的形参总是从外向内传递参数；但在JS中，回调函数表示从内向外反向调用某个外部函数。

匿名函数 与 箭头函数

- 匿名函数：未命名的函数(与之对应的是显函数)。可以当做变量的值、参数值、可以自调用；ES6匿名函数包含了箭头函数。
- 箭头函数的写法：将原函数的“function”关键字和函数名都删掉，并使用“=>”连接参数列表和函数体。当函数参数只有一个，括号可以省略；但是没有参数时，括号不可以省略。箭头函数中，this为词法作用域

()=>{}

(name,age)=>{}

匿名函数的例子

//将匿名函数当做变量的值:

```
let show = function(a,b){ console.log(a,b);//10,20}          show(10,20);
```

```
let stu = {  
  id:1,  
  name:'zhang',  
  show2:function(){ console.log(this.name);}  
}
```

```
stu.show2(); //zhang
```

//将匿名函数当做参数值:

```
function show5(k){ k(); //回调函数 }
```

```
show5( function(){ console.log('a');//a} );
```

//匿名函数自调用:

```
(function(){ console.log('b'); })();
```

```
(function(a,b){ console.log(a + b); })(100,200);//300
```

Methods中的两个函数

```
methods: { save() { }, share(e){}}
```

```
5 | d.net.cn/guide/uniapp/app_download.png" @longtap="save"></image>
```

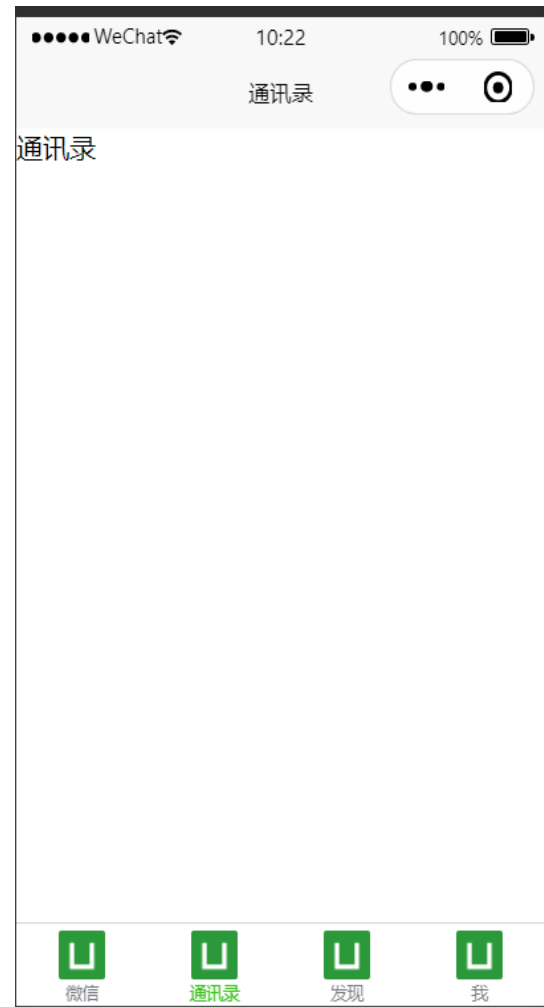
```
26 | <button type="primary" @click="share">分享</button>
```

课堂案例

- tabBar组织页面
- subpackage的应用
- 路由跳转
- 聊天页面（静态）
- 聊天页面（JS动态）

二、tabBar组织页面

- 如果应用是一个多 tab 应用，可以通过 tabBar 配置项指定一级导航栏。在 pages.json 中提供 tabBar 配置，方便快速开发导航，更重要的是在App和小程序端提升性能。在这两个平台，底层原生引擎在启动时无需等待js引擎初始化，即可直接读取 pages.json 中配置的 tabBar 信息，渲染原生tab。
- 准备四个page（pages目录下），注意pages.json的变化，准备4个icon和4个选中icon并放置在static目录的合适位置
 - ◆ 微信（用index页面）、发现、通讯录、我
- 在pages.json中添加根目录级属性tabBar，设置必要的style和数组list，把四个page配置到list中
 - ◆ <https://uniapp.dcloud.net.cn/collocation/pages.html#tabbar>



■ pages.json中tabBar的设置

```
+ "globalStyle": { ... }
- "tabBar": {
  "color": "#7A7E83",
  "selectedColor": "#3cc51f",
  "borderStyle": "black",
  "backgroundColor": "#ffffff",
  "list": [
    {
      "pagePath": "pages/index/index",
      "iconPath": "static/tabbar-icons/logo.png",
      "selectedIconPath": "static/tabbar-icons/logo.png",
      "text": "微信"
    },
    {
      "pagePath": "pages/list/list",
      "iconPath": "static/tabbar-icons/logo.png",
      "selectedIconPath": "static/tabbar-icons/logo.png",
      "text": "通讯录"
    },
    {
      "pagePath": "pages/finder/finder",
      "iconPath": "static/tabbar-icons/logo.png",
      "selectedIconPath": "static/tabbar-icons/logo.png",
      "text": "发现"
    },
    {
      "pagePath": "pages/my/my",
      "iconPath": "static/tabbar-icons/logo.png",
      "selectedIconPath": "static/tabbar-icons/logo.png",
      "text": "我的"
    }
  ]
}
```

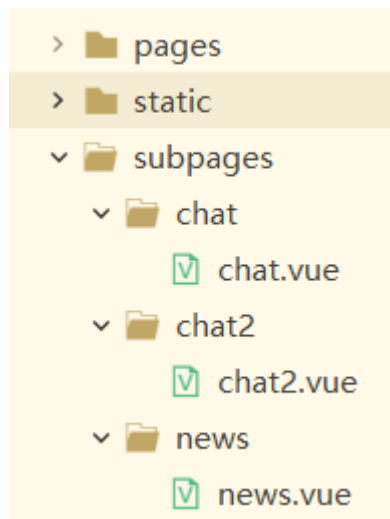
三.小程序分包subPackages

- 因小程序有体积和资源加载限制，各家小程序平台提供了分包方式，优化小程序的下载和启动速度。
- 所谓的主包，即放置默认启动页面/TabBar 页面，以及一些所有分包都需用到公共资源/JS 脚本；而分包则是根据pages.json的配置进行划分。
- 在小程序启动时，默认会下载主包并启动主包内页面，当用户进入分包内某个页面时，会把对应分包自动下载下来，下载完成后再进行展示。此时终端界面会有等待提示。

◆ <https://uniapp.dcloud.net.cn/collocation/pages.html#subpackages>

◆ <https://blog.csdn.net/atu1111/article/details/119027281>

- 根目录下新建“subpages”目录，并在其下新建页面
- 在pages.json中构建和pages同级的subpackages，它只有两个属性，root和pages，在pages下构建新建的页面信息



```
"subPackages": [  
  {  
    "root": "subpages",  
    "pages": [  
      {  
        "path" : "news/news",  
        "style" :  
        {  
          "navigationBarTitleText": "新闻中心",  
          "enablePullDownRefresh": false  
        }  
      },  
      {  
        "path" : "chat/chat",  
        "style" :  
        {  
          "navigationBarTitleText": "与小姐姐聊天",  
          "enablePullDownRefresh": false  
        }  
      }  
    ]  
  },  
  ...  
]
```

四、路由与页面跳转

■ 组件方式页面跳转 与api方式跳转

◆ <https://uniapp.dcloud.net.cn/api/router.html#navigateto>

◆ <https://uniapp.dcloud.net.cn/component/navigator.html#>

■ 之前navigateto404跳回首页的方式需要修改为： uni.switchTab

■ 小程序规定小程序的页面深度最大为5，一个页面中的 navigateTo不要写超过4个

路由与页面跳转 ▼

uni.navigateTo

uni.redirectTo

uni.reLaunch

uni.switchTab

uni.navigateBack

EventChannel

uni.preloadPage

窗口动画

相关API

- **uni.navigateTo(OBJECT):** 保留当前页面，跳转到应用内的某个页面，使用 uni.navigateBack 可以返回到原页面。
- 跳转到 tabBar 页面只能使用 switchTab 跳转
- 路由API的目标页面必须是在pages.json里注册的vue页面。
- **uni.redirectTo(OBJECT):** 关闭当前页面，跳转到应用内的某个页面。
- **uni.reLaunch(OBJECT):** 关闭所有页面，打开到应用内的某个页面。
- **uni.switchTab(OBJECT):** 跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面

- uni.navigateBack(OBJECT): 关闭当前页面，返回上一页面或多级页面。可通过 getCurrentPages() 获取当前的页面栈，决定需要返回几层。
- **uni.preloadPage(OBJECT)**: 预加载页面，是一种性能优化技术。被预载的页面，在打开时速度更快。
- getCurrentPages() 函数用于获取当前页面栈的实例，以数组形式按栈的顺序给出
- tips
 - ◆ navigateTo, redirectTo 只能打开非 tabBar 页面。
 - ◆ switchTab 只能打开 tabBar 页面。
 - ◆ reLaunch 可以打开任意页面。
 - ◆ 不能在 App.vue 里面进行页面跳转。

■ 示例：在index页面实现:组件方式页面跳转

◆ 跳转到其他tabBar页面

◆ 跳转到非tabBar页面

```
<text>首页</text>
```

```
<view>
```

```
  <!-- 路由跳转练习, 注意 open-type的选择, 另外, 不要使用系统提示的相对路径!
```

```
  <navigator url="/pages/finder/finder" open-type="switchTab">
```

```
    <button type="default">跳转finder页面</button>
```

```
  </navigator>
```

```
  <navigator url="/subpages/chat/chat" open-type="redirect">
```

```
    <button type="default">跳转chat页面 (静态) </button>
```

```
  </navigator>
```


课堂练习

- 完成对HelloUniapp项目中about页面<script>和methods的深度解读
- 完成tabBar页面、实现在tabBar页面和非tabBar页面的跳转，保存项目，后续将在此基础上继续开发。
- 以上无需交作业