

02

OPEN ORIENTED

凹凸实验室

科普小程序近期发布的三个新特性

- HM

三个新特性及基础库要求

- 自定义组件，1.6.3
- web-view，1.6.4
- 分包加载，1.6.4

自定义组件

降低模块间的耦合、提高复用性、有助于代码的维护

自定义组件由 JSON、WXML、WXSS、JS 四个文件组成

创建自定义组件

```
{  
  "component": true  
}
```

```
<view class="inner">  
  {{ innerText }}  
</view>  
  
<slot></slot>
```

```
.inner {  
  color: red;  
}
```

```
Component({  
  properties: {  
    innerText: {  
      type: String,  
      value: 'default value',  
    },  
  },  
  data: {  
    someData: {}  
  },  
  methods: {  
    customMethod(){}  
  }  
})
```

使用自定义组件

1、在页面的 JSON 文件中进行引用声明

```
{  
  "usingComponents": {  
    "component-tag-name": "path/to/component"  
  }  
}
```

2、在 WXML 文件进行调用

```
<view>  
  <component-tag-name inner-text="Hello world">  
    <view>Some other contents.</view>  
  </component-tag-name>  
</view>
```

created - 在组件实例进入页面节点树时执行

attached - 在组件实例进入页面节点树时执行

ready - 在组件布局完成后执行

moved - 在组件实例被移动到节点树另一个位置时执行

detached - 在组件实例被从页面节点树移除时执行

通信方式

1、页面 -> 组件，通过传递数据的方式

```
<test-comp refresh-time="{{ refreshTime }}">
</test-comp>
```

```
Component({
  properties: {
    refreshTime: {
      type: Number,
      observer: function() {
        this.refresh()
      }
    }
  },
  methods: {
    refresh() {},
  }
})
```

通信方式

2、组件 -> 页面，通过事件系统

```
<test-comp bindtestevent="onTestEvent"></test-comp>
```

```
Page({  
  onTestEvent: function(e){  
    console.log(e.detail)  
  }  
})
```

```
Component({  
  attached(){  
    this.triggerEvent('testEvent', 'test message')  
  }  
})
```

```
<view class="wrap">

  <search-bar refresh-time="{{ refreshTime }}" scroll-top="{{ scrollTop }}" bind:showToast="onShowToast"></search-bar>
  <carousel refresh-time="{{ refreshTime }}" bind:componentLoad="onComponentLoad"></carousel>
  <c-channel refresh-time="{{ refreshTime }}" bind:componentLoad="onComponentLoad"></c-channel>
  <task refresh-time="{{ refreshTime }}" bind:componentLoad="onComponentLoad" show-time="{{ showTime }}"></task>
  <grid-selected refresh-time="{{ refreshTime }}" bind:componentLoad="onComponentLoad"></grid-selected>

  <block wx:if="{{ loadOthers }}">
    <tab-selected refresh-time="{{ refreshTime }}" config="{{ itemConfig }}" scroll-top="{{ scrollTop }}"
      reach-bottom="{{ reachBottom }}" bind:idleTimeArrived="onIdle"></tab-selected>
  </block>

  <bottom-tips show-time="{{ showTime }}"></bottom-tips>

  <view wx:if="{{ scrollTop > 1000 }}" class="back2top" bindtap="back2top"></view>

</view>
```

- 组件内 WXSS 的样式有很多限制，建议使用 BEM 方式命名
- SLOT 可以有多个
- 组件间的代码复用通过 behaviors 特性来实现

web-view

一个类似全屏的 iframe，用来装载网页的容器

```
<web-view src="https://wqs.jd.com/test.shtml"></web-view>
```

使用 JSSDK 1.3.0 提供的接口进行跳转：

- wx.miniProgram.navigateTo
- wx.miniProgram.navigateBack
- wx.miniProgram.switchTab
- wx.miniProgram.reLaunch
- wx.miniProgram.redirectTo

黑科技：H5 页面向小程序发送数据

```
<web-view src="{{ url }}" bindmessage="onMessage"></web-view>
```

```
Page({  
  onMessage(msg) {  
    const arrData = msg.detail.data || []  
    console.log(arrData)  
  }  
})
```

```
wx.miniProgram.postMessage({ data: { testMsg: 'Hello world' } })  
wx.miniProgram.postMessage({ data: { testMsg: 'Hello universe' } })
```


- 页面 URL 只支持 HTTPS 协议
- 域名需要在后台添加并配置校验文件
- 最多可添加 20 个域名

分包加载

将小程序划分成不同的子包，可以优化加载速度，以及页面解耦

```
├── app.js
├── app.json
├── app.wxss
├── packageA
│   └── pages
│       ├── cat
│       └── dog
├── packageB
│   └── pages
│       ├── apple
│       └── banana
├── pages
│   ├── index
│   └── logs
└── utils
```

```
{
  "pages": [
    "pages/index",
    "pages/logs"
  ],
  "subPackages": [
    {
      "root": "packageA",
      "pages": [
        "pages/cat",
        "pages/dog"
      ]
    }, {
      "root": "packageB",
      "pages": [
        "pages/apple",
        "pages/banana"
      ]
    }
  ]
}
```

- 主包必须包含有默认启动页面，以及 TabBar 页面
- 整个小程序所有分包大小不超过 4M
- 单个分包/主包大小不能超过 2M
- 对于不支持分包的客户端版本，会进行整包加载

- <https://mp.weixin.qq.com/debug/wxadoc/dev/framework/custom-component/>
- <https://mp.weixin.qq.com/debug/wxadoc/dev/component/web-view.html>
- <http://git.jd.com/liuhuimin7/c-doc/blob/master/sub-packages.md>

T H A N K S
FOR YOUR WATCHING



OPEN ORIENTED

凹凸实验室