小程序的架构和配置

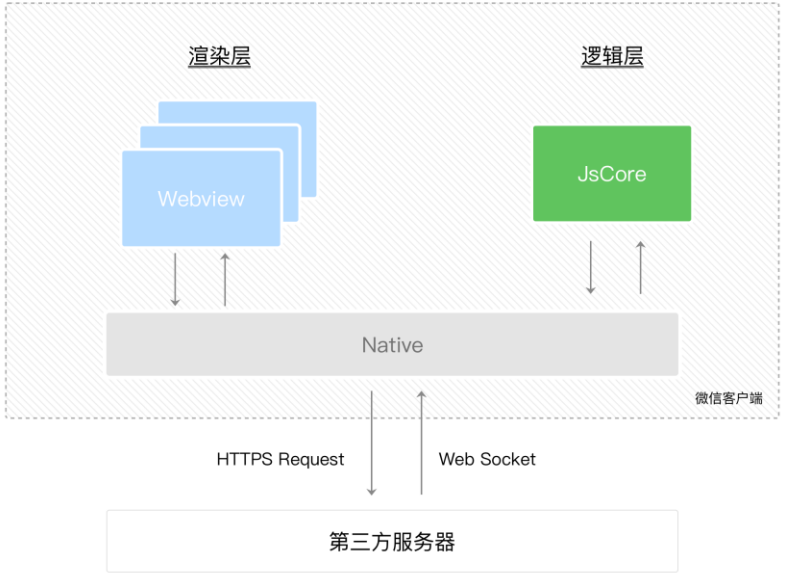
王红元 coderwhy

目录
content

小程序的架构模型

- **谁是小程序的宿主环境呢？微信客户端**
  - 宿主环境为了执行小程序的各种文件：wxml文件、wxss文件、js文件
- **当小程序基于 WebView 环境下时，WebView 的 JS 逻辑、DOM 树创建、CSS 解析、样式计算、Layout、Paint (Composite) 都发生在同一线程，在 WebView 上执行过多的 JS 逻辑可能阻塞渲染，导致界面卡顿。**
- **以此为前提，小程序同时考虑了性能与安全，采用了目前称为「双线程模型」的架构。**

- **双线程模型：**
  - WXML模块和WXSS样式运行于 渲染层，渲染层使用WebView线程渲染（一个程序有多个页面，会使用多个WebView的线程）。
  - JS脚本（app.js/home.js等）运行于 逻辑层，逻辑层使用JsCore运行JS脚本。
  - 这两个线程都会经由微信客户端（Native）进行中转交互。



小程序的配置文件

- **小程序的很多开发需求被规定在了配置文件中。**
- **为什么这样做呢?**
  - 这样做可以更有利于我们的开发效率;
  - 并且可以保证开发出来的小程序的某些风格是比较一致的;
  - 比如导航栏 – 顶部TabBar，以及页面路由等等。

- **常见的配置文件有哪些呢?**
  - project.config.json：项目配置文件, 比如项目名称、appid等;
    - https://developers.weixin.qq.com/miniprogram/dev/devtools/projectconfig.html
  - sitemap.json：小程序搜索相关的;
    - https://developers.weixin.qq.com/miniprogram/dev/framework/sitemap.html
  - app.json：全局配置;
  - page.json：页面配置;

全局app配置文件

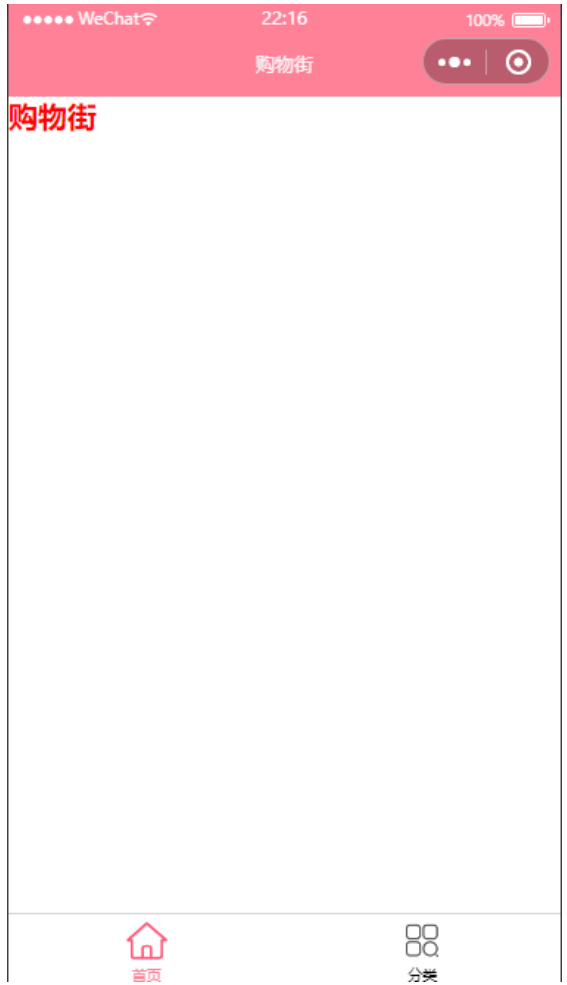
- **全局配置比较多，我们这里将几个比较重要的. 完整的查看官方文档.**
  - https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html

属性	类型	必填	描述
pages	String[]	是	页面路径列表
window	Object	否	全局的默认窗口表现
tabBar	Object	否	底部 tab 栏的表现

- **pages: 页面路径列表**
  - 用于指定小程序由哪些页面组成，每一项都对应一个页面的 路径（含文件名） 信息。
  - 小程序中所有的页面都是必须在pages中进行注册的。
- **window: 全局的默认窗口展示**
  - 用户指定窗口如何展示, 其中还包含了很多其他的属性
- **tabBar: 顶部tab栏的展示**
  - 具体属性稍后我们进行演示

配置案例实现

- 我们来做如下的效果:




```
"tabBar": {
  "selectedColor": "#ff8198",
  "list": [
    {
      "pagePath": "pages/home/home",
      "text": "首页",
      "iconPath": "assets/tabbar/home.png",
      "selectedIconPath": "assets/tabbar/home_active.png"
    },
    {
      "pagePath": "pages/category/category",
      "text": "分类",
      "iconPath": "assets/tabbar/category.png",
      "selectedIconPath": "assets/tabbar/category_active.png"
    }
  ]
},
```

页面page配置文件

- **每一个小程序页面也可以使用 .json 文件来对本页面的窗口表现进行配置。**
  - 页面中配置项在当前页面会覆盖 app.json 的 window 中相同的配置项。
  - https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/page.html

配置项

属性	类型	默认值	描述	最低版本
navigationBarBackgroundColor	HexColor	#000000	导航栏背景颜色，如 `#000000`	
navigationBarTextStyle	string	white	导航栏标题颜色，仅支持 `black` / `white`	
navigationBarTitleText	string		导航栏标题文字内容	
navigationStyle	string	default	导航栏样式，仅支持以下值： `default` 默认样式 `custom` 自定义导航栏，只保留右上角胶囊按钮。	iOS/Android 微信客户端 7.0.0，Windows 微信客户端不支持

```
{
  "usingComponents": {},
  "navigationBarTitleText": "哈哈哈",
  "enablePullDownRefresh": true,
  "backgroundTextStyle": "dark"
}
```

注册小程序 – App函数

- **每个小程序都需要在 app.js 中调用 App 函数 注册小程序示例**
  - 在注册时, 可以绑定对应的生命周期函数;
  - 在生命周期函数中, 执行对应的代码;
  - https://developers.weixin.qq.com/miniprogram/dev/reference/api/App.html

- **我们来思考：注册App时，我们一般会做什么呢?**
  - 1. 判断小程序的进入场景
  - 2. 监听生命周期函数，在生命周期中执行对应的业务逻辑，比如在某个生命周期函数中进行登录操作或者请求网络数据;
  - 3. 因为App()实例只有一个，并且是全局共享的（单例对象），所以我们可以将一些共享数据放在这里;

App函数的参数

Object object

属性	类型	默认值	必填	说明	最低版本
onLaunch	function		否	生命周期回调——监听小程序初始化。	
onShow	function		否	生命周期回调——监听小程序启动或切前台。	
onHide	function		否	生命周期回调——监听小程序切后台。	
onError	function		否	错误监听函数。	
onPageNotFound	function		否	页面不存在监听函数。	1.9.90
onUnhandledRejection	function		否	未处理的 Promise 拒绝事件监听函数。	2.10.0
onThemeChange	function		否	监听系统主题变化	2.11.0
其他	any		否	开发者可以添加任意的函数或数据变量到 `Object` 参数中，用 `this` 可以访问	

作用一：判断打开场景

- **小程序的打开场景较多：**
  - 常见的打开场景：群聊会话中打开、小程序列表中打开、微信扫一扫打开、另一个小程序打开
  - https://developers.weixin.qq.com/miniprogram/dev/reference/scene-list.html
- **如何确定场景?**
  - 在onLaunch和onShow生命周期回调函数中，会有options参数，其中有scene值；

场景值ID	说明	图例
1000	其他	/
1001	发现栏小程序主入口，「最近使用」列表（基础库2.2.4版本起包含「我的小程序」列表）	/
1005	微信首页顶部搜索框的搜索结果页	查看
1006	发现栏小程序主入口搜索框的搜索结果页	查看
1007	单人聊天会话中的小程序消息卡片	查看
1008	群聊会话中的小程序消息卡片	查看

作用二：定义全局App的数据

- 作用二：可以在Object中定义全局App的数据

```
globalData: {
  token: "",
  userInfo: {}
}
```

- 定义的数据可以在其他任何页面中访问：

```
onLoad() {
  const app = getApp()
  console.log(app.globalData.token);
  console.log(app.globalData.user);
},
```

作用三 – 生命周期函数

- **作用二：在生命周期函数中，完成应用程序启动后的初始化操作**
  - 比如登录操作（这个后续会详细讲解）；
  - 比如读取本地数据（类似于token，然后保存在全局方便使用）
  - 比如请求整个应用程序需要的数据；

```
onLaunch() {
  // 1.读取本地数据
  const token = wx.getStorageSync('token')
  const user = wx.getStorageSync('user')
  this.globalData.token = token
  this.globalData.user = user

  // 2.登录逻辑
  wx.login({ success: res => {} })

  // 3.请求数据
  wx.request({ url: 'url' })
},
```

注册页面 – Page函数

- **小程序中的每个页面，都有一个对应的js文件，其中调用Page函数注册页面示例**
  - 在注册时，可以绑定初始化数据、生命周期回调、事件处理函数等。
  - https://developers.weixin.qq.com/miniprogram/dev/reference/api/Page.html

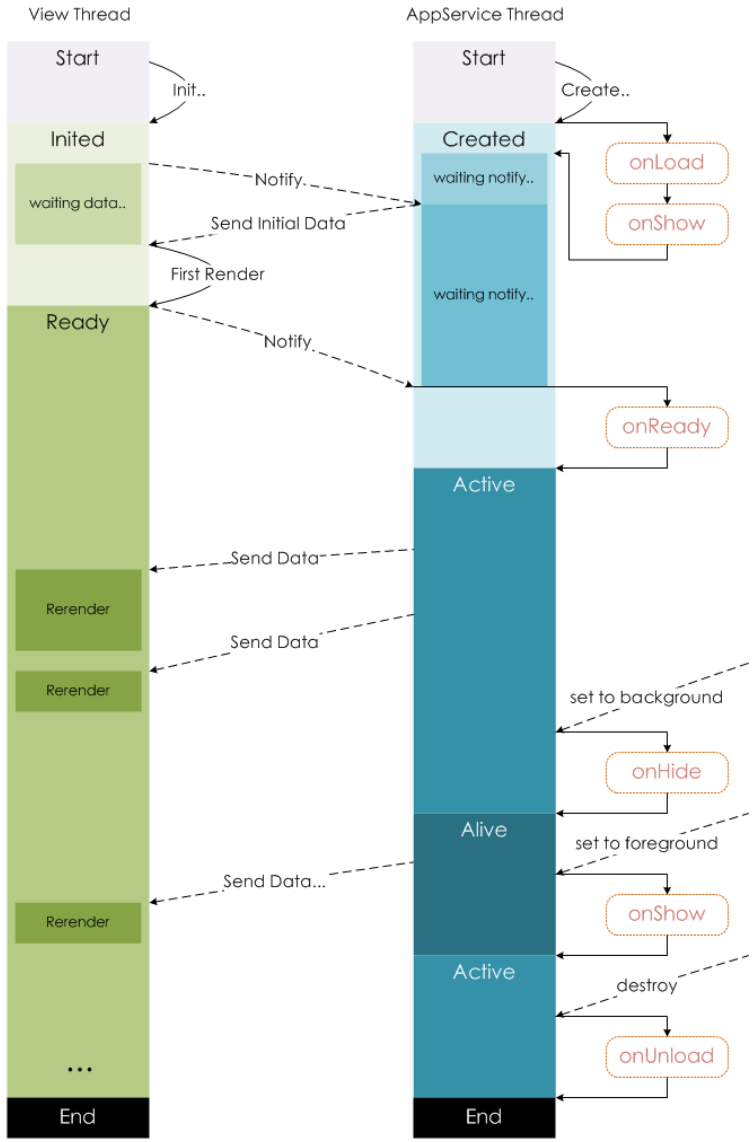
- **我们来思考：注册一个Page页面时，我们一般需要做什么呢?**
  - 1.在生命周期函数中发送网络请求，从服务器获取数据；
  - 2.初始化一些数据，以方便被wxml引用展示；
  - 3.监听wxml中的事件，绑定对应的事件函数；
  - 4.其他一些监听（比如页面滚动、上拉刷新、下拉加载更多等）；

注册Page时做什么呢?

Object object

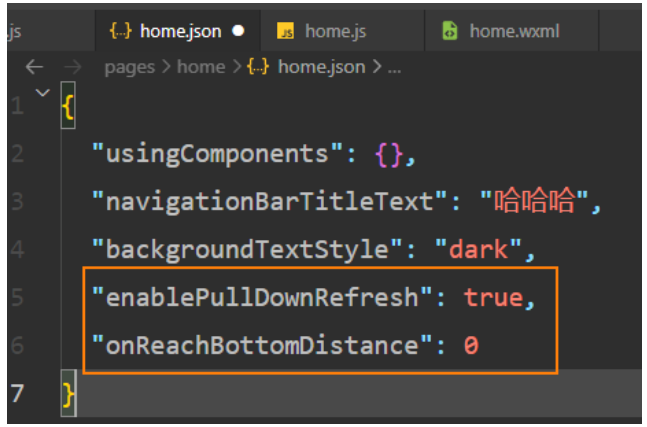
属性	类型	默认值	必填	说明
data	Object			页面的初始数据
options	Object			页面的组件选项，同 `Component` 构造器 中的 `options` ，需要基础库版本 2.10.1
behaviors	String Array			类似于 mixins 和traits的组件间代码复用机制，参见 behaviors，需要基础库版本 2.9.2
onLoad	function			生命周期回调—监听页面加载
onShow	function			生命周期回调—监听页面显示
onReady	function			生命周期回调—监听页面初次渲染完成
onHide	function			生命周期回调—监听页面隐藏
onUnload	function			生命周期回调—监听页面卸载

Page页面的生命周期



上拉和下拉的监听

- 监听页面的下拉刷新和上拉加载更多：
  - 步骤一：配置页面的json文件；
  - 步骤二：代码中进行监听；




```
onPullDownRefresh() {
  console.log("监听到下拉刷新");
  setTimeout(() => {
    console.log('----');
    wx.stopPullDownRefresh({
      success: (res) => {
        console.log(res);
      },
      fail: (err) => {
        console.log(err);
      }
    })
  }, 1000);
},
```

```
onReachBottom() {
  console.log("onReachBottom");
  this.setData({
    listCount: this.data.listCount + 30
  })
},
```