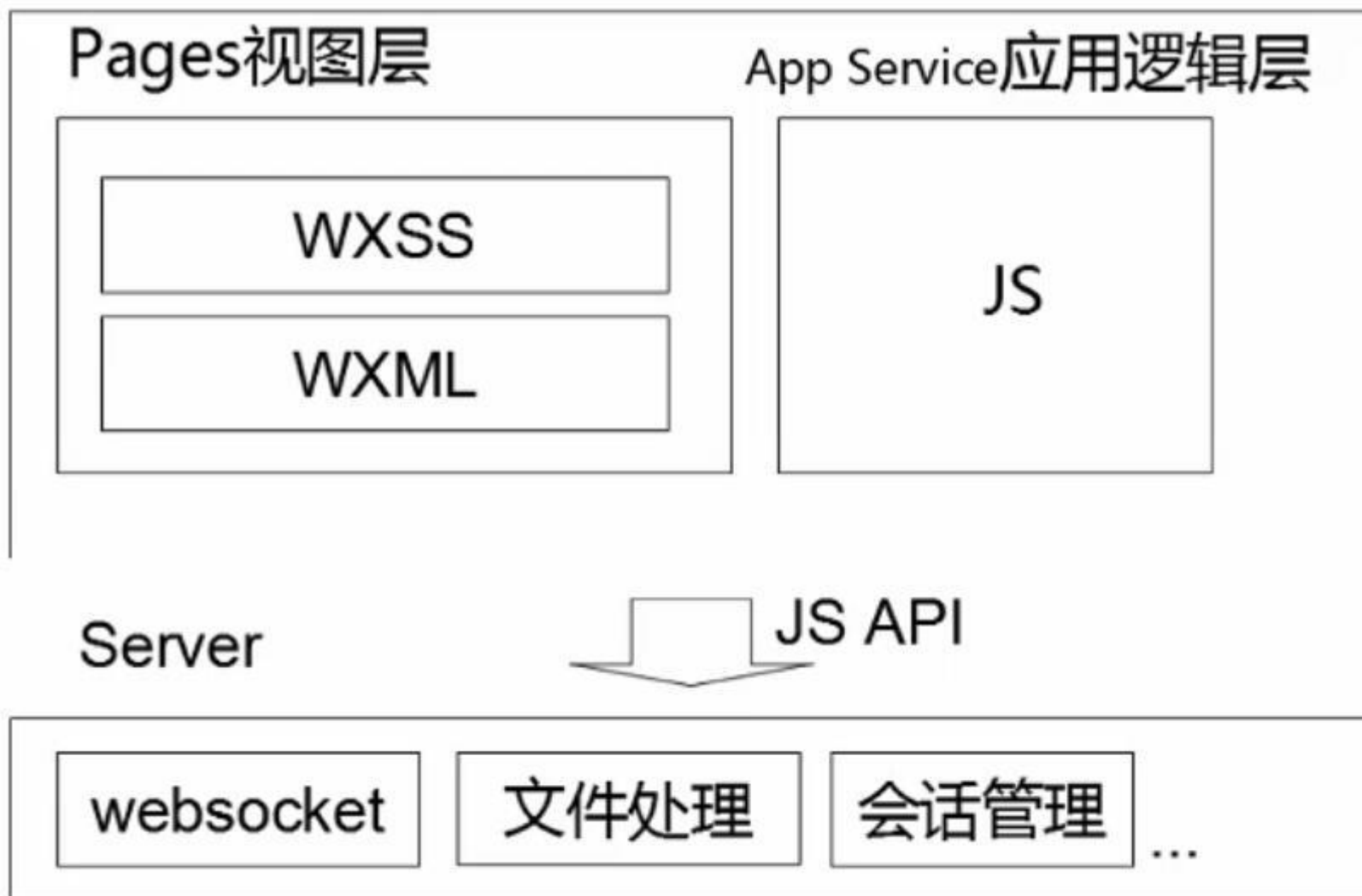


金于虎：微信小程序认知、安装与 **demo** 学习摘要

一：认知

1. 技术架构



2.前端的视图

WXSS

样式定义类似 css 的样式表，定义页面元素的样式

wxml

标签语言对 html 的扩展，定义我这里面会有那些组件，会有那些元素里面。更多的是做渲染

js 触发事件

定义触发后又后端的那些程序来处理

3.后端的 server

很多接口以 jsapi 的方式提供

官网地址

[地址 1 工具](#)

[例子](#)

二：组件和数据绑定

1. 组件的基本使用
2. 数据绑定
3. 条件渲染

4. 列表渲染 <http://www.runoob.com/cssref/pr-border-collapse.html>

三：下载安装

1. 下载和安装开发程序

下载地址：[开发者文档](#)

1. 扫描登陆
2. 添加项目（条件：微信号是需要是微信开发者）
3. 无 Appid（条件：需要 Nodejs 支持）

注意点：

1. 在 SDK 中有些 api 是依赖 nodejs 的
2. 部分功能受限 指的是无法再手机上看或者预览

2. 熟悉开发工具的使用

3. 熟悉目录架构

四：demo 学习

1. 项目核心文件

app.js

app.js 是小程序的脚本代码。我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量。调用框架提供的丰富的 API，

app.json

app.json 是对整个小程序的全局配置。我们可以在这个文件中配置小程序是由哪些页面组成，配置小程序的窗口 背景色，配置导航条样式，配置默认标题。注意该文件不可添加任何注释。

app.wxss

app.wxss 是整个小程序的公共样式表。我们可以在页面组件的 class 属性上直接使用 app.wxss 中声明的样式规则。

1. 事件
2. 全局函数
3. 全局属性

2. 页面核心文件

页面目录和配置以及小程序首页

微信小程序中的每一个页面的【路径+页面名】都需要写在 app.json 的 pages 中，且 pages 中的第一个页面是小程序的首页。

页面组成

每一个小程序页面是由同路径下同名的四个不同后缀文件的组成，如：

1. index.js
2. index.wxml
3. index.wxss

4. index.json

.js 后缀的文件是脚本文件，.json 后缀的文件是配置文件，.wxss 后缀的是样式表文件，.xml 后缀的文件是页面结构文件。 注意：目录名称和文件名称最好一样

index.js

index.js 是页面的脚本文件，在这个文件中我们可以监听并处理页面的生命周期函数、获取小程序实例，声明并处理数据，响应页面交互事件等。

index.wxss

index.wxss 是页面的样式表，

样式的覆盖，继承和重写

页面的样式表是非必要的。当有页面样式表时，页面的样式表中的样式规则会层叠覆盖 app.wxss 中的样式规则。如果不指定页面的样式表，也可以在页面的结构文件中直接使用 app.wxss 中指定的样式规则。

index.json

index.json 是页面的配置文件：

配置的覆盖，继承和重写

页面的配置文件是非必要的。当有页面的配置文件时，配置项在该页面会覆盖 app.json 的 window 中相同的配置项。如果没有指定的页面配置文件，则在该页面直接使用 app.json 中的默认配置。

demo 片段

```
onLaunch: function () {  
    //调用 API 从本地缓存中获取数据  
  
    var logs = wx.getStorageSync('logs') || []  
  
    logs.unshift(Date.now())  
  
    wx.setStorageSync('logs', logs)  
  
},  
getUserInfo:function(cb){  
    var that = this  
  
    if(this.globalData.userInfo){  
        typeof cb == "function" && cb(this.globalData.userInfo)  
    }else{  
        //调用登录接口  
  
        wx.login({  
            success: function () {  
                wx.getUserInfo({  
                    success: function (res) {
```

```
        that.globalData.userInfo = res.userInfo

        typeof cb == "function" && cb(that.globalData.userInfo)
    }

    })

}

})

}

},

globalData:{

    userInfo:null

}

})
```

3.微信小程序的一些思想

1. 微信小程序是以数据进行驱动的 (data 属性)
2. 单向数据绑定。 (数据渲染 , 模板引擎)
3. MVC 思想 (DVC 思想) data view controller

4. wxml 可以想成 html 但他的本质是 xml(自定义的标签)

4. 变量输出{{}}

```
<image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size="cover"></image>

<text class="userinfo-nickname">{{userInfo.nickName}}</text>

</view>

<view class="usermotto">

  <text class="user-motto">{{motto}}</text>
```

5.事件处理

1. 绑定事件 bindtap= "事件"

dmeo 轻巧事件

```
<view bindtap="bindViewTap" class="userinfo">
```

1. 事件处理函数

2. //事件处理函数

3. bindViewTap: function() {

4. wx.navigateTo({

5. url: '../logs/logs'

```
6.  })
```

```
7.  },
```

6. 页面跳转

```
wx.navigateTo({  
  url: '../logs/logs'  
})
```

7. 作用域和优先级

页面作用域(page.js)和全局作用域(app.js)

app

```
"window":{  
  "backgroundTextStyle":"light",  
  
  "navigationBarBackgroundColor": "#fff",  
  "navigationBarTitleText": "WeChat",  
  "navigationBarTextStyle":"black"  
}
```

页面

```
{  "navigationBarTitleText": "WeChat"}
```