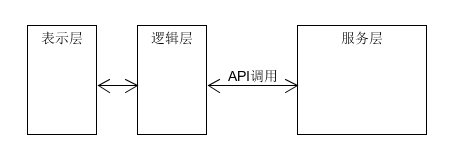
# 技术选型及理由

微信小程序采用官方指定的框架进行开发。因此小程序的开发有指定的语言和框架，因此在技术选型上并没有其它选择。

# 架构和框架

架构



微信小程序框架提供了自己的视图层描述语言 WXML 和 WXSS，以及基于 JavaScript 的逻辑层框架，并在视图层与逻辑层间提供了数据传输和事件系统，可以让开发者可以方便的聚焦于数据与逻辑上。

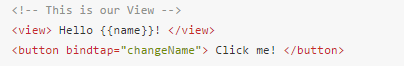
### 响应的数据绑定

框架的核心是一个响应的数据绑定系统。

整个系统分为两块视图层（View）和逻辑层（App Service）

框架可以让数据与视图非常简单地保持同步。当做数据修改的时候，只需要在逻辑层修改数据，视图层就会做相应的更新。

通过这个简单的例子来看：





开发者通过框架将逻辑层数据中的 name 与视图层的 name 进行了绑定，所以在页面一打开的时候会显示 Hello WeChat!

当点击按钮的时候，视图层会发送 changeName 的事件给逻辑层，逻辑层找到对应的事件处理函数

逻辑层执行了 setData 的操作，将 name 从 WeChat 变为 MINA，因为该数据和视图层已经绑定了，从而视图层会自动改变为 Hello MINA! 。

### 页面管理

框架 管理了整个小程序的页面路由，可以做到页面间的无缝切换，并给以页面完整的生命周期。开发者需要做的只是将页面的数据，方法，生命周期函数注册进 框架 中，其他的一切复杂的操作都交由 框架 处理。

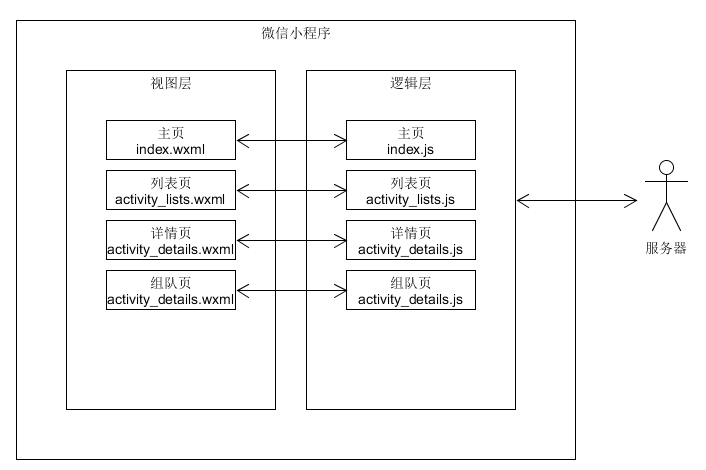
### 基础组件

框架 提供了一套基础的组件，这些组件自带微信风格的样式以及特殊的逻辑，开发者可以通过组合基础组件，创建出强大的微信小程序 。

### 丰富的 API

框架 提供丰富的微信原生 API，可以方便的调起微信提供的能力，如获取用户信息，本地存储，支付功能等。

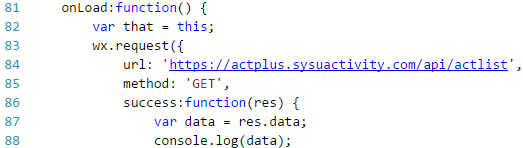
# 三、模块划分



# 使用的软件设计技术

### SOA

通过API获取后端服务数据示例，如index.js中获取活动列表



获取到的数据如下：



### 面向对象

在小程序中，整个页面被视为一个对象，该页面对象拥有data属性（页面数据）和各种相应事件的方法。

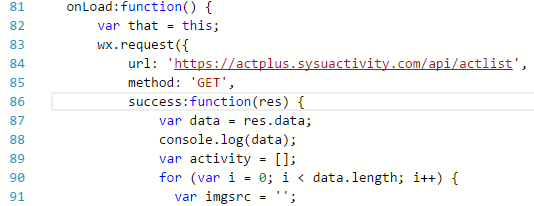
以index.js为例，



主页为Page对象，其中data属性是页面的数据，页面效果如下



方法示例：



该方法在页面创建时被调用，向服务器请求活动列表数据。