

# 微信门户开发框架 使用指导说明书

V2.0

序号	修改人	修改日期	修改后版本	修改说明
1	伍华聪	2014-12-26	V1.0	正式版
2	伍华聪	2015-4-14	V1.1	完善修订版
3	伍华聪	2020-3-5	V2.0	全部修改完善

## 目 录

<b>1.</b>	<b>引言.....</b>	<b>3</b>
1.1.	背景.....	3
1.2.	参考资料.....	3
1.3.	术语与缩写.....	3
1.4.	系统的重要特性总结.....	3
<b>2.</b>	<b>系统使用指导 .....</b>	<b>5</b>
2.1.	环境准备.....	5
2.2.	模块说明.....	5
2.3.	模块关系.....	9
2.3.1.	开发模块划分.....	9
2.3.2.	公众号、企业微信、小程序公共部分.....	9
2.3.3.	数据管理模块和 API 模块调用关系.....	9
2.3.4.	微信界面项目.....	10
2.3.5.	基础类库支持部分.....	10
2.3.6.	公众号和企业微信实现接口.....	11
2.4.	账号配置.....	13
2.4.1.	微信公众号配置.....	13
2.4.2.	企业微信配置.....	16
2.4.3.	测试文件配置.....	20
2.5.	开发过程处理.....	21
2.5.1.	微信开发者工具.....	22
2.5.2.	微信 H5 页面及 JSDK 开发.....	23
2.5.3.	微信框架中 H5 控制器关系.....	25
2.5.4.	账号微信菜单管理.....	27
2.5.5.	微信公众号开发.....	31

2.5.6.	企业微信开发.....	44
2.6.	数据库配置.....	49
2.6.1.	SqlServer 数据库的配置.....	49
2.6.2.	MySql 数据库的配置.....	50
2.6.3.	SQLite 数据库的配置.....	50
2.7.	企业微信 WINFORM 客户端.....	50
<b>3.</b>	<b>JQUERY WEUI 控件的使用 .....</b>	<b>51</b>
3.1.	界面控件使用.....	51

## 1. 引言

### 1.1. 背景

微信应用如火如荼，很多公司都希望搭上信息快车，这个是一个非常重要的商机。用微信进行营销，一方面可以提高企业产品的宣传效果，另一方面，也能提高我们企业的竞争力，缩短和直接客户之间的距离。

该微信门户应用开发框架，就是为了协助企业快速搭建微信应用以及管理的平台，使我们快速融入微信的市场化浪潮中，在市场激烈的竞争中占据有利条件，获得更多的市场回报。

本文档主要介绍《微信门户开发框架》的开发使用过程，通过对项目的说明，以及如何配置账号等信息，对框架的入门使用进行指导。

### 1.2. 参考资料

序号	名称	版本/日期	来源
1	《Web 开发框架-架构设计说明书.doc》		内部
	《微信门户开发框架功能介绍白皮书.doc》		内部

### 1.3. 术语与缩写

- 1 在本文件中出现的“微信门户开发框架”或“框架”一词，除非特别说明，均指本文介绍的微信门户开发框架。
- 2 在本文安装.NET框架中，除非特别说明，均指.NET 4.5框架。
- 3 在本文所指的VS，是指Visual Studio 2017/2019开发工具。

### 1.4. 系统的重要特性总结

微信门户应用管理系统，采用基于 MVC + Bootstrap + Ajax + Enterprise Library

的技术路线，界面层采用 Bootstrap + Metronic 组合的前端框架，数据访问层支持 Oracle、SQLServer、MySQL、PostgreSQL 等数据库。框架以 MVC5, C#4.5 作为开发基础，能够部署在任何 .NET 服务器上，建议部署到阿里云服务器。

框架包含了公众号、企业微信和小程序的 API 封装和后台管理操作，并且公众号或者企业微信后台都支持多个账号的使用，也就是一个后台管理系统可以管理多个公众号、企业微信、小程序的消息处理。

在微信门户系统里面，《**公众号**》实现下面这些功能操作：

- 1) 实现菜单的动态配置及更新到服务器上；
- 2) 动态定义事件和响应消息，实现对不同行业，不同需求的菜单动作响应；
- 3) 动态的应答指令配置处理，实现整套应答链的消息处理；
- 4) 获取订阅用户和用户分组信息，并可以实现用户分组信息的维护等操作；
- 5) 管理并更新多媒体文件、图文消息等内容，方便为客户推送消息做准备。
- 6) 使用向选定订阅用户或者分组进行消息的群发功能。
- 7) 提供对用户语音的识别，并将识别结果和事件处理进行对接，实现语音内容的处理。
- 8) 提供对地理位置的记录和扩展应用，实现对百度天气、热映影片、影院、旅游、交通消息等方面的第三方应用的对接。
- 9) 记录用户输入的会话记录，并可以根据会话记录向客户进行消息的发送。
- 10) 提供用户二维码扫描事件的处理。
- 11) 使用重定向功能获取订阅用户信息，并可以通过短信集成客户的手机验证功能；
- 12) 集成微信扫一扫的功能，实现条码和二维码扫码的功能整合，可以在后台整合自身应处理；
- 13) 集成微信其他最新功能，包括拍照和图库选择、快速发送地理位置等功能；
- 14) 集成获取关注成员的唯一 ID，更新用户备注，获取客服聊天记录等功能。

企业版包括《**企业微信**》各种封装管理：

- 1) 实现企业通讯录的管理，包括部门管理、成员管理、标签管理，使得我们可以更方便同步企业微信信息；
- 2) 实现企业微信无限制的消息发送，可以使企业和关注成员之间沟通更顺畅；
- 3) 集成企业微信文本消息、图片消息、文件消息、语音消息、视频消息、图文消息等

内容的发送整合，统一处理接口；

4) 实现企业微信菜单的动态配置及更新到服务器上，动态定义菜单事件和响应消息，实现不同的菜单动作响应；

5) 管理并更新企业微信的多媒体文件，方便同步到自己文件服务器进行存储和处理；

6) 使用重定向功能获订阅用户信息，实现企业成员身份的认证和处理。

7) 平台和微信开放平台整合扫描二维码登录，通过扫码绑定并直接登陆系统。

企业版包括《小程序》各种封装管理：

集成微信小程序相关接口处理，提供多个微信小程序的案例和 Web API 接入代码。

另外，我们还为企业微信开发了一个基于 C/S 的 Winform 管理客户端，可以方便管理通讯录的组织机构、标签、人员信息，以及文本、图片、语音、视频等消息的发送功能。

## 2. 系统使用指导

### 2.1. 环境准备

本框架采用 C#开发，Web 管理端采用基于 MVC + Bootstrap + Ajax + Enterprise Library 的技术路线的架构，需要使用 VS2017/2019 进行开发，默认数据库采用 SQL Server，可以安装 SQL Server 2008R2 及以上。

微信公众号开发，一般需要进行相关的网页页面调试，请下载微信提供的《微信开发者工具》，下载地址为：<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>。

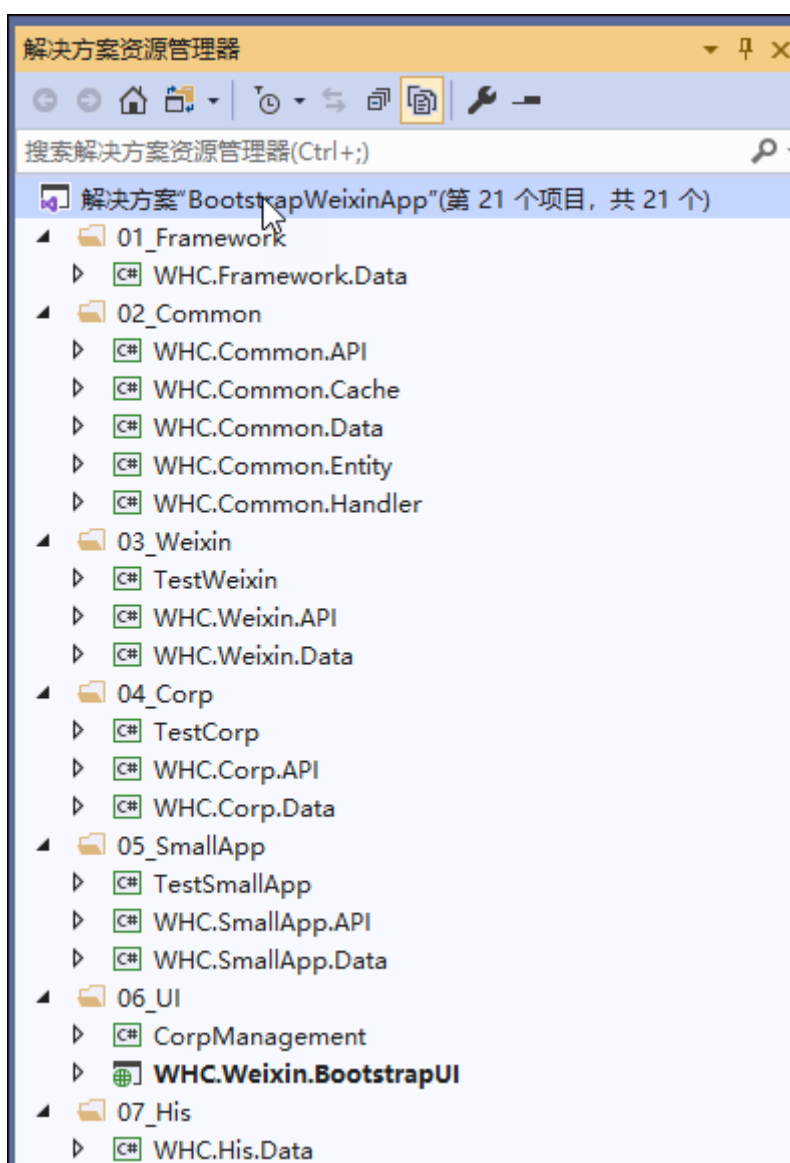
框架开发一般还需要使用到代码生成工具，工具可以从我公司的官网下载 (<http://www.iqidi.com/Download/Database2SharpSetup.rar>)。

### 2.2. 模块说明

整个框架项目包含了微信公众号、企业微信、小程序三个部分内容，其中三者之间有一些相同的部分抽象出来放在一个公共的项目里面，然后公众号、企业微信、小程序共享一个后台管理系统，这个后台系统主要就是实现账号信息的配置、和微信后台事件的对接、以及

权限和系统数据的管理维护等功能。

使用 VS 打开项目源码后，解决方案的工程如下所示。



下面是对上面项目工程进行说明，方便理解其中各个模块的意义。

编号	名称	说明
01_Framework 目录里面的项目，是框架的基础内容，包括权限、字典、附件模块的管理。		
1	Security	权限管理模块业务层。
2	Dictionary	字典管理模块业务层。
3	Attachment	附件管理模块业务层
4	Other	其他系统内容，如图标、产品、测试用户等基础信息
02_Common 目录里面的项目，是公众号/企业微信/小程序公用的部分，如账号配置、基础		

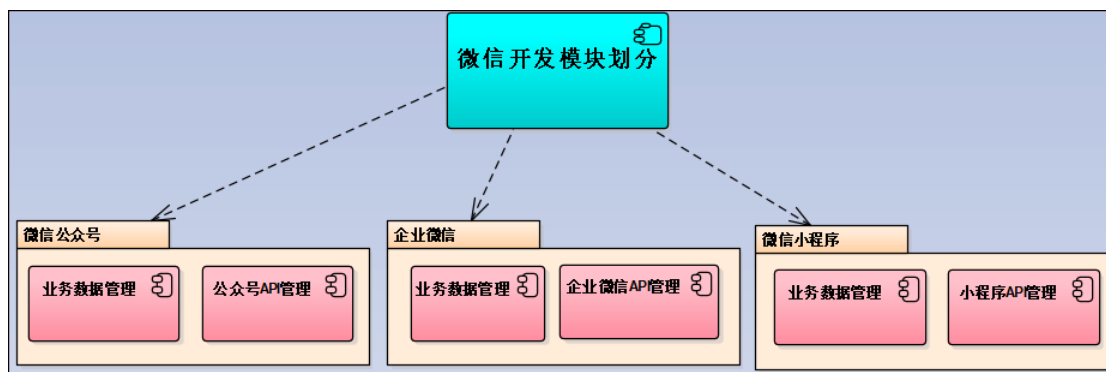


消息对象类，也可以是 Web 后台管理所需的模块内容。		
1	WHC.Common.API	提供一些第三方模块的封装，如邮件处理、短信处理以及一些第三方的应用扩展模块等。
2	WHC.Common.Cache	基于 CacheManager 的接口处理的缓存接口，可以实现 SystemRuntimeCache、Redis 等缓存的整合。
3	WHC.Common.Data	提供常用模块，需要对数据库处理的模块的封装，包括一些如账号、菜单、事件、消息内容等公用的数据存储。
4	WHC.Common.Entity	提供常用模块里面设计到的实体类信息。包括公众号和企业微信都共同用到的对象类。
5	WHC.Common.Handler	多个 IOC 方式的模块，可以在所需系统注册使用。 自定义接口包含有测试处理接口、企业微信消息推送接口、设备信息二维码扫码接口、短信和邮件发送接口。 外部的插件业务接口包括：百度的地理位置解析接口、电影院信息查询、天气信息查询、交通信息查询、旅游信息查询等。
03_Weixin 目录里面的项目，是微信公众号 API 相关的内容。		
1	TestWeixin	是对公众号的 API 封装的项目模块进行测试。
2	WHC.Weixin.API	针对微信公众号 API 的单独封装。
3	WHC.Weixin.Data	结合数据库存储内容，对微信公众号模块的管理。这个模块包含对 WHC.Weixin.API 的使用。
04_Corp 目录里面的项目，是对企业微信 API 的封装和对数据库管理模块的封装。		
1	TestCorp	是对企业微信的 API 封装的项目模块进行测试。
2	WHC.Corp.API	针对微信企业微信 API 的单独封装。
3	WHC.Corp.Data	结合数据库存储内容，对企业微信模块的管理。这个模块包含对 WHC.Corp.API 的使用。
05_SmallApp 目录里面的项目，是对小程序 API 的封装和对数据库管理模块的封装。		

1	TestSmallApp	是对小程序的 API 封装的项目模块进行测试。
2	WHC.SmallApp.API	针对小程序 API 的单独封装。
3	WHC.SmallApp.Data	结合数据库存储内容，对小程序模块的管理。这个模块包含对 WHC. SmallApp.API 的使用。
06_UI 是微信后台管理系统部分内容，包括后台管理系统及企业微信的管理客户端程序。		
1	WHC.Weixin.BootstrapUI	微信管理后台，包括对公众号和企业微信的统一管理，实现各个模块数据的同步和查看管理操作。
2	CorpManagement	基于 C/S 的 Winform 企业微信管理客户端，可以方便管理企业微信的通讯录的组织机构、标签、人员信息，以及文本、图片、语音、视频等消息的发送功能。
07_His 是扩展的项目部分		
1	WHC.His.Data	包含对医疗设备维修项目、处方审核项目等扩展项目的模块管理。
其他模块		
1	公用类库 WHC.Framework.Commons WHC.Framework.ControlUtil	提供日常各种开发操作的辅助类库。 WHC.Framework.Commons 项目是对常规.NET 共用类库的封装使用，已达到统一、高效、方便的目的。 WHC.Framework.ControlUtil 项目是对第三方组件的封装使用，以及对整个框架各层抽象类的定义和封装。
2	分页类 WHC.Pager.Entity	这个是独立定义的一个分页参数对象，方便框架里面 对数据进行分页处理。
4	参数配置管理 SettingsProvider.Net	提供了一个第三方的配置信息的管理类。这个是在 GitHub 上的一款参数配置组件，能够基于普通配置文件、ProgramData 目录文件、独立存储区文件等方式的 配置文件存储，它主要是基于 Json 格式进行的配置保存，因此我们还可以把它存储到数据库

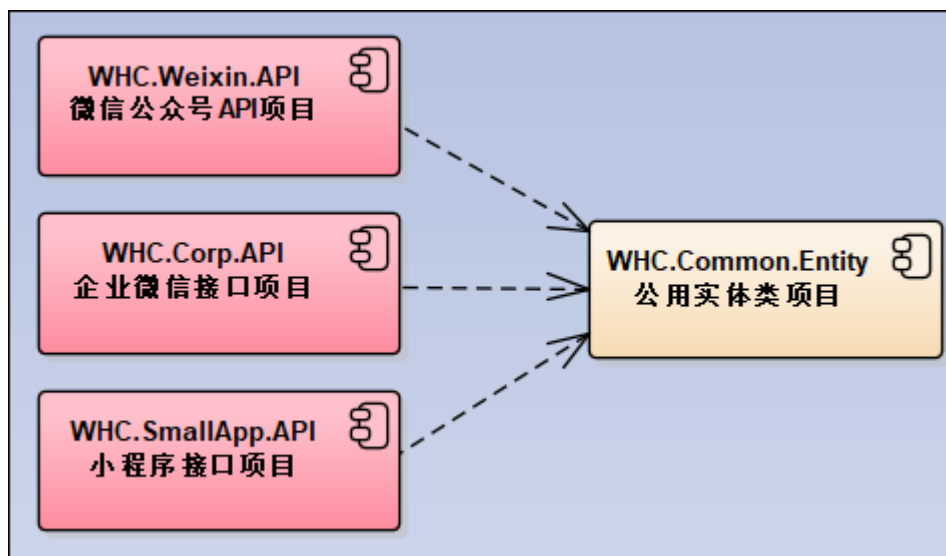
## 2.3. 模块关系

### 2.3.1. 开发模块划分



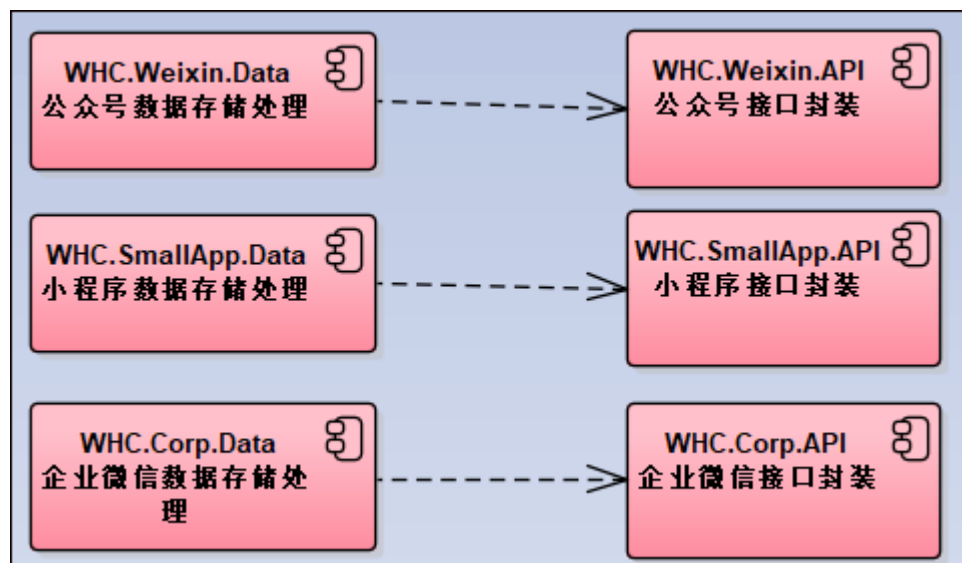
### 2.3.2. 公众号、企业微信、小程序公共部分

这几个部分又有一些基础一样的内容，抽取出来作为公共的项目。



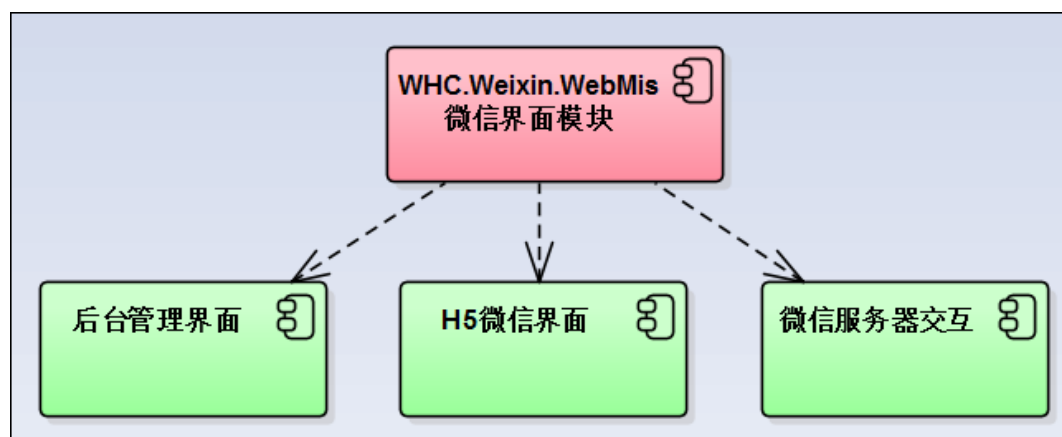
### 2.3.3. 数据管理模块和 API 模块调用关系

公众号、企业微信、小程序几个分类，它们包含了数据存储的模块内容，也包括了对应分类的微信 API 操作部分。



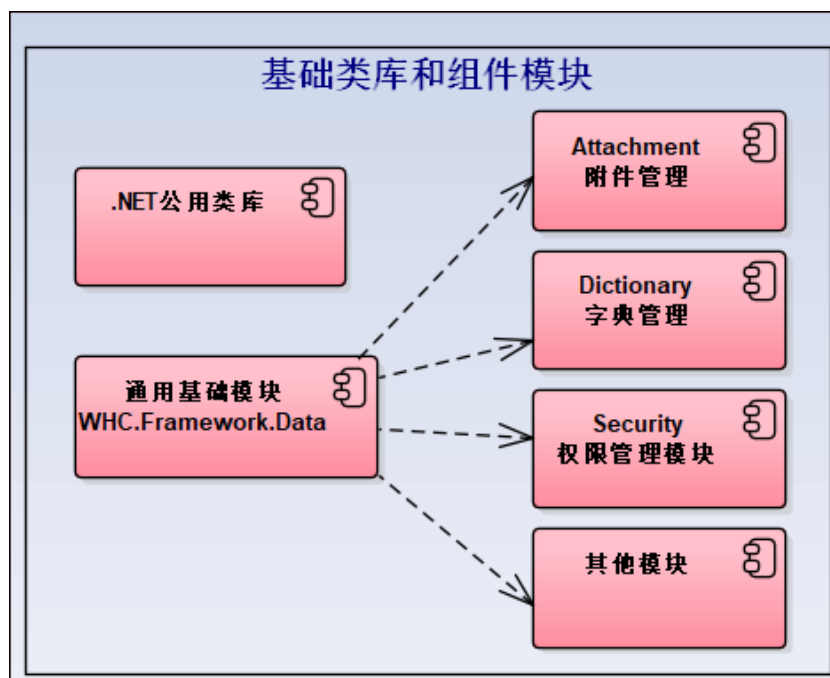
### 2.3.4. 微信界面项目

微信界面部分是前面模块组件的综合使用，在微信应用里面，一般需要使用 80 端口和微信服务器做交互，而这个同时往往也是我们项目的端口地址。



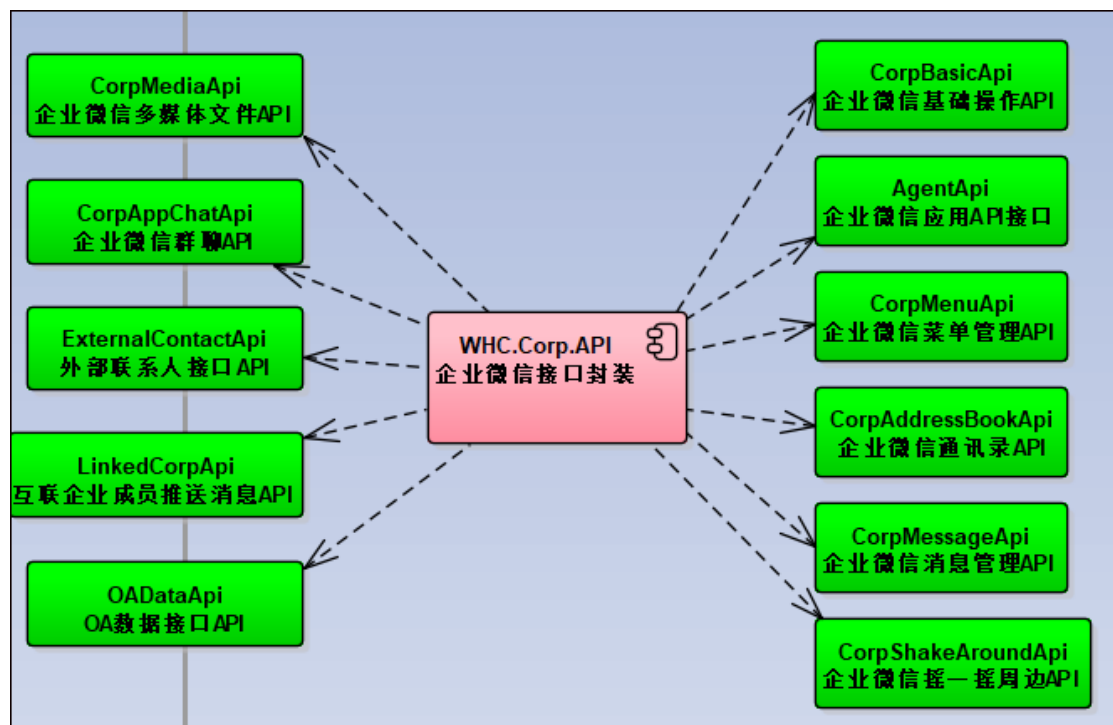
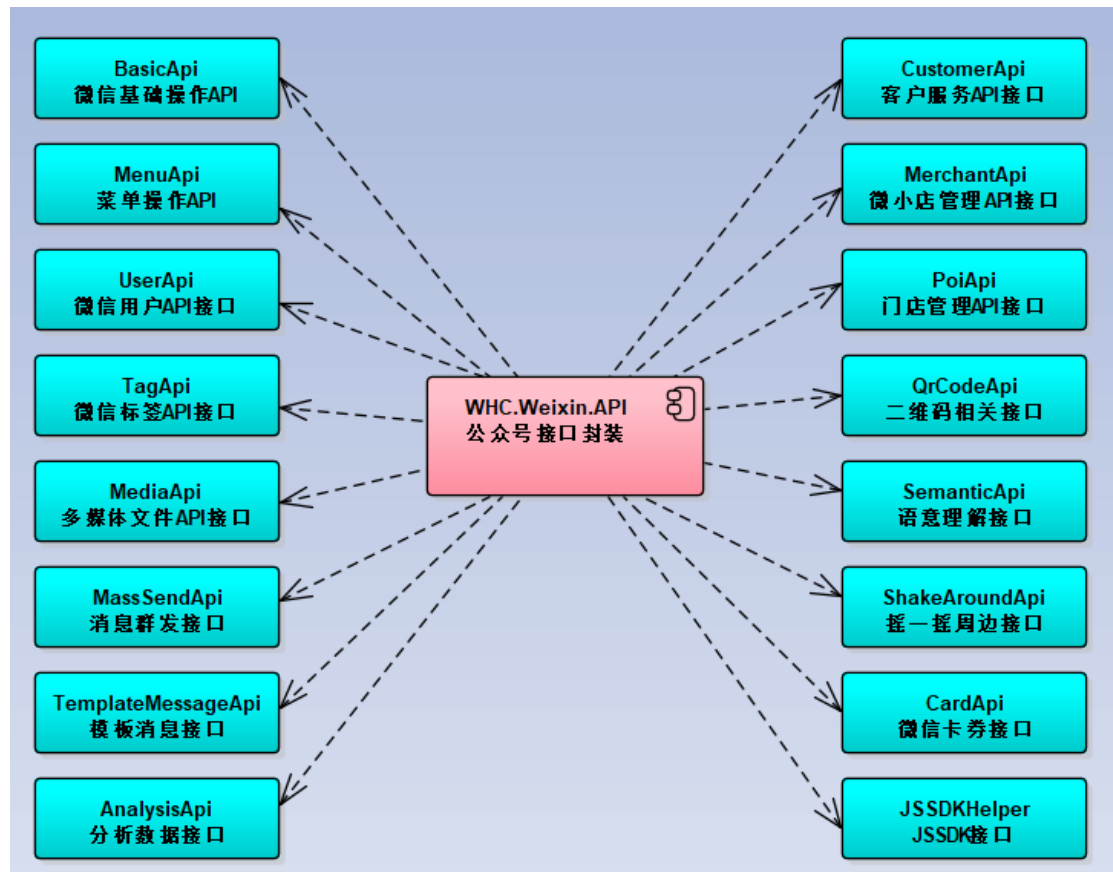
### 2.3.5. 基础类库支持部分

另外，微信框架还包括基础类库支持部分内容，包括权限管理、字典管理、附件管理等基础内容。



### 2.3.6. 公众号和企业微信实现接口

微信开发，我们首先需要利用我们的语言（这里是利用 C# 语言），为所有用到的 API 接口实现进一步的封装，方便使用，微信 API 模块包含的内容很多，其中公众号接口封装包括内容如下所示。



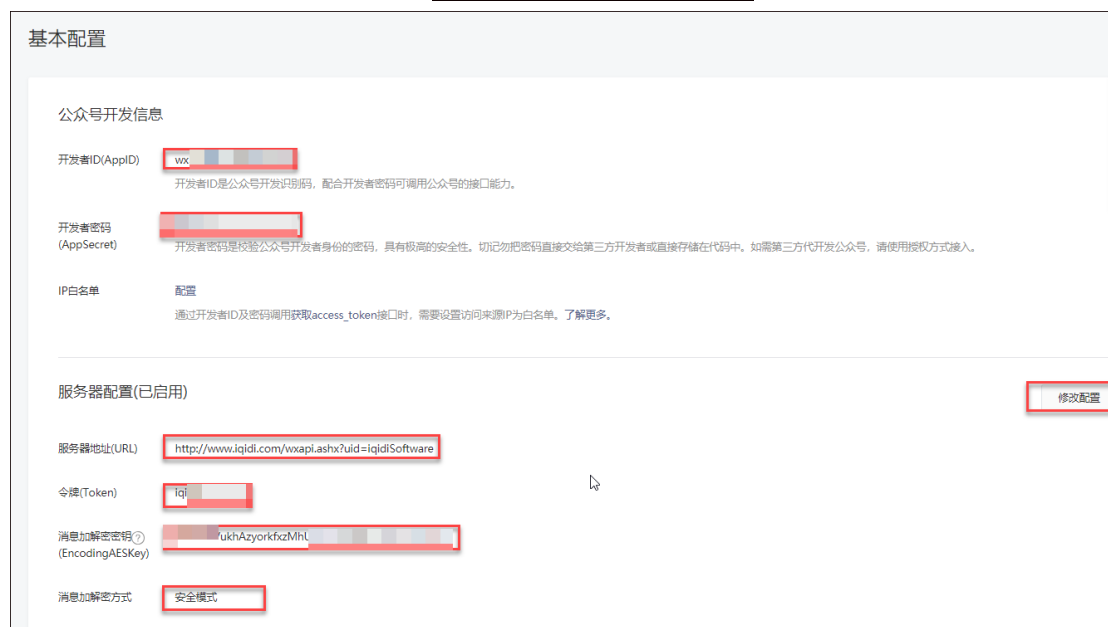
## 2.4. 账号配置

### 2.4.1. 微信公众号配置

不管是微信公众号，还是企业微信，都需要申请账号才能使用，并且最好通过认证，这样才能具有更多的接口权限使用。

微信公众号的注册申请参考文章：<http://www.cnblogs.com/wuhuacong/p/3613826.html>。

输入账号密码，登录微信公众号后台 <https://mp.weixin.qq.com>，然后扫码确认登陆，在左侧【开发】【基本配置】中获取对应的账号配置参数。



在【修改配置】中进行修改内容，如下界面所示。

### 基本配置

[基本配置](#) / 填写服务器配置

请填写接口配置信息，此信息需要你拥有自己的服务器资源。  
填写的URL需要正确响应微信发送的Token验证，请阅读接入指南。

URL   
必须以http://或https://开头，分别支持80端口和443端口。

Token   
必须为英文或数字，长度为3-32字符。  
什么是Token?

EncodingAESKey    
消息加密密钥由43位字符组成，可随机修改，字符范围为A-Z, a-z, 0-9。  
什么是EncodingAESKey?

消息加解密方式 请根据业务需要，选择消息加解密类型，启用后将立即生效

- 明文模式  
明文模式下，不使用消息体加解密功能，安全系数较低
- 兼容模式  
兼容模式下，明文、密文将共存，方便开发者调试和维护
- 安全模式 (推荐)  
安全模式下，消息包为纯密文，需要开发者加密和解密，安全系数高

然后启动微信管理后台系统（一般启动，本地地址为 <http://localhost:32973/Login/Index>），  
如果部署在服务器后，请通过域名进行访问即可。

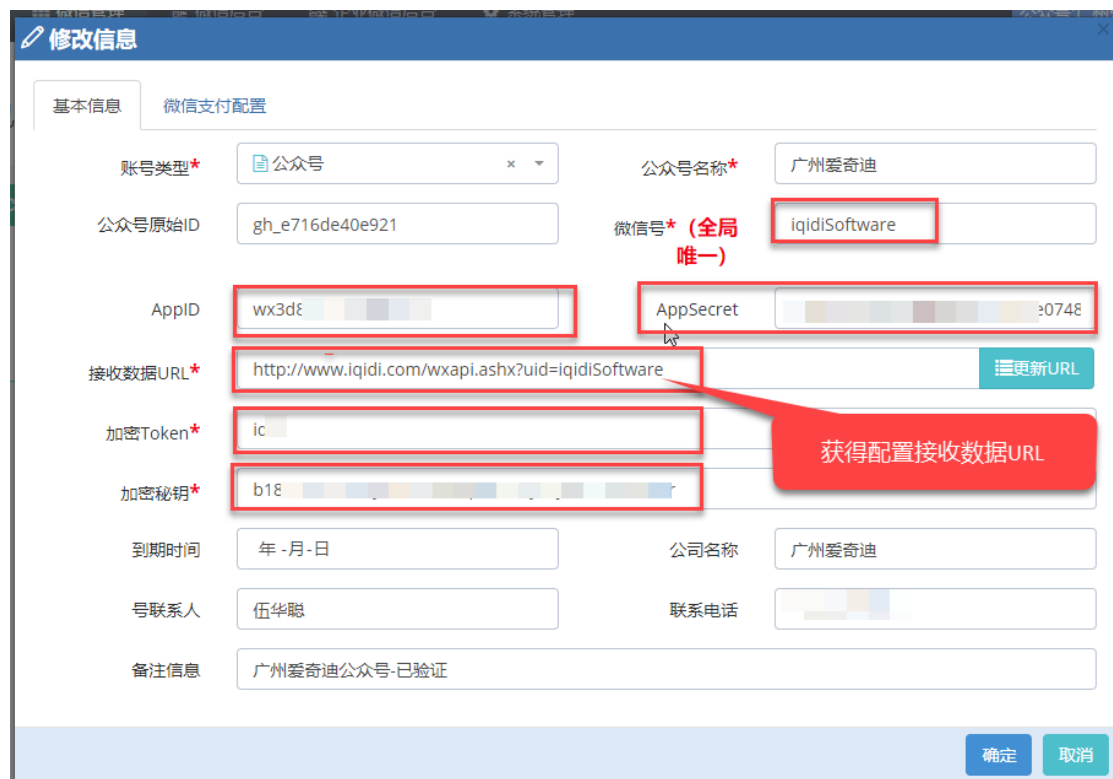


登陆后台系统后，在【微信管理】【微信账号配置】管理界面里面进行维护对应账号信息，如下界面所示。





修改或者增加相关的公众号账号信息。（根据微信公众号参数，录入红色方框的内容）



再次回到刚才后台的界面，修改对应的 URL，提交参数，如提示【提交成功】即可正确完成配置。



## 2.4.2. 企业微信配置

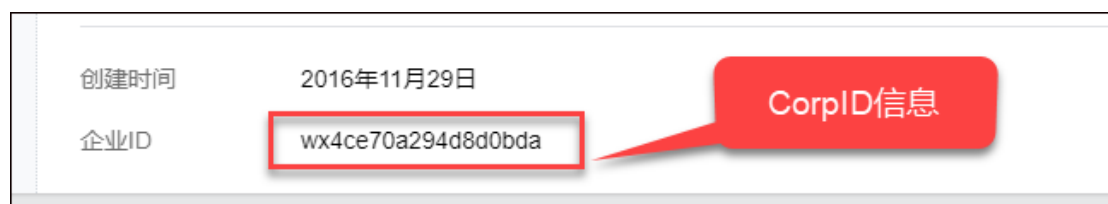
微信企业微信的注册申请参考文章：<http://www.cnblogs.com/wuhuacong/p/3991735.html>。

企业微信后台登录地址为：<https://work.weixin.qq.com>，扫码登录进行获取对应的参数，找到【我的企业】【企业信息】界面，如下界面所示。



然后滚动到底部，找到对应的企业 ID，如下界面所示，这个就是后面填写需要用到的 CorpID。

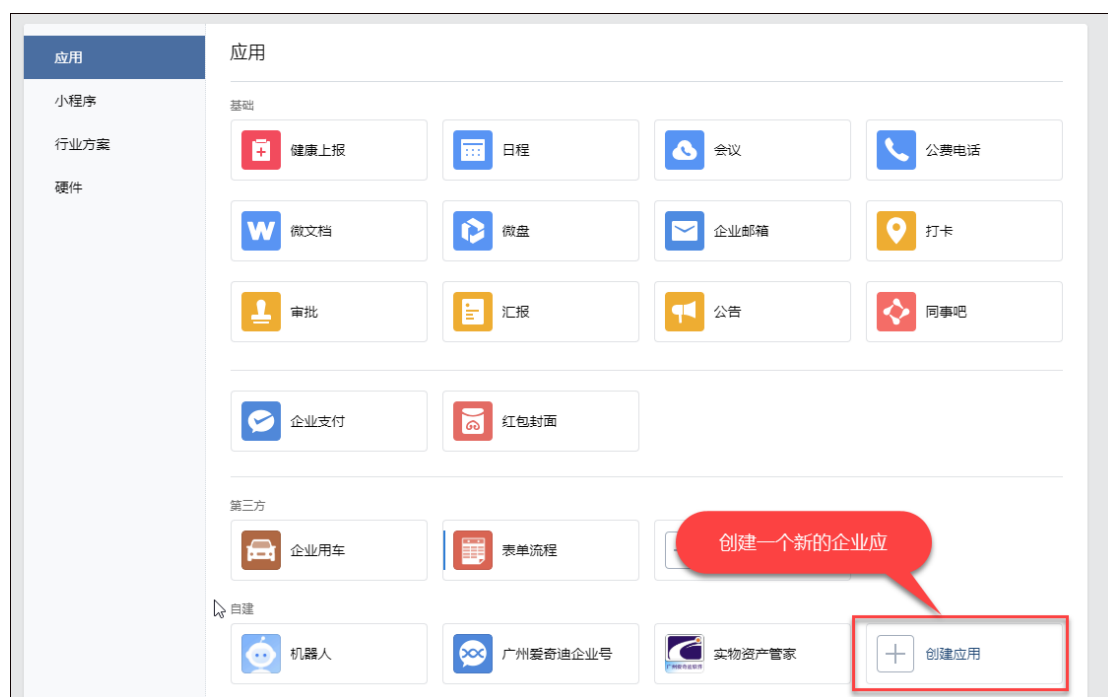
文件名称：微信门户开发框架-使用指导说明书



再次切换到【应用管理】【应用】选项卡里面。



在应用底部，找到【创建应用】按钮，创建一个新的应用；或者单击进入现有的应用进行查看参数。



查看创建了的应用信息，如下界面所示。



记录下前面的 CorpID、CorpSecret、加密 Token、加密密钥等参数。

然后启动 Web 管理后台，登录后台系统后，在【微信管理】【微信账号配置】管理界面里面进行维护对应账号信息，如下界面所示。



在账号管理界面里面，选择企业微信，修改或者增加相关的账号信息。**注意，不同的企业微信应用需要创建不同的账号。如一个企业微信下面有两个应用，则需要创建两个账号来管理，以便对应不同的应用菜单。**



修改信息

基本信息 微信支付配置

账号类型\* 企业微信 公众号名称\* 广州爱奇艺企业微信

公众号原始ID 公众号原始ID...

微信号\*(全局唯一) iqidi

CorpID wx4ce... CorpSecret ru...3Xr24

企业应用AgentId\* 1000004

接收数据URL\* http://www.iqidi.com/corpapi.ashx?uid=iqidi 更新URL

加密Token\* iqi

加密密钥\* VOsBMre2rYiI...

到期时间 年-月-日 公司名称 广州爱奇艺

号联系人 伍华聪 联系电话

备注信息 广州爱奇艺企业微信

确定 取消

再次回到企业微信后台，录入上面获得的接收 URL，然后保存成功即可，如下界面所示。

企业微信 API文档 联系客服 退出

首页 通讯录 应用管理 客户联系 管理工具 我的企业

返回 保存成功

接收消息服务器配置

URL http://www.iqidi.com/corpapi.ashx?uid=iqidi

Token iqi

EncodingAESKey VOsBMre2rYiI... 编辑 删除

接收的消息事件类型

用户发送的普通消息, 自定义菜单操作, 上报地理位置, 上报进入应用事件, 审批状态通知事件

### 2.4.3. 测试文件配置

测试模块里面的配置文件，如果要正常测试 API，也需要配置好对应的配置项。

```
app.config  + x
<?xml version="1.0"?>
<configuration>
  <configSections>
    <section name="dataConfiguration" type="Microsoft.Practices.EnterpriseLibrar
  </configSections>
  <connectionStrings>
    <add name="sqlserver" providerName="System.Data.SqlClient" connectionString=
    <add name="sqlite" providerName="System.Data.SQLite" connectionString="Data
  </connectionStrings>
  <dataConfiguration defaultDatabase="sqlite">
    <providerMappings>
      <add databaseType="EntLibContrib.Data.SQLite.SQLiteDatabase, EntLibContrib
    </providerMappings>
  </dataConfiguration>
  <appSettings>
    <!-- 组件的数据库类型 : access、sqlserver、sqlite等-->
    <add key="ComponentDbType" value="sqlite"/>
    <!-- 系统名称-->
    <add key="AppName" value="企业信息化平台管理系统"/>

    <!-- 微信的Token-->
    <add key="WeixinToken" value="iqidi"/>
    <add key="AppId" value="" />
    <add key="AppSecret" value="" />
  </appSettings>
</configuration>
```

企业微信的测试模块配置。

```
App.config  + x  app.config
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <!-- 企业号配置信息-->
    <add key="CorpToken" value="iqidi"/>
    <add key="CorpId" value="" />
    <add key="EncodingAESKey" value="" />
    <add key="CorpSecret" value="" />
  </appSettings>

  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
    <supportedRuntime version="v2.0.50727"/>
  </startup>
</configuration>
```

## 2.5. 开发过程处理

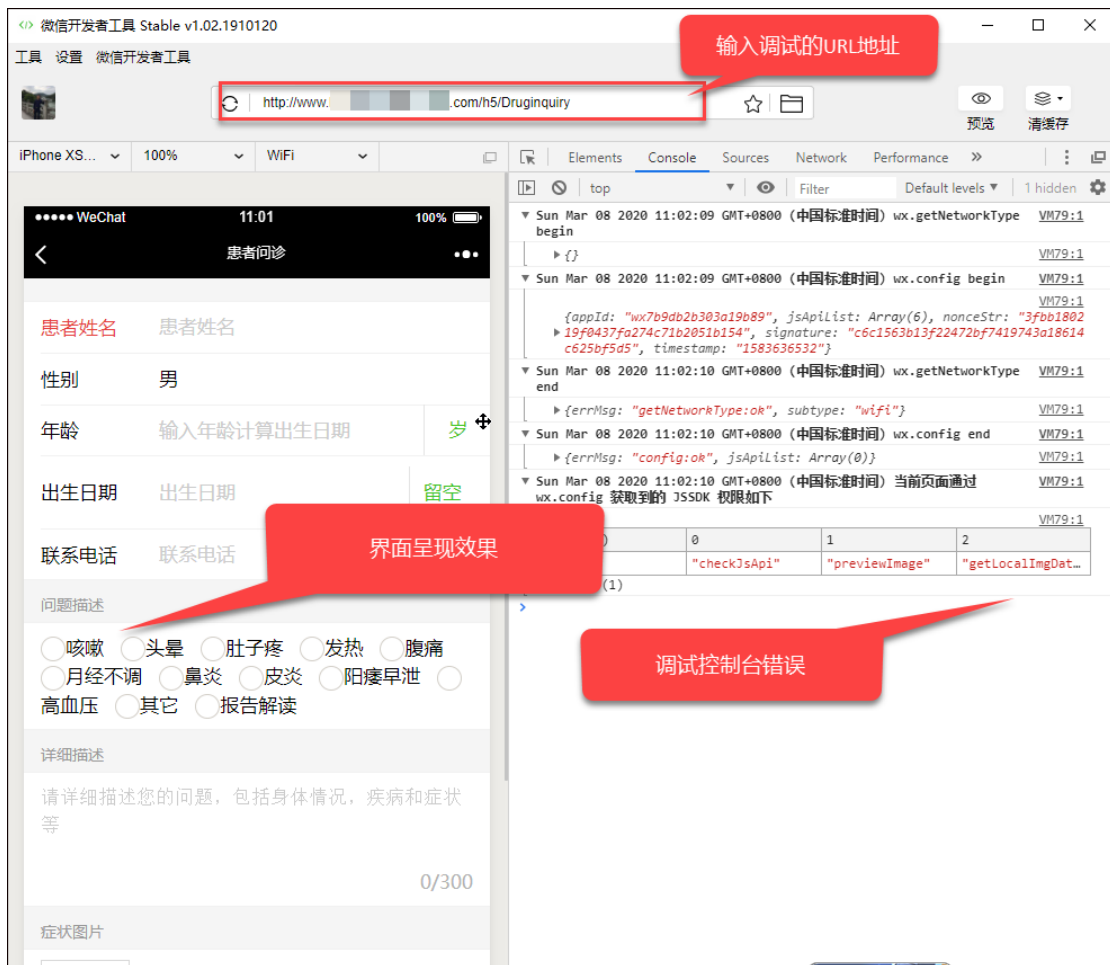
微信公众号和企业微信，主要就是利用微信后台管理系统的入口，创建对应应用 H5 页面。本章节主要介绍基于微信框架的基础上，开发对应的 H5 页面过程，从而实现微信公众号或者企业应用的菜单对接或者事件对接处理。

## 2.5.1. 微信开发者工具

微信公众号或者企业微信开发，一般需要进行相关的网页页面调试，请下载微信提供的《微信开发者工具》，下载地址为：

<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>。

软件启动后即可扫码进入，例如我们对其中一个页面进行调试，界面如下所示。



一般情况下，我们的 H5 页面需要进行重定向获取用户的信息，因此也需要在公众号后台允许该工具重定向获取对应的用户信息，因此需要在后台绑定账号。





然后在下面界面中绑定开发者账号即可。



## 2.5.2. 微信 H5 页面及 JSDK 开发

在我们开发微信页面的时候，需要大量用到了各种呈现的效果，一般可以使用 Bootstrap

的效果来设计不同的页面，不过微信团队也提供很多这方面的资源，包括 JSSDK 的接口，以及 Weui 的页面样式和相关功能页面，给我们提供了很大的便利。

## 1) JSSDK

微信 JS-SDK 是[微信公众平台](#)面向网页开发者提供的基于微信内的网页开发工具包。通过使用微信 JS-SDK，网页开发者可借助微信高效地使用拍照、选图、语音、位置等手机系统的能力，同时可以直接使用微信分享、扫一扫、卡券、支付等微信特有的能力，为微信用户提供更优质的网页体验。

目前 JSSDK 支持的接口分类包括下面几类：基础接口、分享接口、图像接口、音频接口、智能接口、设备信息、地理位置、摇一摇周边、界面操作、微信扫一扫、微信小店、微信卡券、微信支付，随着微信功能的全部整合，估计更多的接口会陆续开放出来。

## 2) WeUI 和 JQuery WeUI

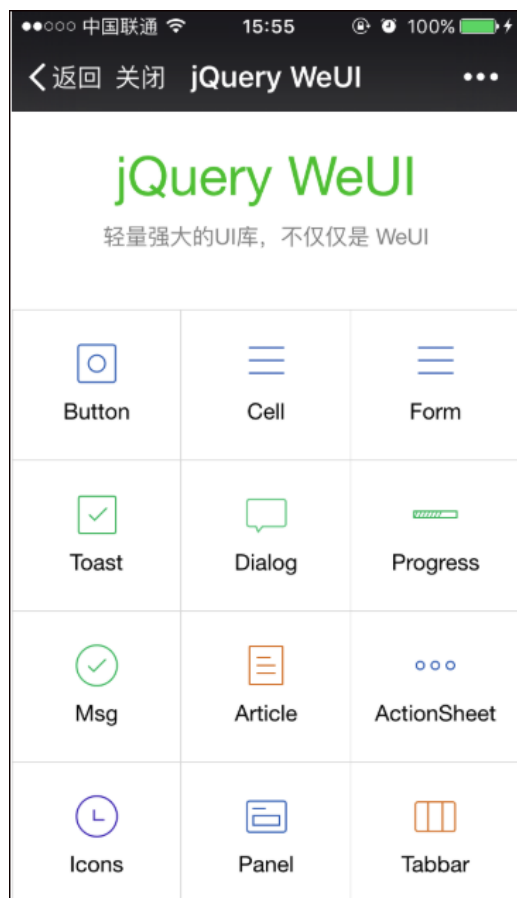
**WeUI** 是一套同微信原生视觉体验一致的基础样式库，由微信官方设计团队为微信内网页开发量身设计，可以令用户的使用感知更加统一。在微信网页开发中使用 WeUI，有如下优势：

- 同微信客户端一致的视觉效果，令所有微信用户都能更容易地使用你的网站
- 便捷获取快速使用，降低开发和设计成本
- 微信设计团队精心打造，清晰明确，简洁大方

该样式库目前包含 button、cell、dialog、progress、toast、article、icon 等各式元素，已经在 [GitHub](#) 上开源。访问 <http://weui.github.io/weui/> 或微信扫码即可预览。

**jQuery WeUI** 中使用的是官方 WeUI 的 CSS 代码，并提供了 jQuery/Zepto 版本的 API 实现。jQuery WeUI 相对于在官方 WeUI 的基础上做了一些功能扩展，已丰富界面设计和相关功能，因此我们可以考虑直接基于 JQuery WeUI 的基础上进行页面开发即可。下载地址：<http://jqweui.com/>。

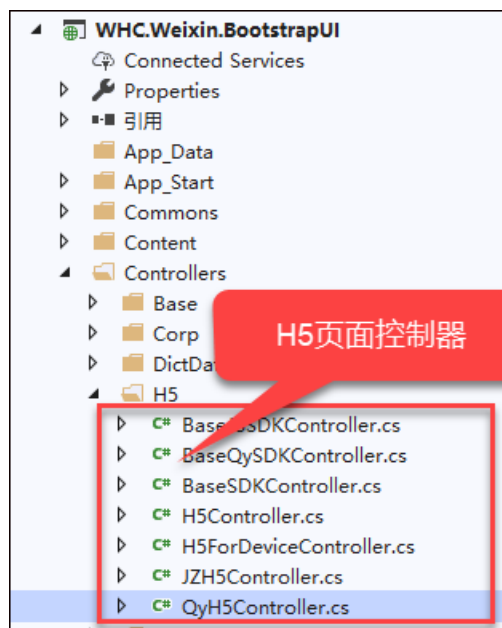
在微信框架里面的一些案例中，都利用了 We UI 样式来进行很多微信 H5 页面的功能设计，我们主要利用 JQuery WeUI 库来实现相对应的一些控件处理功能。



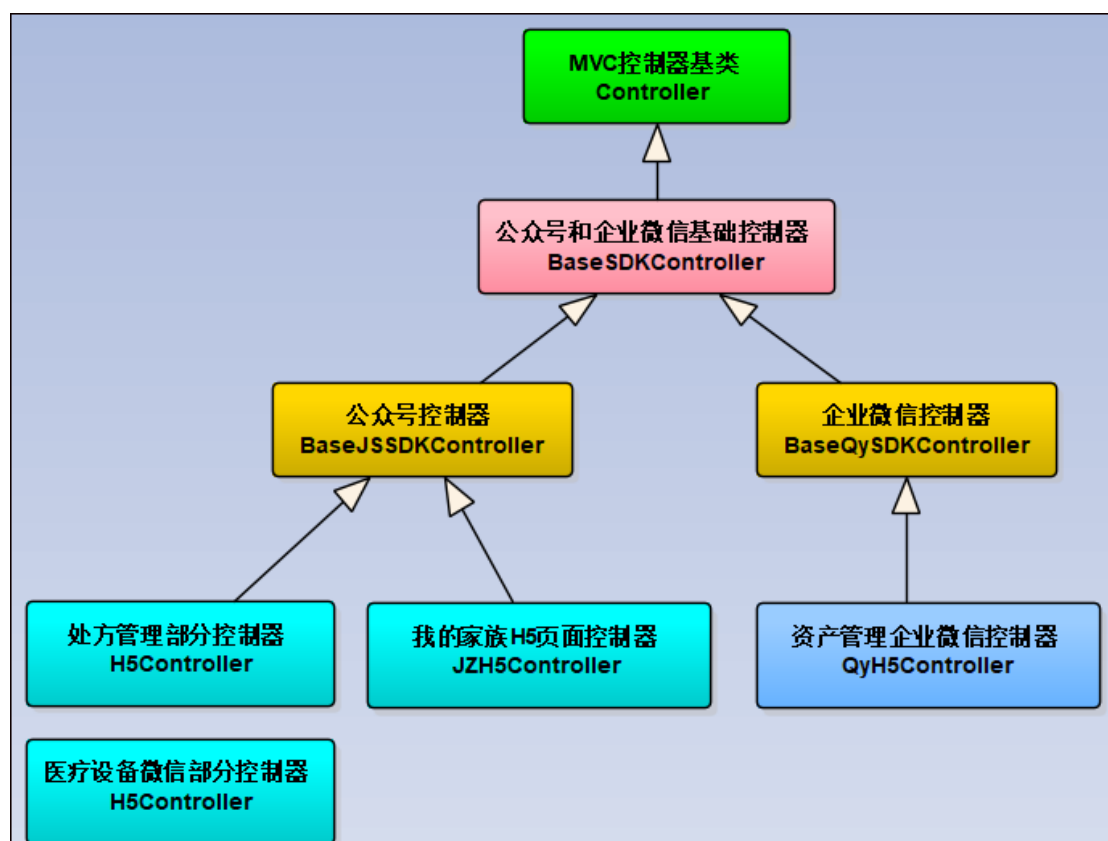
### 2.5.3. 微信框架中 H5 控制器关系

我们知道，H5 页面也是基于 Asp.net 的开发模式，因此 H5 页面的入口也是 MVC 的控制器。为了实现简单、高效的原则，我们一般采用了一定层次的控制器继承关系，因此 MVC 的控制器继承关系需要了解一下。

在解决方案控制器目录如下所示。



这些控制器的继承关系如下所示，继承能够极大程度利用基类的公用函数，减少子类的代码臃肿。



通过上面的继承关系，我们可以看到公众号和企业微信部分，有一个基类控制器是实现两者共同的功能的。然后再次分支，分为公众号控制器其或者企业微信控制器，这两个对于用户来说也是一个基类，我们一般在这个基础上创建自己的子类控制器，以便实现差异化的

功能。针对公众号和企业号的处理，它们的接口对象不同，因此处理代码也是不同的，这个即使他们分开的原因。

例如，基于上面的继承关系，如果我们需要创建一个新的公众号模块“ABC”，那么我们只需要继承自 BaseJSSDKController，创建一个 ABCController 控制器，在里面实现自己的业务控制器接口即可；对于企业微信也一样，如果我们要创建一个新的企业应用“XYZ”，那么我们只需要继承自 BaseQySDKController，创建一个 XYZController 控制器即可。

## 2.5.4. 账号微信菜单管理

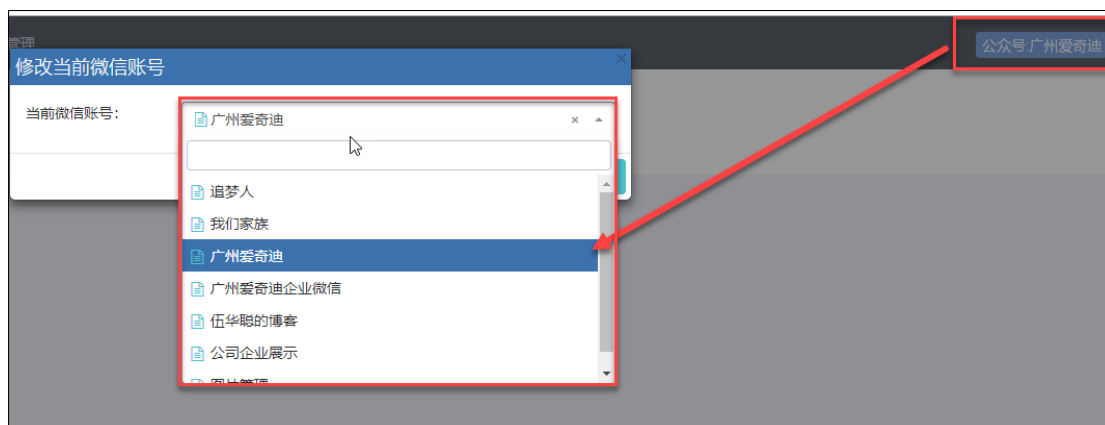
启动微信管理后台系统（一般启动，本地地址为 <http://localhost:32973/Login/Index>），如果部署在服务器后，请通过域名进行访问即可。



进入对应的【微信菜单】模块功能，如下所示。



由于不同账号的菜单不一样，因此注意从右上角选择之前创建好的公众号账号。



默认框架提供了我们标准的一些菜单，如果选择您新建的账号，可能菜单栏是空白的。

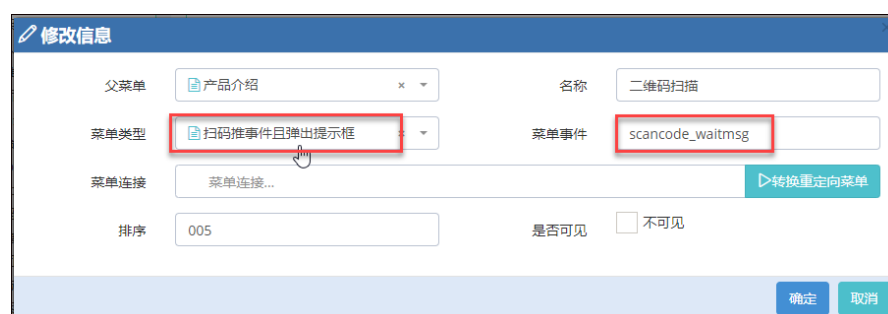
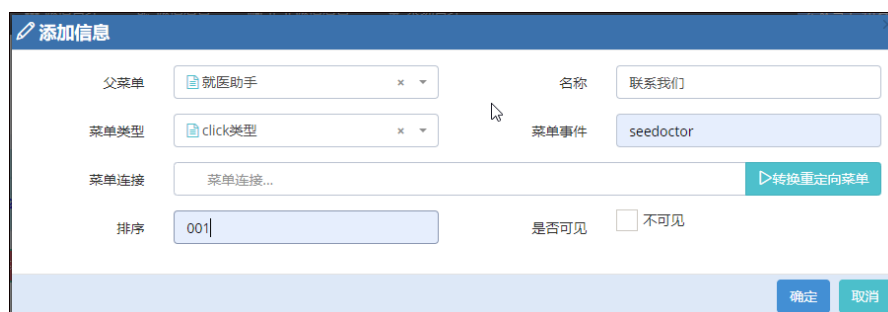
**注意：**为了方便，我们一般允许多个菜单分组存在，但只有一个是启用的（黑色），其他分组是禁用的（红色），提交菜单到微信，也只是提交当前启用的菜单分组内的菜单列表。



如果是空白的，请先创建一个菜单分组，然后需要单击【进入菜单维护】进行菜单的创建操作。



菜单可以新增，然后指定对应的类型即可，如是 Click 类型，必须指定事件；如果是 View 类型，必须指定 URL。菜单是树形列表，你可以在某个菜单下面创建子菜单。菜单的文字大小、菜单项目等都有限制，必须遵循微信指定的规则创建。如果菜单设置为不可见，那么该项菜单及其下面的菜单列表都不会提交到微信服务器。菜单排序是用于确定菜单列表的顺序位置的。



常见的一些菜单列表如下所示。



<input type="checkbox"/>	名称	菜单类型	菜单事件	菜单连接	排序	是否可见	操作
<input type="checkbox"/>	关于我们	click	event_company		001	可见	
<input type="checkbox"/>	联系我们	click	event_contact		002	可见	
<input type="checkbox"/>	极速问诊	view		http://www.iqidi.com/h5/DrugInquiry	003	可见	
<input type="checkbox"/>	就医指南	click	seedoctor		004	可见	
<input type="checkbox"/>	支付码	click	paymentcode		005	可见	

菜单的 URL 我们要注意填写，例如对于 <http://www.iqidi.com/h5/Myinfo>，这个地址前面部分是域名，h5 代表是 H5Controller，MyInfo 代表 H5Controller 控制器或其基类里面的一个方法入口。这个地址也可以在方法里面进行重定向，从而可以获取用户的 openid 等身份信息。

```
/// <summary>
/// 我的信息页面
/// </summary>
public ActionResult MyInfo()
{
    if (!ConfigData.EnableNormalBrowser) {...}

    Session["back_url"] = Request.Url.AbsoluteUri;
    AuthOpenId();

    //通过重定向的code获取对应的OpenID
    var openid = (string)Session["openid"];
    if (!string.IsNullOrEmpty(openid))
    {
        ViewBag.openid = openid;

        //系统用户信息
        var userInfo = BLLFactory<User>.Instance.FindByO
        ViewBag.Binded = (userInfo != null); //用户是否绑

        return View("MyInfo");
    }
}
```

菜单维护好后，我们可以通过【提交菜单到微信】功能，发布我们编辑好的菜单列表。





如果发布成功，那么我们公众号的菜单就会及时更新；如果有错误信息，请检查菜单文字的大小、URL 格式和大小、菜单项目的数量、事件代码是否填写等是否符合微信的要求。

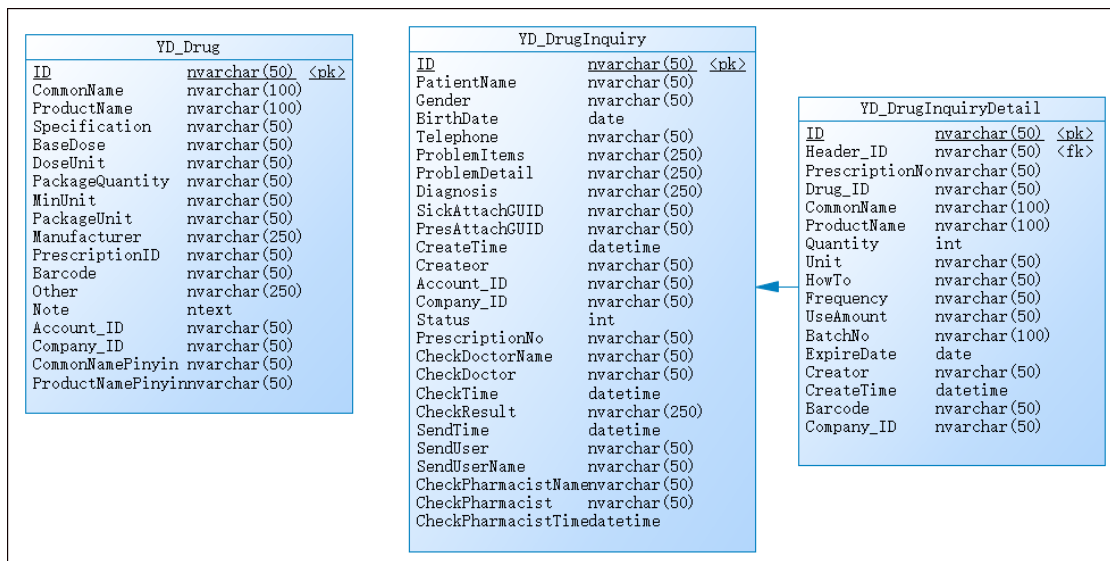
## 2.5.5. 微信公众号开发

公众号功能的开发，除了前面我们在后台创建账号，并录入账号的账号名、AppID、AppSecret、Token、URL、EncodingAESKey 并在公众号配置成功后，我们只是完成了第一步和微信服务器的对接工作。接下来需要基于这个账号下面创建对应的菜单，以及实现菜单对应的 H5 页面控制器和视图代码的处理。这两个步骤都做好了，我们就可以着手对应公众号 H5 页面的功能开发了。

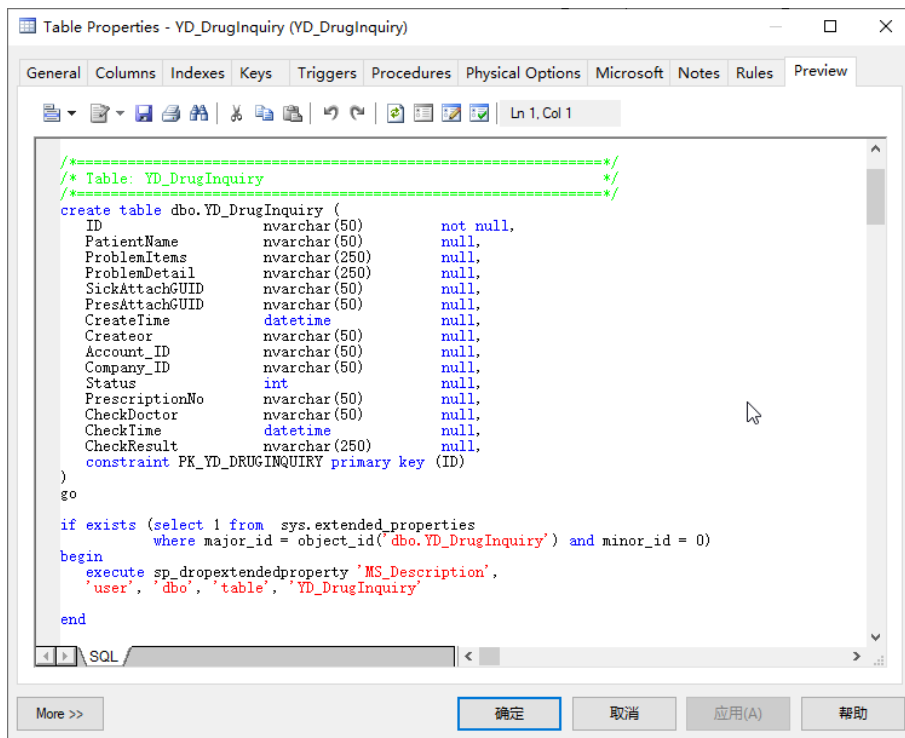
每开发一个应用功能，我们都需要根据我们的业务场景，设计好对应的数据库表，然后根据业务表删除对应的框架代码 BLL/DAL/IDAL/Entity 等这些，增量复制到框架对应位置即可，这个和我们开发其他应用的模式一样。下面我们以处方审核项目为例进行介绍。

### 1) 数据库设计

我们根据业务需要，创建对应的业务表即可。这里涉及一个药品表和一个处方记录表即可，如下截图所示是基于 PowerDesigner 工具进行的设计视图。表的设计，一般来说需要有一个 ID 字段，并作为主键，其他根据情况处理即可。



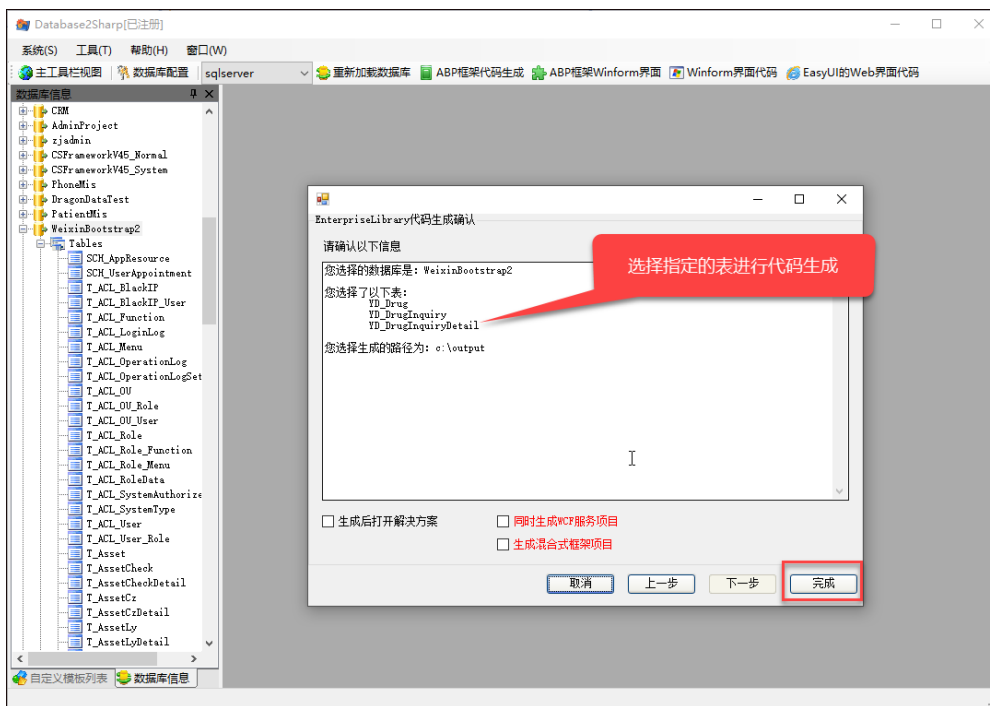
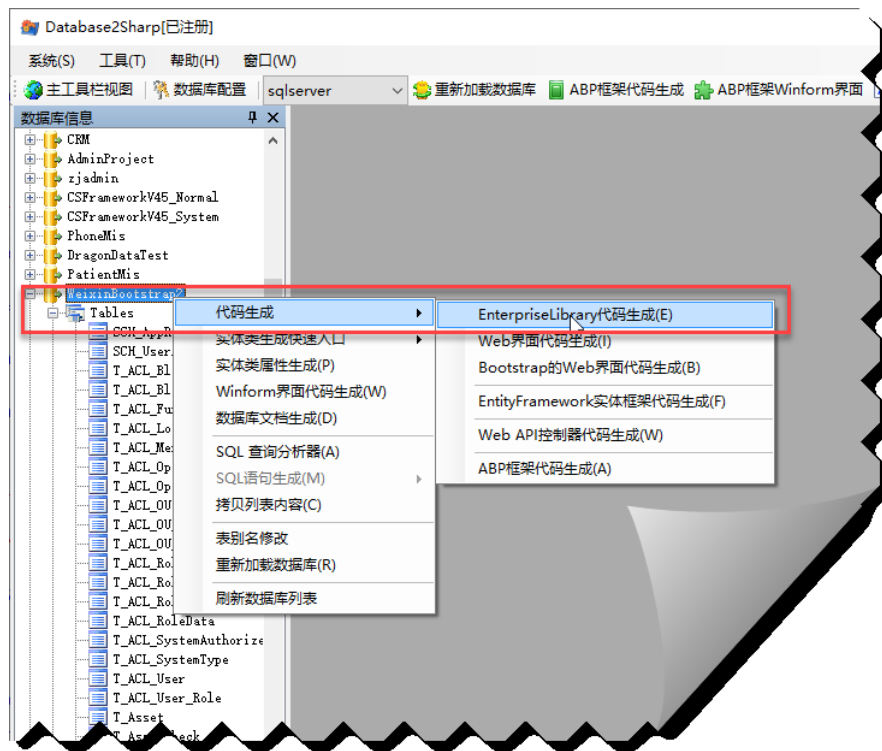
设计好数据库表后，利用工具提供的 SQL 生成脚本，获取脚本，在数据库上执行即可创建对应的表。



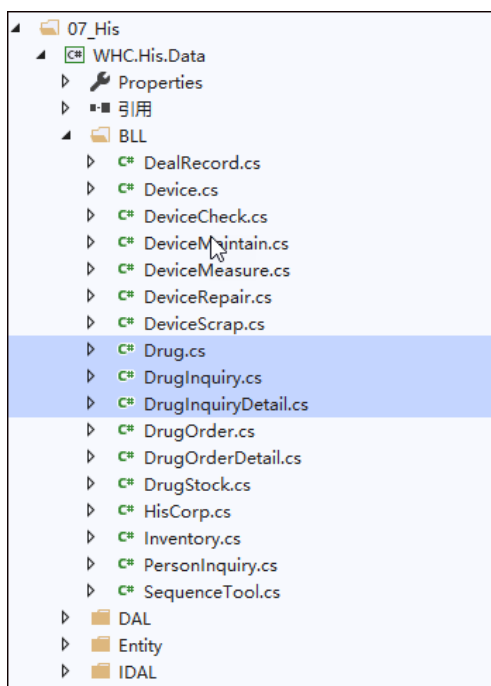
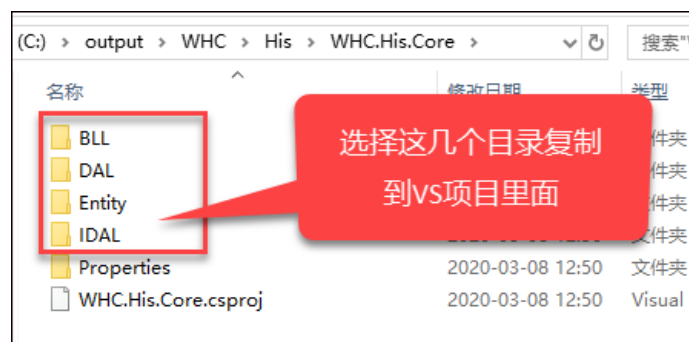
## 2) 生成框架代码

确保生成对应的数据库表后，我们就可以利用代码生成工具 Database2Sharp(代码生成工具下载地址：<http://www.iqidi.com/database2sharp.htm>)进行代码的生成。

我们这里生成代码，只需要选择“【代码生成】【EnterpriseLibrary 代码生成】”即可。

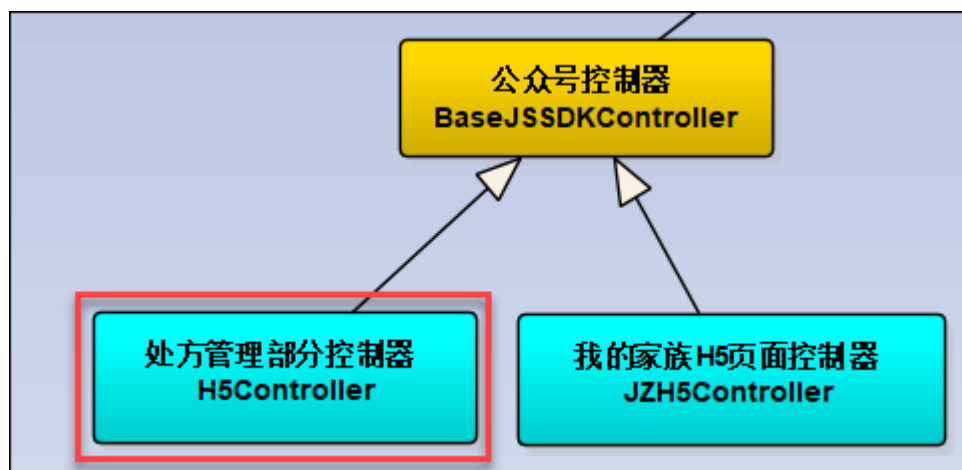


代码生成后把生成的文件复制到 VS 项目中，然后编译测试是否有错误，有错误则修正即可。



### 3) H5 后台代码处理

接下来我们就需要创建一个对应业务的 H5 控制器了，如果我们业务为 ABC，那么你可以定义一个 ABCController，继承自公众号控制器 BaseJSSDKController 即可。我处方管理页面则是定义 H5Controller 控制器名称。



它的控制器定义代码如下所示。

```
using ...

namespace WHC.Weixin.BootstrapUI.Controllers
{
    /// <summary>
    /// H5页面的处理操作控制 (处方管理部分)
    /// </summary>
    public partial class H5Controller : BaseJSSDKController
    {
        #region 患者查询及处方管理

        /// <summary> 药店信息
        public ActionResult DrugShop()...

        /// <summary> 保存药店信息
        public ActionResult DrugShopOK(string openid, FormCollection f...

        /// <summary> 药品信息列表
        public ActionResult DrugList()...

        /// <summary> 药品信息
        public ActionResult DrugInfo(string id)...
```

在 H5 页面里面，如果需要获取用户的 OpenID，则需要重定向才能获取，框架里面封装了一个方法 AuthOpenId 可以实现重定向获取，我们使用如下代码即可。

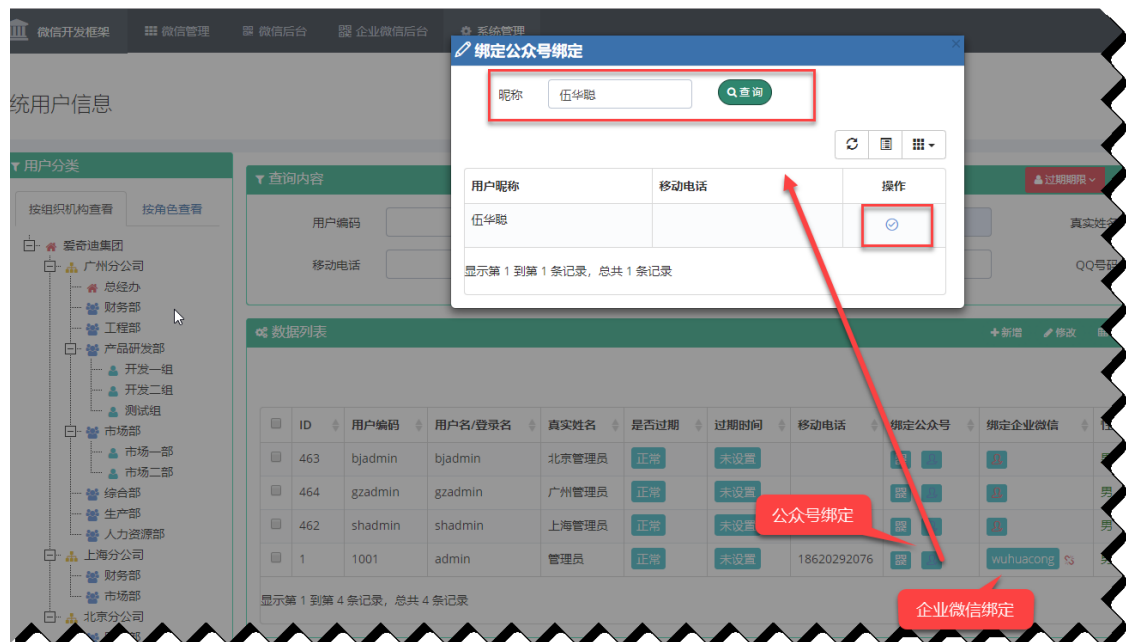
```
/// <summary>
/// 患者问诊
/// </summary>
/// <returns></returns>
public ActionResult DrugInquiry()
{
    Session["back_url"] = Request.Url.AbsoluteUri;
    AuthOpenId();//自动获取当前用户的身份。

    string openid = (string)Session["openid"];
    if (!string.IsNullOrEmpty(openid))...
    else
    {
        return CanNotGetAuthuried();//提示未获取用户身份
    }
}
```

一旦有了用户的 OpenID，就可以通过后台接口获取对应的订阅用户或者系统用户信息（用户关注的时候，订阅用户记录就会写入数据库里面；或者系统用户表绑定对应的 OpenID，可以和系统用户关联），如下代码所示是根据 OpenID 获取绑定的用户记录。

```
var userInfo = BLLFactory<User>.Instance.FindByOpenId(openid);
if (userInfo != null)
{
    //设置药店信息
    ViewBag.Dept_ID = userInfo.Dept_ID;
    ViewBag.DeptName = userInfo.DeptName;
}
```

如果我们需要在后台管理系统中绑定用户，那么我们可以在系统用户管理中进行处理，如下界面所示。



在 H5 页面数据保存的时候，后台需要接受来自页面的数据集合，然后逐一取出进行处理即可，常规的数据可以通过 FormCollection 进行处理，如下代码所示。

```
/// <summary>
/// 患者问诊保存
/// </summary>
/// <returns></returns>
public ActionResult DrugInquirySave(string openid, FormCollection form)
{
    CommonResult result = new CommonResult();
    try
    {
        if (!string.IsNullOrEmpty(openid))
        {
            string ID = form["ID"];
            string PatientName = form["PatientName"];
            string Gender = form["Gender"];
            string BirthDate = form["BirthDate"];
            string Telephone = form["Telephone"];
            string ProblemDetail = form["ProblemDetail"];
            string Diagnosis = form["Diagnosis"];
            string Creator = form["Creator"];
            string ProblemItems = form["ProblemItems"];
            string SickAttachGUID = form["SickAttachGUID"];
            string PresAttachGUID = form["PresAttachGUID"];
        }
    }
}
```

页面数据提交保存

对于记录的保存，一般新增和编辑保存处理都是一个方法，需要我们根据它传过来的ID是否为空判断是新增还是编辑的，如下代码所示。

```
var success = false;
if (!string.IsNullOrEmpty(ID))
{
    success = BLLFactory<DrugInquiry>.Instance.Update(info, info.ID);
}
else
{
    保存的时候，随机上传医生信息
    success = BLLFactory<DrugInquiry>.Instance.Insert(info);
}
```

对于前端需要列表数据，一般通过 MVC 返回对应的 JSON 对象给前端使用即可，如下是分页列表展示数据的函数。

```
/// <summary>
/// 获取药品信息的分页数据
/// </summary>
/// <returns></returns>
public ActionResult FindDrugWithPager()
{
    string where = GetPagerCondition();
    var name = Request["Name"];
    if (!string.IsNullOrEmpty(name))
    {
        where += string.Format@" AND (Barcode like '{0}%' Or CommonName like '{0}%'
        Or ProductName like '{0}%' or CommonNamePinyin like '{1}%'
        Or ProductNamePinyin like '{1}%' ", name, name.ToUpper());
    }

    PagerInfo pagerInfo = GetPagerInfo();
    var sort = GetSortOrder();
    List<DrugInfo> list = null;
    if (sort != null && !string.IsNullOrEmpty(sort.SortName))
    {
        list = BLLFactory<Drug>.Instance.FindWithPager(where, pagerInfo, sort.SortName, sort.IsDesc);
    }
    else
    {
        list = BLLFactory<Drug>.Instance.FindWithPager(where, pagerInfo);
    }

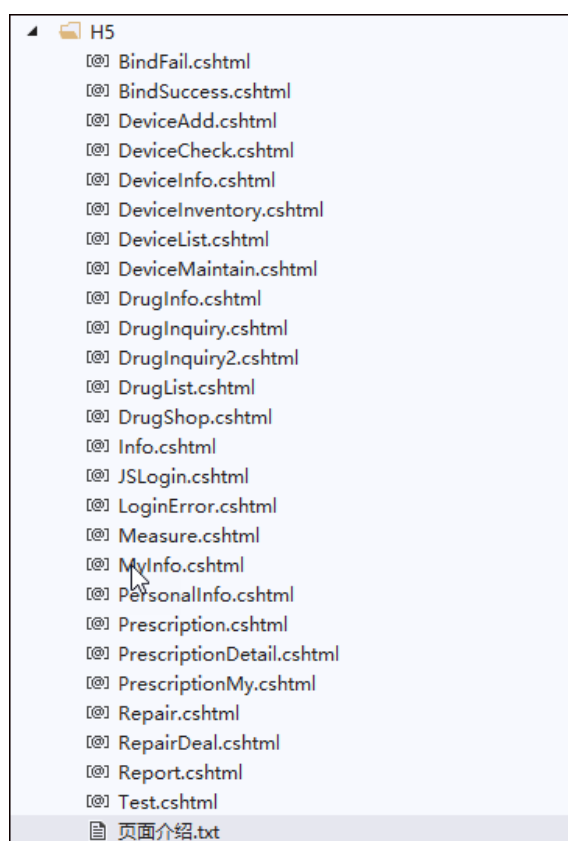
    //Json格式的要求{total:22,rows:{}}
    //构造成Json的格式传递
    var result = new { total = pagerInfo.RecordCount, rows = list }; //如果使用转义, 请使用objList对象
    return ToJsonContent(result);
}
```

返回分页数据列表

#### 4) H5 页面代码处理

H5 页面一般是基于某个 MVC 控制器下进行的视图页面处理，一旦我们创建某个 MVC 控制器，VS 就会创建一个对应的视图目录，如下是 H5Controller 控制器下的视图页面。





页面模块说明如下所示。

```
//通用模块的内容
BindFail.cshtml --- 微信公众号绑定失败处理
BindSuccess.cshtml --- 微信公众号绑定成功处理
Info.cshtml --- 显示一般信息的页面
JSLogin.cshtml --- 使用微信公众号扫码登陆页面
LoginError.cshtml --- 公众号扫码登陆错误页面
Test.cshtml --- 测试页面
Report.cshtml --- 报表测试展示

//医疗设备维修项目页面
DeviceAdd.cshtml --- 设备添加页面
DeviceCheck.cshtml --- 设备检查页面
DeviceInfo.cshtml --- 设备信息页面
DeviceInventory.cshtml --- 设备盘点页面
DeviceList.cshtml --- 设备列表页面
DeviceMaintain.cshtml --- 设备预防性维护页面
Measure.cshtml --- 设备度量页面
Repair.cshtml --- 设备维修工单页面
RepairDeal.cshtml --- 设备微信工单详细页面

//医疗处方审核项目页面
```

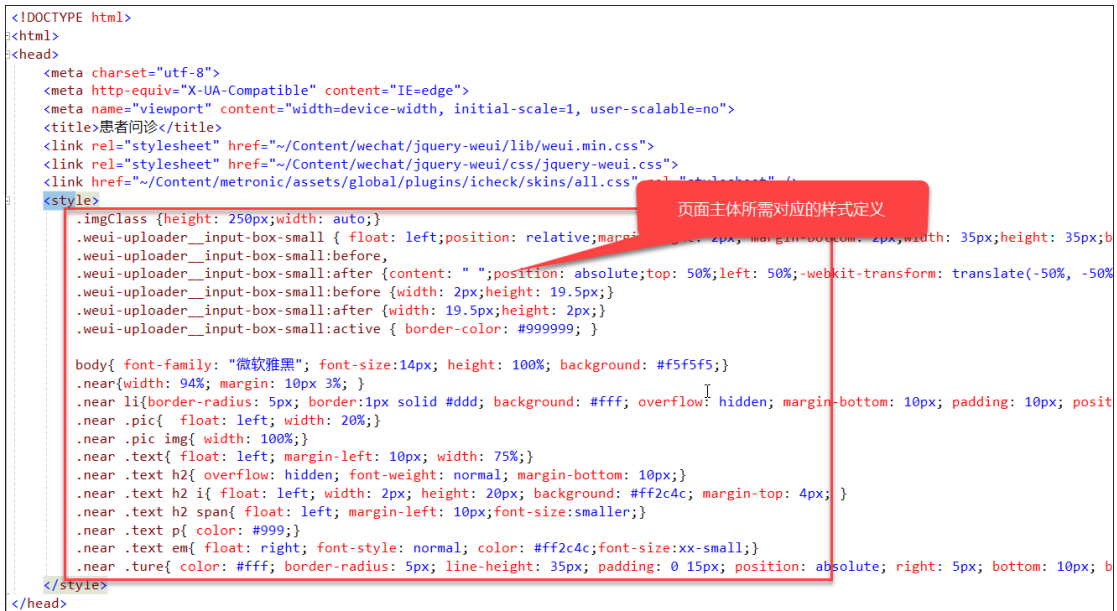
DrugInfo.cshtml --- 药品信息页面  
DrugInquiry.cshtml --- 问诊信息页面  
DrugInquiry2.cshtml --- 问诊信息页面（使用另一种上传方式的处理）  
DrugList.cshtml --- 药品列表页面  
DrugShop.cshtml --- 药店信息页面  
Prescription.cshtml --- 处方信息  
PrescriptionDetail.cshtml --- 处方详细信息  
PrescriptionMy.cshtml --- 本店处方页面  
  
MyInfo.cshtml --- 我的信息  
PersonalInfo.cshtml --- 个人信息

一般的 H5 页面文件头部如下所示，没有母版，指定对应的 JQueryWeui 样式等。



```
1  @{\n2      Layout = null;\n3  }\n4\n5  <!DOCTYPE html>\n6  <html>\n7  <head>\n8      <meta charset="utf-8">\n9      <meta http-equiv="X-UA-Compatible" content="IE=edge">\n10     <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">\n11     <title>患者问诊</title>\n12     <link rel="stylesheet" href="~/Content/wechat/jquery-weui/lib/weui.min.css">\n13     <link rel="stylesheet" href="~/Content/wechat/jquery-weui/css/jquery-weui.css">\n14
```

而其他页面所需的 CSS 样式，我们根据情况参考对应页面 Style 创建即可。



```
<!DOCTYPE html>\n<html>\n<head>\n  <meta charset="utf-8">\n  <meta http-equiv="X-UA-Compatible" content="IE=edge">\n  <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">\n  <title>患者问诊</title>\n  <link rel="stylesheet" href="~/Content/wechat/jquery-weui/lib/weui.min.css">\n  <link rel="stylesheet" href="~/Content/wechat/jquery-weui/css/jquery-weui.css">\n  <link href="~/Content/metronic/assets/global/plugins/icheck/skins/all.css" rel="stylesheet" />\n  <style>\n    .imgClass {height: 250px;width: auto;}\n    .weui-uploader__input-box-small { float: left;position: relative;margin-bottom: 2px;margin-left: 2px;width: 35px;height: 35px;}\n    .weui-uploader__input-box-small:before,\n    .weui-uploader__input-box-small:after {content: " ";position: absolute;top: 50%;left: 50%;-webkit-transform: translate(-50%, -50%);\n    .weui-uploader__input-box-small:before {width: 2px;height: 19.5px;}\n    .weui-uploader__input-box-small:after {width: 19.5px;height: 2px;}\n    .weui-uploader__input-box-small:active { border-color: #999999; }\n\n    body{ font-family: "微软雅黑"; font-size:14px; height: 100%; background: #f5f5f5;}\n    .near{width: 94%; margin: 10px 3%; }\n    .near li{border-radius: 5px; border:1px solid #ddd; background: #fff; overflow: hidden; margin-bottom: 10px; padding: 10px; posit\n    .near .pic{ float: left; width: 20%;}\n    .near .pic img{ width: 100%;}\n    .near .text{ float: left; margin-left: 10px; width: 75%;}\n    .near .text h2{ overflow: hidden; font-weight: normal; margin-bottom: 10px;}\n    .near .text h2 i{ float: left; width: 2px; height: 20px; background: #ff2c4c; margin-top: 4px; }\n    .near .text h2 span{ float: left; margin-left: 10px;font-size:smaller;}\n    .near .text p{ color: #999;}\n    .near .text em{ float: right; font-style: normal; color: #ff2c4c;font-size:xx-small;}\n    .near .ture{ color: #fff; border-radius: 5px; line-height: 35px; padding: 0 15px; position: absolute; right: 5px; bottom: 10px; b\n  </style>\n</head>
```

H5 页面底部需要引入对应的 JS 文件以及编写对应的 JS 脚本代码函数。

```
<script src="~/Content/wechat/jquery-weui/lib/jquery-2.1.4.js"></script>
<script src="~/Content/wechat/jquery-weui/js/jquery-weui.js"></script>
<script type="text/javascript" src="http://res.wx.qq.com/open/js/jweixin-1.2.0.js"></script>
<script src="~/Content/wechat/assets/global/plugins/icheck/icheck.min.js"></script>
<script src="~/Content/wechat/jquery-weui/lib/fastclick.js"></script>
<script src="~/Scripts/H5Util.js"></script>
```

**页面底部引入对应的js文件和编写自定义脚本**

```
<script>...</script>
<script>...</script>
<script>...</script>
<script>...</script>
<!--对编辑界面的处理-->
<script>...</script>
```

**自定义脚本处理函数**

```
</body>
</html>
```

H5 页面的表单，参考 JQuery WeUI 的表单录入处理样式即可，如下所示。

### 表单

表单是基于列表的布局实现的 包括很多常用的表单控件:

```
<div class="weui-cells weui-cells_form">
  <div class="weui-cell">
    <div class="weui-cell__hd"><label class="weui-label">qq</label></div>
    <div class="weui-cell__bd">
      <input class="weui-input" type="number" pattern="[0-9]*" placeholder="请输入qq号">
    </div>
  </div>
  <div class="weui-cell weui-cell_vc">
    <div class="weui-cell__hd">
      <label class="weui-label">手机号</label>
    </div>
    <div class="weui-cell__bd">
      <input class="weui-input" type="tel" placeholder="请输入手机号">
    </div>
    <div class="weui-cell__ft">
      <button class="weui-vcode-btn">获取验证码</button>
    </div>
  </div>
  <div class="weui-cell">
    <div class="weui-cell__hd"><label for="" class="weui-label">日期</label></div>
    <div class="weui-cell__bd">
      <input class="weui-input" type="date" value="">
    </div>
  </div>
  <div class="weui-cell">
    <div class="weui-cell__hd"><label for="" class="weui-label">时间</label></div>
    <div class="weui-cell__bd">
      <input class="weui-input" type="datetime-local" value="" placeholder="">
    </div>
  </div>
  <div class="weui-cell weui-cell_vc">
    <div class="weui-cell__hd"><label class="weui-label">验证码</label></div>
    <div class="weui-cell__bd">
```

最后可以一个类似下面的表单页面内容。

```
<form id="ffAdd" method="post" novalidate="novalidate">
  <div class="weui-cells weui-cells_form">...</div>
  <div class="weui-cells__title">问题描述</div>
  <div class="weui-cells weui-cells_form">...</div>

  <div class="weui-cells__title">详细描述</div>
  <div class="weui-cells weui-cells_form">...</div>

  <div class="weui-cells__title">症状图片</div>
  <div class="weui-cells weui-cells_form">...</div>

  <div class="weui-cells__title">处方照片</div>
  <div class="weui-cells weui-cells_form">...</div>
  <div class="weui-cells weui-cells_form">...</div>
  <div class="weui-cells__title">
    选择处方药品
  </div>
  <div class="weui-cells">...</div>
  <br /><br />

  <div class="weui_opr_area">
    <p class="weui-btn-area">...</p>
  </div>
  <br />
</form>
```

如对于一些弹出框的选择页面，如选择药品的页面，那么可以通过一个连接进行打开，如下效果所示。





页面代码如下所示。

```
<div class="weui-cells">
  <div id="divDrug">...</div>
  <div class="weui-cell">
    <div class="weui-cell_bd">
      <a href="javascript:;" class="weui-uploader__input-box-small open-popup" data-target="#half">
      </a>
    </div>
  </div>
</div>
<br /><br />
<div class="weui_opr_area">
  <p class="weui-btn-area">...</p>
</div>
<br />
</form>

<div id="half" class='weui-popup__container'>
  <div class="weui-popup__overlay"></div>
  <div class="weui-popup__modal">...</div>
</div>
```

弹框按钮代码

弹框层内容

对于输入框里面即时从服务端获取数据的操作，我们一般通过绑定文本控件的“input propertychange”事件进行响应的，通过获取服务端的 JSON 数据，然后组装层对应的界面 HTML 代码后，把它赋值给某个控件就可以实现内容的动态展示了，如下是药品查询并显示在界面上的代码。

```
<script>
    $('#searchInput').bind('input propertychange', function () {
        var name = $("#searchInput").val();

        //获取Json对象集合, 并生成数据显示内容
        var url = "/h5/FindDrugWithPager?page=1&rows=5&Name=" + encodeURIComponent(name);
        $.getJSON(url, function (data) {
            $("#searchContent").html("");
            var html = "";
            $.each(data.rows, function (i, item) {
                var background = (i % 2 == 0) ? "background-color:Highlight" : "";
                html += `<li>
                    <div class='pic'><img src='@ViewBag.WebsiteDomain/Content/images/drug.png' /></div>
                    <div class='text'>
                        <h2><i></i><span>` + item.CommonName + `</span><em>` + item.Specification + `</em></h2>
                        <p>商品名: ` + item.ProductName + `</p>
                        <p>厂家: ` + item.Manufacturer + `</p>
                    </div>
                    <a class='ture' href="javascript:;" onclick="AddDrug(` + item.ID + `,` +
                </li>`;
            });
            $("#searchContent").html(html);
        });
    });
};
```

上面代码通过绑定 input 的 propertychange 事件, 根据 URL 地址获取对应的 JSON 数据, 然后遍历集合进行构建 HTML 内容, 最后赋值 HTML 给对应控件, 实现界面内容的实时变化。

## 2.5.6. 企业微信开发

企业微信的 H5 页面开发和公众号的开发类似, 都需要先在后台管理系统中创建对应账号 (注意一个企业微信下不同的应用需要创建不同的账号), 然后在对应账号下面创建好对应的菜单资源, 并同步到微信上即可更新应用的菜单。具体的账号配置参考前面小节关于【企业微信配置】说明。

同样, 每开发一个应用功能, 我们都需要根据我们的业务场景, 设计好对应的数据库表, 然后根据业务表删除对应的框架代码 BLL/DAL/IDAL/Entity 等这些, 增量复制到框架对应位置即可, 这个和我们开发其他应用的模式一样。

### 1) 数据库设计

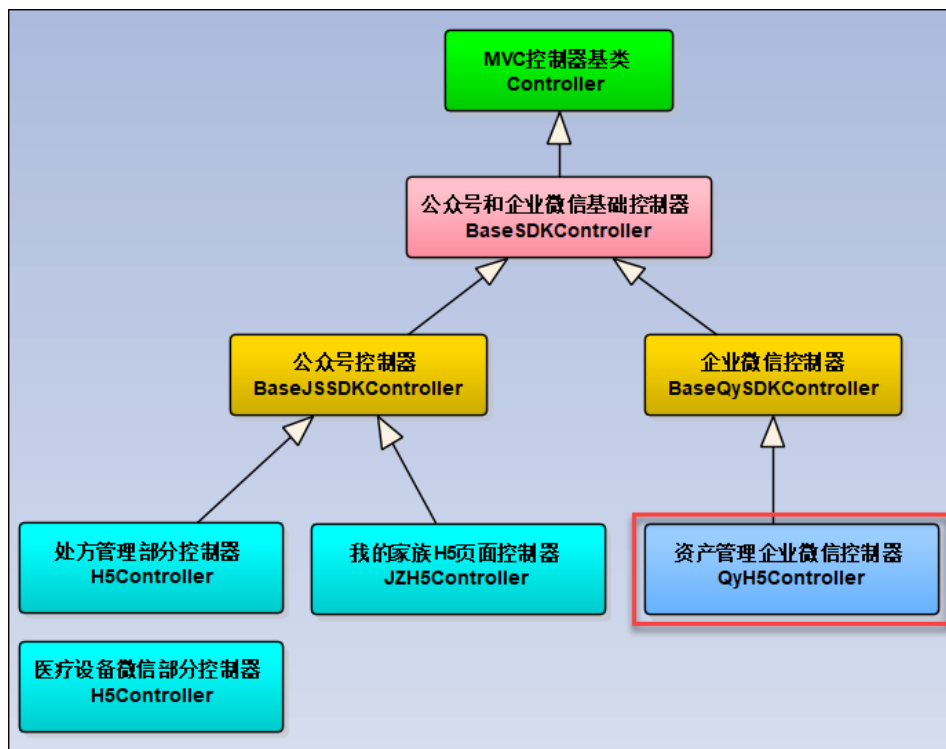
步骤和公众号类似, 此处忽略。

### 2) 生成框架代码

步骤和公众号类似, 此处忽略。

### 3) H5 后台代码处理

前面我们介绍过 H5 控制器类的继承关系, 如下所示。这里我们以资产管理项目的企业微信控制器为例介绍。



QyH5Controller 的定义代码只需要继承自 BaseQySDKController 即可。

```
namespace WHC.Weixin.BootstrapUI.Controllers
{
    /// <summary>
    /// 企业H5页面的处理（基于JSSDK开发业务）
    /// </summary>
    public partial class QyH5Controller : BaseQySDKController
    {
        public QyH5Controller()
        {
        }

        #region 资产管理

        /// <summary>
        /// 实现重定向到H5页面的资产增加状态
        /// </summary>
        /// <param name="code">资产编码</param>
        /// <returns></returns>
        public ActionResult Asset(string code)...

        /// <summary>
        /// 检查资产编码是否重复
        /// </summary>
        /// <param name="code">设备编码</param>
        /// <returns></returns>
        public ActionResult AssetCodeCheck(string code)...

        /// <summary>
        /// 保存设备信息
    }
}
```

和公众号有些不一样，企业微信的用户身份获得的是 user\_id 的内容，这个是对应企业微信的标识的。我们也可以通过重定向页面获得用户身份 user\_id 的内容，如下代码所示。

```
/// <summary>
/// 实现重定向到H5页面的资产增加状态
/// </summary>
/// <param name="code">资产编码</param>
/// <returns></returns>
public ActionResult Asset(string code)
{
    if (!ConfigData.EnableNormalBr // 重定向方式获取用户user_id
    {
        Session["back_url"] = Request.Url.AbsoluteUri;
        AuthUserId();

        string userid = (string)Session["user_id"];
        if (string.IsNullOrEmpty(userid))
        {
            return CanNotGetAuthuried();//提示未获取用户身份
        }
        else
        {
            ViewBag.userid = userid;
            ViewBag.assetcode = code;
            //刷新配置JS的参数
            RefreshTicket();

            return View("AssetAdd");
        }
    }
}
```

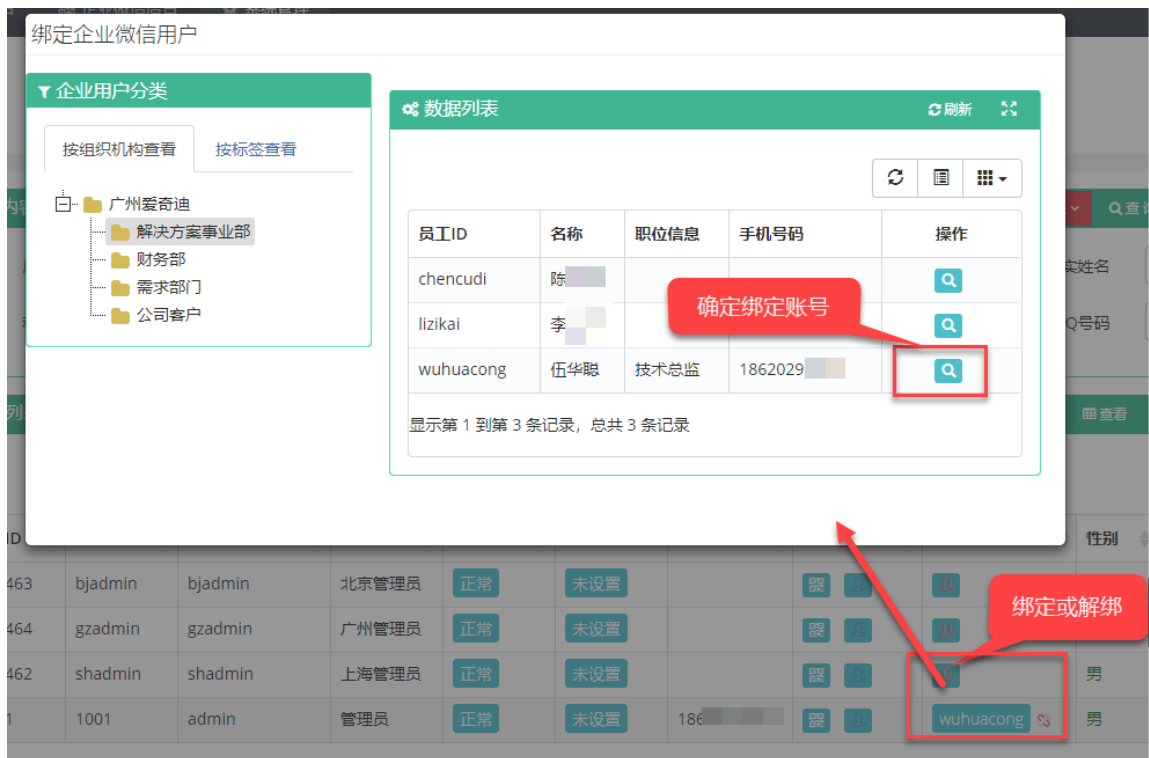
一旦有了用户的 user\_id，就可以通过后台接口获取对应的系统用户信息(系统用户表绑定对应的 user\_id，可以和系统用户关联)，如下代码所示是根据 user\_id 获取绑定的用户记录。

```
BLLFactory<User>. Instance. FindByCorpUserId (userId);
```

如果我们需要在后台管理系统中绑定用户，那么我们可以在系统用户管理中进行处理，如下界面所示。







一般情况下面，对于多个参数的情况，我们也是可以通过 FormCollection 的集合获取，如下函数定义所示。

```
/// <summary>
/// 资产任务盘点
/// </summary>
/// <param name="id">指定的记录ID</param>
/// <param name="userid">操作企业微信用户ID</param>
/// <returns></returns>
public ActionResult AssetCheckOK(string userid, FormCollection form)
{
    if (!ConfigData.EnableNormalBrowser)...

    CommonResult result = null;
    try
    {
        var userInfo = GetUser(userid);
        if (userInfo != null)
        {
            //根据铭牌图片和其他图片，下载图片到本地服务器
        }
    }
}
```

对于前端需要列表数据，一般通过 MVC 返回对应的 JSON 对象给前端使用即可，如下是分  
页列表展示数据的函数。

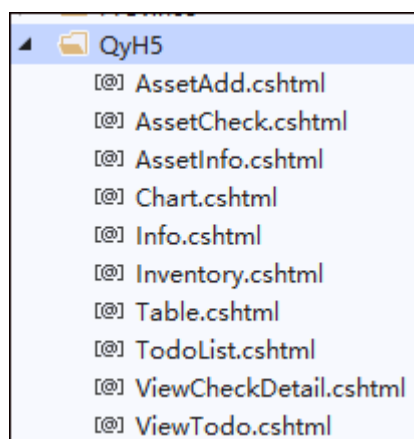
```
/// <summary>
/// 获取资产信息的分页数据
/// </summary>
/// <returns></returns>
public ActionResult FindAssetWithPager()
{
    string where = GetPagerCondition();
    PagerInfo pagerInfo = GetPagerInfo();
    var sort = GetSortOrder();
    List<AssetInfo> list = null;
    if (sort != null && !string.IsNullOrEmpty(sort.SortName))
    {
        list = BLLFactory<Asset>.Instance.FindWithPager(where, pagerInfo, sort.SortName, sort.IsDesc);
    }
    else
    {
        list = BLLFactory<Asset>.Instance.FindWithPager(where, pagerInfo);
    }

    //Json格式的要求{total:22,rows:{}}
    //构造Json的格式传递
    var result = new { total = pagerInfo.RecordCount, rows = list }; //如果使用转义, 请使用objList对象
    return ToJsonContent(result);
}
```

分页函数，返回对应的JSON  
给前端页面使用。

#### 4) H5 页面代码处理

和公众号类似，创建对应企业应用 H5 控制器后，也同样会创建对应的视图目录，如我们创建了 QyH5Controller 控制器，那么对应有 QyH5 的视图目录，我们创建一些我们需要的视图文件页面，配合 QyH5 控制器里面的对应入口，如下所示。



企业微信的应用和公众号的 H5 页面应用本质上没有差异，只需要自己指定对应的样式，创建好对应的 HTML 代码即可。

```
@{
    Layout = null;
}

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
    <title>资产添加</title>
    <link rel="stylesheet" href="~/Content/wechat/jquery-weui/lib/weui.min.css">
    <link rel="stylesheet" href="~/Content/wechat/jquery-weui/css/jquery-weui.min.css">
    <style>
        .imgClass {
            height: 250px;
            width: auto;
        }
    </style>
</head>
```

同样在 H5 页面底部引入对应的 JS 文件，编写对应的 JS 处理函数来处理页面逻辑。

```
@*脚本引用放在此处可以实现自定义函数自动提示*@
<script src="~/Scripts/H5Util.js"></script>
<script src="~/Content/wechat/jquery-weui/lib/jquery-2.1.4.js"></script>
<script src="~/Content/wechat/jquery-weui/js/jquery-weui.min.js"></script>
<script src="~/Content/wechat/jquery-weui/js/swiper.min.js"></script>
<script type="text/javascript" src="http://res.wx.qq.com/open/js/jweixin-1.2.0.js"></script>

<script>...</script>

<script>...</script>
</body>
</html>
```

## 2.6. 数据库配置

### 2.6.1. SqlServer 数据库的配置

- 1) 修改数据库配置信息。

可以通过配置切换为 SqlServer 数据库（同时需要修改 ComponentDbType 的值为 sqlserver）。

```
<dataConfiguration defaultDatabase="sqlserver"/>
```

```
.....
```

```
<add key="ComponentDbType" value="sqlserver"/>
```

- 2) 还原数据库文件。

SqlServer 数据库备份文件是 WeixinApp.bak，该数据库备份文件已经包含了常规的业务数据。如需要在 Sqlserver 数据库上运行系统，恢复数据库到 WeixinApp 即可。

## 2.6.2. MySql 数据库的配置

1) 修改数据库配置信息。

可以通过配置切换为 MySql 数据库（同时需要修改 ComponentDbType 的值为 MySql）。

```
<dataConfiguration defaultDatabase="MySql"/>  
.....  
<add key="ComponentDbType" value="MySql"/>
```

## 2.6.3. SQLite 数据库的配置

SQLite 数据库文件为 Weixin.db（建议自行从 SQLite Developer 导出 Sqlserver 数据库到 SQLite 数据库），其中包含了常规的业务数据。

修改为数据库配置文件，修改 Web.Config 文件中下面两个地方，保持一致数据库类型。

```
<dataConfiguration defaultDatabase="sqlite"/>  
.....  
<add key="ComponentDbType" value=" sqlite "/>
```

## 2.7. 企业微信 Winform 客户端

软件功能界面如下所示，具体使用可以参考《企业微信软件-使用说明书》进行了解配置和功能使用。



### 3. JQuery WeUI 控件的使用

#### 3.1. 界面控件使用

文本控件

日期控件

下拉列表控件

省市区列表级联

附件处理

图片预览和 Gallery