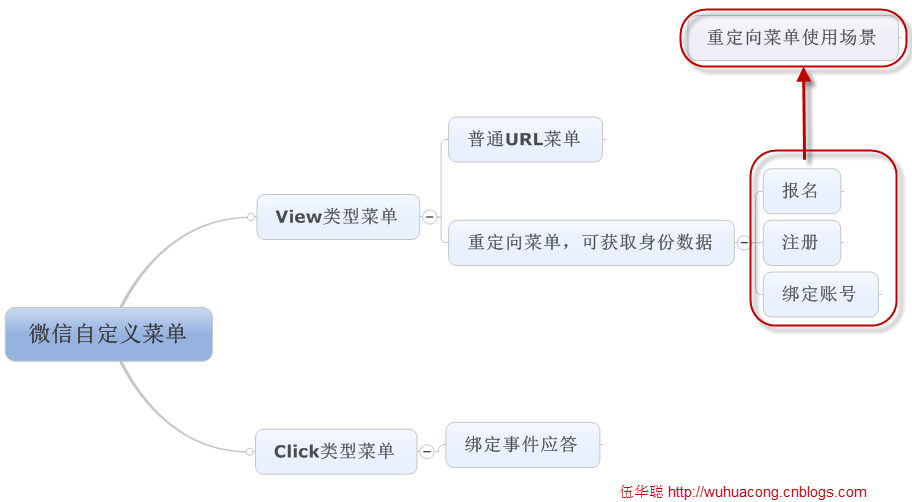
**[C#开发微信门户及应用(14)-在微信菜单中采用重定向获取用户数据](http://www.cnblogs.com/wuhuacong/p/3778972.html)**

我曾经在系列文章中的《[C#开发微信门户及应用(11)--微信菜单的多种表现方式介绍](http://www.cnblogs.com/wuhuacong/p/3735739.html)》中介绍了微信菜单里面的重定向操作，通过这个重定向操作，我们可以获取一个code值，然后获取用户的openID，进而就能获取到更多的用户信息，这个在会员信息的场景里面用的很多，本篇介绍在网站中迅速配置这样的菜单链接，并介绍如何在后台获取相关的用户信息，实现页面数据个性化的展现操作。

我们知道，微信的自定义菜单分为两大类，分别对应Click类型和View类型的，而重定向属于View类型的一种，如下所示。



**1、微信重定向菜单的配置**

微信重定向的菜单，就是通过传入一个地址参数，让微信服务器进行跳转，它的主要规则如下所示。

对于scope=snsapi\_base方式的链接如下：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx3d81fc2886d86526&redirect\_uri=http%3A%2F%2Fwww.iqidi.com%2Ftestwx.ashx&response\_type=code&scope=snsapi\_base&state=123#wechat\_redirect

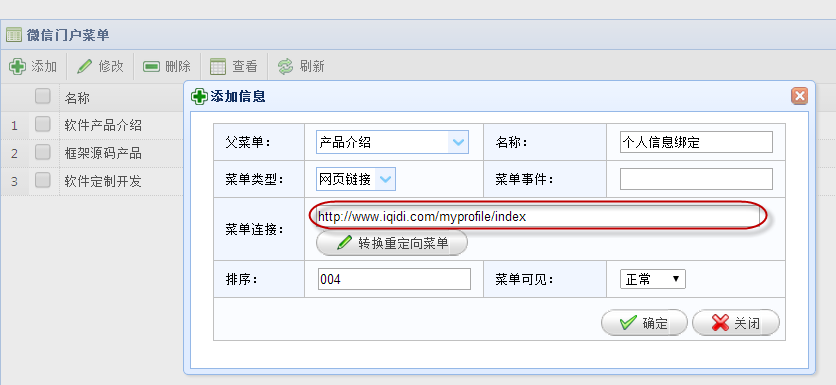
而对于scope=snsapi\_userinfo方式的链接如下：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx3d81fc2886d86526&redirect\_uri=http%3A%2F%2Fwww.iqidi.com%2Ftestwx.ashx&response\_type=code&scope=snsapi\_userinfo&state=123#wechat\_redirect

这两个菜单链接主要就是对我们给定的链接地址进行UrlEncode处理，然后把它赋值给参数redirect\_uri实现的。

由于链接地址比较长，如果每次需要在配置菜单的时候，都复制过来修改，非常不方便，我们可以在自定义菜单的配置界面里面，增加一个按钮功能，对内容进行处理，以便实现我们需要的地址转换，我的门户应用平台对自定义菜单的操作就是基于这个思路实现。

默认我们只需要填写一个需要重定向的url地址就可以了，如下所示。



如果需要配置成重定向的菜单链接地址，那么调用【转换重定向菜单】按钮操作，使用脚本函数进行转换就可以了，转换后的结果如下所示。



原来就是利用后台的javascript实现参数的URL转码，还需要获取后台的AppId，这样才能构造成完整的地址连接。

**2、脚本转换操作的实现代码**

前面说了，第一是需要实现URL转码，第二是获取后台的AppId，然后生成一个完整的URL就可以了。为了避免大家的重复研究，我把这部分代码贴出来一起学习下。

在使用前，我们还需要注意一个问题，就是重定向到指定页面后，这个页面会带有一个code的参数，这个参数非常重要，我们需要获取出来，当然也是通过javascript来获取对应的code参数了。

这个逻辑可以用一个脚本函数来实现，如下所示

[复制代码](javascript:void(0);)

function getUrlVars(){

var vars = [], hash;

var hashes = window.location.href.slice(window.location.href.indexOf('?') + 1).split('&');

for(var i = 0; i < hashes.length; i++)

{

hash = hashes[i].split('=');

vars.push(hash[0]);

vars[hash[0]] = hash[1];

}

return vars;

}

[复制代码](javascript:void(0);)

定义了这个函数后，我们在重定向的页面里面，可以获取code参数的操作如下所示。

var code = getUrlVars()["code"];

先放下这些，我们先来讨论如何把链接地址转换为需要的链接地址操作。

我们为了实现链接地址的互相转换（为了方便），我们可以判断链接地址是否含有qq的域名就可以了。

[复制代码](javascript:void(0);)

if (url.indexOf("https://open.weixin.qq.com/connect/oauth2/authorize?") == 0) {

var redirect\_uri = getUrlVars(url)["redirect\_uri"];

if (redirect\_uri != "") {

var newUrl = decodeURIComponent(redirect\_uri);

$("#" + ctrlName).val(newUrl);

}  
}

[复制代码](javascript:void(0);)

而如果是我们输入的正常链接，那么就应该把它转换为重定向的链接地址，如下所示。

[复制代码](javascript:void(0);)

else {

var newUrl = encodeURIComponent(url);

var reNewUrl = "https://open.weixin.qq.com/connect/oauth2/authorize?appid=@ViewBag.appid&redirect\_uri=" + newUrl + "&response\_type=code&scope=snsapi\_base&state=123#wechat\_redirect";

$("#" + ctrlName).val(reNewUrl);

}

[复制代码](javascript:void(0);)

其中重定向链接需要带有一个当前微信开发用户的appId,这个不是固定的，是不同的开发人员都不一样的东西，这里使用了MVC的动态对象进行绑定：@ViewBag.appid。

在对应的MenuController控制器里面，给它赋值就可以了。

[复制代码](javascript:void(0);)

/// <summary>

/// 默认的视图控制方法

/// </summary>

/// <returns></returns>

public override ActionResult Index()

{

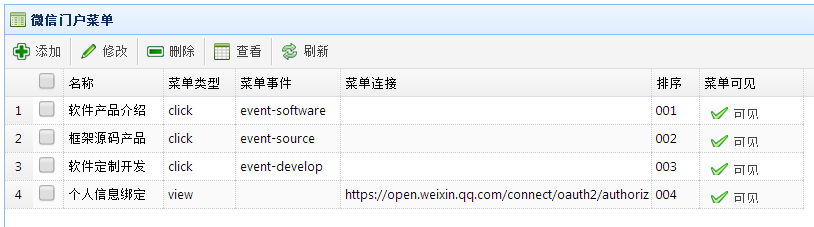
ViewBag.appid = GetAppId();

return View();

}

[复制代码](javascript:void(0);)

这样配置后的重定向菜单地址列表就如下所示了，我们打开对应的记录详细页面，可以通过页面里面的功能按钮，随时对重定向菜单的地址进行转换，方便了解详细的链接内容。

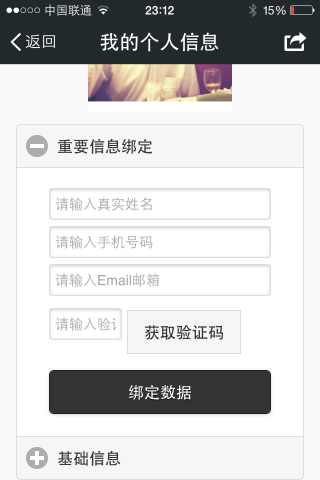


**3、重定向页面的设计及处理**

配置了上面的链接地址后，我们需要在网站里面增加这样的一个页面进行处理用户的信息，一般情况下，我们可能是为了方便用户查看自己的微信基础信息，也为了给用户绑定用户个人数据使用的用途的，如用户可以绑定手机、Email邮箱等操作，还可以绑定和业务系统相关的用户名。这样用户就可以快速注册会员或者和后台的系统进行关联了。

我设计的两个用户信息展示界面如下所示。

这两个界面主要使用了Jquery Mobile的相关内容，对界面进行了处理，整个模块结合了短信验证码的方式，对用户的手机进行验证处理，这样能够更高效的实现信息准确的绑定操作，当然，还可以结合外部系统，绑定用户的账号密码，这样用户可以在微信进入微网站平台进行购物、数据维护、业务管理等操作了，其实一旦绑定外部系统的ID，也就是提供了一个快速进行外部系统的入口了。

具体的内容在下一篇继续介绍了。