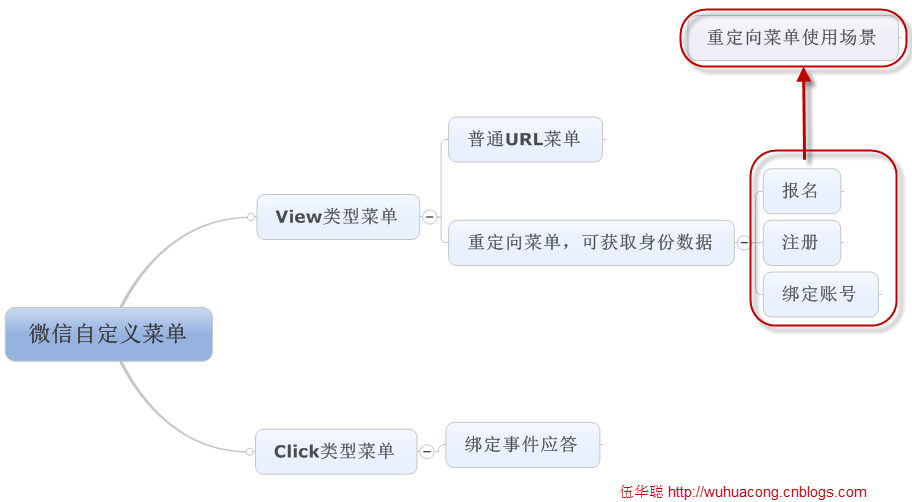
**[C#开发微信门户及应用(11)--微信菜单的多种表现方式介绍](http://www.cnblogs.com/wuhuacong/p/3735739.html)**

在前面一系列文章中，我们可以看到微信自定义菜单的重要性，可以说微信公众号账号中，菜单是用户的第一印象，我们要规划好这些菜单的内容，布局等信息。根据微信菜单的定义，我们可以看到，一般菜单主要分为两种，一种是普通的Url菜单（类型为View的菜单），一种是事件菜单（类型为Click的菜单），一般情况下，微信的Url菜单，是无法获得用户的任何信息的，但微信用户信息非常重要，因此也提供了另外一种方式（类似重定向的方式）来给我们使用，本篇主要介绍这种重新定向的方式菜单的使用，以使我们能够尽可能和用户进行交互。

**1、微信自定义菜单的分类**

微信对自定义菜单的要求：**目前自定义菜单最多包括3个一级菜单，每个一级菜单最多包含5个二级菜单。一级菜单最多4个汉字，二级菜单最多7个汉字，多出来的部分将会以“...”代替。**

根据菜单的分类，我们可以把它通过图形进行分类展示：



我对各种微信公众号进行了解，发现多数账号采用的都是普通的View类型的菜单链接方式，通过它们链接到自己的微网站上，但也有一些做的好的，如省立中山图书馆，就能通过重定向的方式，提供一个绑定图书馆用户和微信OpenID的入口，绑定后，用户就可以查看借阅的书籍，然后可以通过一键续借功能实现图书的快速续借功能。

对于这种重定向类型的Url菜单事件，微信的说明如下：

如果用户在微信中（Web微信除外）访问公众号的第三方网页，公众号开发者可以通过此接口获取当前用户基本信息（包括昵称、性别、城市、国家）。利用用户信息，可以实现体验优化、用户来源统计、帐号绑定、用户身份鉴权等功能。**请注意，“获取用户基本信息接口是在用户和公众号产生消息交互时，才能根据用户OpenID获取用户基本信息，而网页授权的方式获取用户基本信息，则无需消息交互，只是用户进入到公众号的网页，就可弹出请求用户授权的界面，用户授权后，就可获得其基本信息（此过程甚至不需要用户已经关注公众号。）”**

****

**2、重定向类型菜单的URL**

上面说了，重定向类型的菜单分为了两种，其实他们也仅仅是参数Scope类型的不同，其他部分也还是一样的。

为了展示，我们在假设用户单击菜单的时候，切换到http://www.iqidi.com/testwx.ashx这个页面，并带过来当前用户的OpenID等参数信息

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

不过他们给手机客户端的体验是不同的，第一种可以平滑切换，但是第二种会弹出一个对话框供用户确认才能继续。



为了演示上面两种获取数据的不同，我把他们传过来的code的值，用户换取OpenID后进行用户信息的解析，他们两者的结果都是一样了。具体测试界面如下所示。



其中TestWX.ashx的页面后台代码如下所示：

[复制代码](javascript:void(0);)

/// <summary>

/// TestWX 的摘要说明

/// </summary>

public class TestWX : IHttpHandler

{

string appId = ""; //换成你的信息

string appSecret = ""; //换成你的信息

public void ProcessRequest(HttpContext context)

{

context.Response.ContentType = "text/plain";

string content = "";

if (context.Request != null && context.Request.Url != null)

{

NameValueCollection list = HttpUtility.ParseQueryString(context.Request.Url.Query);

foreach (string key in list.AllKeys)

{

content += string.Format("{0}:{1} \r\n", key, list[key]);

}

}

string code = context.Request.QueryString["code"] ?? "";

if (!string.IsNullOrEmpty(code))

{

IBasicApi api = new BasicApi();

try

{

AppConfig config = new AppConfig();

appId = config.AppConfigGet("AppId");//从配置中获取微信程序ID

appSecret = config.AppConfigGet("AppSecret");//从配置中获取微信程序秘钥

AccessTokenResult result = api.GetAccessToken(appId, appSecret, code);

if (result != null)

{

content += string.Format("openid:{0}\r\n", result.openid);

string token = api.GetAccessToken(appId, appSecret);

IUserApi userApi = new UserApi();

UserJson userDetail = userApi.GetUserDetail(token, result.openid);

if (userDetail != null)

{

content += string.Format("nickname:{0} sex:{1}\r\n", userDetail.nickname, userDetail.sex);

content += string.Format("Location:{0} {1} {2} {3}\r\n", userDetail.country, userDetail.province, userDetail.city, userDetail.language);

content += string.Format("HeadUrl:{0} \r\n", userDetail.headimgurl);

content += string.Format("subscribe:{0},{1}\r\n", (userDetail.subscribe == 1) ? "已订阅" : "未订阅", userDetail.subscribe\_time.GetDateTime());

}

}

}

catch { }

}

context.Response.Write(content);

}

[复制代码](javascript:void(0);)

在上面的代码中，我主要分为几步，一个是打印当前用户重定向过来的链接的参数信息，代码如下。

NameValueCollection list = HttpUtility.ParseQueryString(context.Request.Url.Query);

foreach (string key in list.AllKeys)

{

content += string.Format("{0}:{1} \r\n", key, list[key]);

}

然后获取到Code参数后，通过API接口，获取AccessTokenResult的数据，这里面有用户的OpenID

AccessTokenResult result = api.GetAccessToken(appId, appSecret, code);

当正常调用后，我们把用户标识的OpenID进一步进行解析，调用API获取用户的详细信息，具体代码如下所示。

UserJson userDetail = userApi.GetUserDetail(token, result.openid);

当我们把用户的相关信息获取到了，就可以做各种用户信息的展示了，如下代码所示。

[复制代码](javascript:void(0);)

if (userDetail != null)

{

content += string.Format("nickname:{0} sex:{1}\r\n", userDetail.nickname, userDetail.sex);

content += string.Format("Location:{0} {1} {2} {3}\r\n", userDetail.country, userDetail.province, userDetail.city, userDetail.language);

content += string.Format("HeadUrl:{0} \r\n", userDetail.headimgurl);

content += string.Format("subscribe:{0},{1}\r\n", (userDetail.subscribe == 1) ? "已订阅" : "未订阅", userDetail.subscribe\_time.GetDateTime());

}

[复制代码](javascript:void(0);)

**3、重定向链接菜单的用途**

这种菜单就是需要指定域名，在微信后台中进行设置，重定向的链接必须属于这个域名之中，否则不会转到你希望的链接。

这个方式，让我们的微信应用程序后台可以获得用户的标识、用户详细信息等，我们就可以用来绑定和用户相关的业务信息了，如上面提到的图书馆借阅信息，送水客户的信息，客户的积分信息，或者可以和后台账号进行关联实现更加复杂的应用等。用户的身份信息如此重要，如果结合到我们的CRM系统、业务管理系统，就可以发挥用户信息应用的作用了。

以上就是我对这个类型菜单链接的应用了解，具体还需要进一步深化其应用，希望和大家共同探讨这方面的应用场景。