**[C#开发微信门户及应用(12)-使用语音处理](http://www.cnblogs.com/wuhuacong/p/3753834.html)**

我们知道，微信最开始就是做语音聊天而使得其更加流行的，因此语音的识别处理自然也就成为微信交流的一个重要途径，微信的开发接口，也提供了对语音的消息请求处理。本文主要介绍如何利用语音的识别，对C#开发的微信门户应用的整个事件链的处理操作，使得在我们的微信账号里面，更加方便和多元化对用户的输入进行处理。

**1、微信语音接口的定义0**

微信的API这么定义语音的识别的：**开通语音识别功能，用户每次发送语音给公众号时，微信会在推送的语音消息XML数据包中，增加一个Recongnition字段**。

语音的消息格式如下所示。

[复制代码](javascript:void(0);)

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName>

<FromUserName><![CDATA[fromUser]]></FromUserName>

<CreateTime>1357290913</CreateTime>

<MsgType><![CDATA[voice]]></MsgType>

<MediaId><![CDATA[media\_id]]></MediaId>

<Format><![CDATA[Format]]></Format>

<MsgId>1234567890123456</MsgId>

</xml>

[复制代码](javascript:void(0);)

|  |  |
| --- | --- |
| **参数** | **描述** |
| ToUserName | **开发者**微信号 |
| FromUserName | 发送方帐号（一个OpenID） |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 语音为voice |
| MediaId | 语音消息媒体id，可以调用多媒体文件下载接口拉取数据。 |
| Format | 语音格式，如amr，speex等 |
| MsgID | 消息id，64位整型 |

根据以上微信接口的定义，我们可以定义一个实体类来对消息的传递进行处理，如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 接收的语音消息

/// </summary>

[System.Xml.Serialization.XmlRoot(ElementName = "xml")]

public class RequestVoice : BaseMessage

{

public RequestVoice()

{

this.MsgType = RequestMsgType.Voice.ToString().ToLower();

}

/// <summary>

/// 语音格式，如amr，speex等

/// </summary>

public string Format { get; set; }

/// <summary>

/// 语音消息媒体id，可以调用多媒体文件下载接口拉取数据。

/// </summary>

public string MediaId { get; set; }

/// <summary>

/// 消息ID

/// </summary>

public Int64 MsgId { get; set; }

/// <summary>

/// 语音识别结果，UTF8编码

/// </summary>

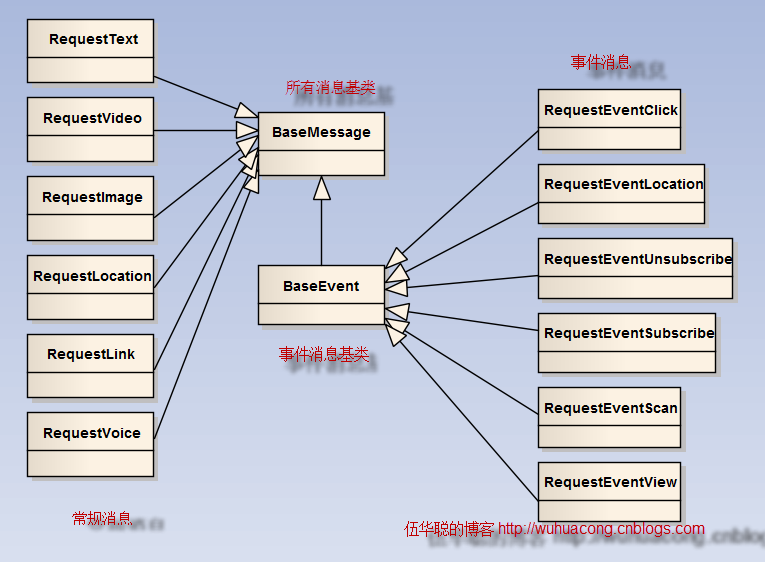
public string Recognition { get; set; }

}

[复制代码](javascript:void(0);)

我们看到，这里我们最感兴趣的是语音的识别结果，也就是Recognition的字段，这个就是微信服务器自动根据用户的语音转换过来的内容，我测试过，识别率还是非常高的。

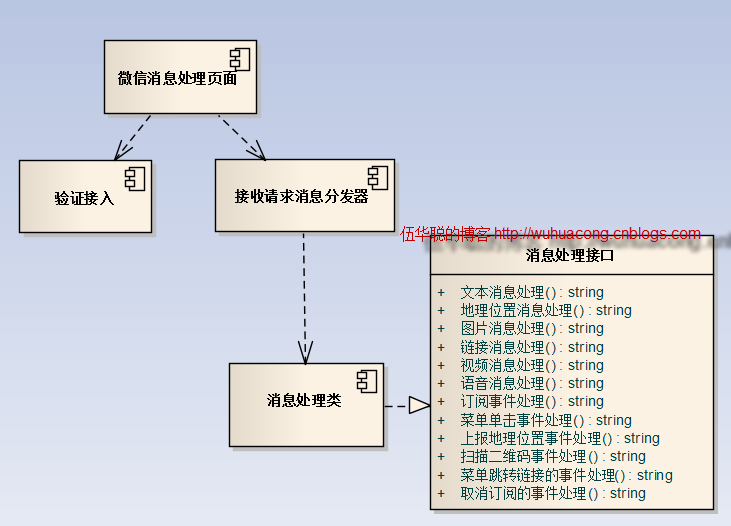
这个实体类，在整个微信应用的消息传递中的关系如下所示：



**2、语音的处理操作**

明确了上面的语音对象实体，我们就可以看看它们之间是如何处理的。

微信消息的处理逻辑如下图所示。



其中我们来看看语音的处理操作，我的代码处理逻辑如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 对语音请求信息进行处理

/// </summary>

/// <param name="info">语音请求信息实体</param>

/// <returns></returns>

public string HandleVoice(Entity.RequestVoice info)

{

string xml = "";

// 开通语音识别功能，用户每次发送语音给公众号时，

// 微信会在推送的语音消息XML数据包中，增加一个Recongnition字段。

if (!string.IsNullOrEmpty(info.Recognition))

{

TextDispatch dispatch = new TextDispatch();

xml = dispatch.HandleVoiceText(info, info.Recognition);

}

else

{

xml = "";

}

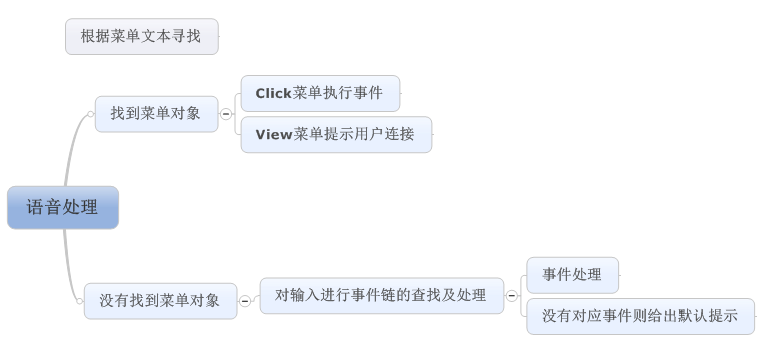
return xml;

}

[复制代码](javascript:void(0);)

在这里，我先看看，是否获得了微信的语音识别结果，如果获得，那么这个时候，就是和处理用户文本输入的操作差不多了，因此把它转给TextDispatch的处理类进行处理。

其中这里面的处理逻辑如下所示。



首先我根据识别结果，寻找是否用户读出了微信门户的菜单名称，如果根据语音结果找到对应的菜单记录，那么我们执行菜单事件（如果是URL的View类型菜单，我们没办法重定向到指定的链接，因此给出一个链接文本提示，给用户单击进入；如果没有找到菜单记录，那么我们就把语音识别结果作为一般的事件进行处理，如果事件逻辑没有处理，那么我们最后给出一个默认的语音应答提示结果就可以了。

具体的处理代码如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 如果用户用语音读出菜单的内容，那么我们应该先根据菜单对应的事件触发，最后再交给普通事件处理

/// </summary>

/// <param name="info"></param>

/// <returns></returns>

public string HandleVoiceText(BaseMessage info, string voiceText)

{

string xml = "";

MenuInfo menuInfo = BLLFactory<Menu>.Instance.FindByName(voiceText);

if (menuInfo != null)

{

#region 如果找到菜单对象的处理

if (menuInfo.Type == "click")

{

//模拟单击事件

RequestEventClick eventInfo = new RequestEventClick();

eventInfo.CreateTime = info.CreateTime;

eventInfo.EventKey = menuInfo.Key;

eventInfo.FromUserName = info.FromUserName;

eventInfo.ToUserName = info.ToUserName;

xml = base.DealEvent(eventInfo, eventInfo.EventKey);

}

else

{

//由于无法自动切换到连接，

//转换为连接文本供用户进入

string content = string.Format("请单击链接进入<a href=\"{0}\">{1}</a> ", menuInfo.Url, menuInfo.Name);

ResponseText textInfo = new ResponseText(info);

textInfo.Content = content;

xml = textInfo.ToXml();

}

#endregion

}

else

{

//交给事件机制处理

if (string.IsNullOrEmpty(xml))

{

xml = HandleText(info, voiceText);

}

}

//最后如果没有处理到，那么提示用户的语音内容

if (string.IsNullOrEmpty(xml))

{

ResponseText textInfo = new ResponseText(info);

textInfo.Content = string.Format("非常抱歉，您输入的语音内容没有找到对应的处理方式。您的语音内容为：{0}", voiceText);

xml = textInfo.ToXml();

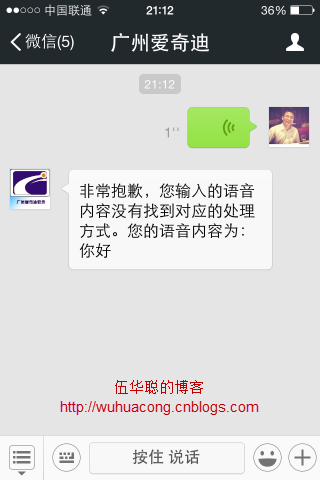
}

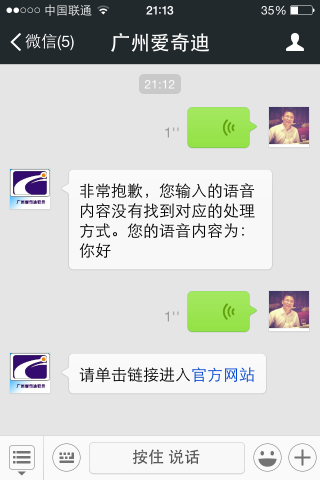
return xml;

}

[复制代码](javascript:void(0);)

微信门户测试界面效果如下所示。



为了方便对客户会话的记录，我的微信门户后台，会记录用户的语音输入内容，如下所示。



 当然，微信后台的管理界面，也能够查到相应的语音记录，界面如下所示。



以上就是我对微信语音的消息定义和事件处理的逻辑，其实语音是一个重要的输入，如果正确的识别内容，比手工输入的效果更好，给用户提供另外一种高效的输入和事件处理操作。

这样的处理模式，能够使得我们整个微信门户框架，不管是对于用户的语音输入，还是文本输入，还是菜单事件的处理，都可以融为一体，实现更加完美的衔接。