

V1.0

Windows 8 平台新浪微博 SDK 使用说明

新浪
微博开放平台

目录

封 面.....	0
一、 概述.....	2
二、 授权模式.....	3
三、 SDK 使用说明	4
1. 环境安装.....	4
2. 引入 SDK 所需文件	4
四、 接口说明.....	5
1. OAuth2.0 授权	5
2. 常用数据请求.....	5
3. 其他数据请求.....	6
五、 SDK 实例使用说明	8
1. 设置 SDK 参数	8
2. OAuth 授权	8
3. 常用微博请求.....	9
4. 其他数据请求.....	10
六、 返回数据处理及错误处理.....	12
1. 错误类型.....	12
2. 处理返回数据示例代码.....	13
七、 问题反馈.....	14

一、概述

新浪微博 Windows 8 平台官方 SDK 包提供给第三方开发者 OAuth 授权登录及基本 Open API 的请求，大大简化了第三方集成微博相关服务的难度。本 SDK 采用 Windows Runtime Component 形式发布，支持 Windows Store App 应用程序开发，可以使用 C#/VB/C++/JavaScript&Html5 语言调用本 SDK 提供的相应功能。集成的 OPEN API 版本为 V2 版，以后会随着 OPEN API 的版本升级更新的微博 SDK 包。

二、 授权模式

OAUTH 授权：OAuth 是一种常见的授权模式，用户可以更安全放心的使用用户身份，而不必把用户名和密码透露给第三方。而 OAuth2.0 是从 2006 年开始设计 OAuth 协议的下一个版本，OAuth2.0 同时提供 Web，桌面和移动应用程序的支持，并较 1.0 相比整个授权验证流程更简单更安全。也是新浪微博开放平台未来最主要的用户身份验证和授权方式。我们推荐用户使用 OAuth2.0 鉴权方式，本微博 SDK 对 OAuth2.0 中的授权页方式进行了封装，您只需简单的调用即可获得访问微博接口的授权密钥。

三、 SDK 使用说明

1. 环境安装

安装 Windows 8 Store App 相关开发环境。

2. 引入 SDK 所需文件

- **C#/VB、JavaScript&Html5 项目**

选中项目，右键 References->Add Reference->Browse...，在打开文件选择窗口找到 Release 目录下的 RestSharp.winmd 和 WeiboSDKForWinRT.winmd 文件，点击 Add 将它们添加到项目引用中去。

- **C++/CX 项目**

选中项目，右键项目属性->Common Properties->Framework and References，点击 Add new Reference，在打开文件选择窗口找到 Release 目录下的 RestSharp.winmd 和 WeiboSDKForWinRT.winmd 文件，点击 Add 将它们添加到项目引用中去。

四、接口说明

接口使用前需进行参数设置,详见“设置 SDK 所需参数”

1. OAuth2.0 授权

接口	
OAuthClient.BeginAuth()	
参数名称	说明
无	
1) 需向微博开发者平台申请客户端鉴权的权限方能使用。 2) 为获取数据返回信息, 需注册 ClientOAuth.LoginCallback	

LoginCallback 返回参数	
OAuth2LoginBack(bool isSuccess, SdkAuthError err, SdkAuth2Res response)	
参数名称	说明
isSuccess	授权是否成功
err	授权错误信息
response	授权成功的数据信息

2. 常用数据请求

接口	
SdkNetEngine.RequestCmd(SdkRequestType type, SdkCmdBase data)	
参数名称	说明

type	请求数据枚举类型
data	参数类的对象

1) 该接口对一些常用微博数据的请求做出封装,用户只需传入类型+赋值好参数的类即可使用,而不用关心具体 HTTP 的 API 调用和组包细节。

2) 该接口支持的功能有:

SdkRequestType.TIMELINES - 批量获取当前登录用户及其所关注用户的最新微博消息 “

参数类: `cmdNormalMessages`

SdkRequestType.USER_TIMELINE - 批量获取某人的微博

参数类: `cmdUserTimeline`

SdkRequestType.POST_MESSAGE - 发送微博

参数类: `cmdUploadMessage`

SdkRequestType.POST_MESSAGE_PIC - 发送带图片的微博

参数类: `cmdUploadPic`

SdkRequestType.FRIENDSHIP_CREATE - 关注某用户

SdkRequestType.FRIENDSHIP_DESTROY - 取消关注

SdkRequestType.FRIENDSHIP_SHOW - 获取两者关系

参数类: `cmdFriendShip`

SdkRequestType.AT_USERS - @用户时的联想建议

参数类: `cmdAtUsers`

SdkRequestType.SHORTEN_URL - 短链请求

参数类: `cmdShortenUrl`

3. 其他数据请求

接口
<code>SdkNetEngine. SendRequest(SdkRequestType type, SdkCmdBase data)</code>

参数名称	说明
type	请求数据枚举类型
data	参数类的对象
<p>本 SDK 封装的微博接口请求调用已经能够基本满足大部分开发者对于微博接入的需求。但也有可能由于具体需求的原因需要调用本 SDK 没有封装的接口，这就需要开发者对该函数自行传入接口地址和自行对 RestRequest 组参，具体使用方法见“SDK 实例使用说明”中对其介绍。</p>	

五、 SDK 实例使用说明

1. 设置 SDK 参数

SDK 使用及修改必须遵守微博开发者协议，以及相关开源软件协议。

使用 SDK 的各功能前，需要先设置 SDK 所需的常量。其中 AppKey 和 AppSecret 需要 <http://open.weibo.com> 申请, RedirectUri 为应用地址，在我的应用中可以编辑。

```
// TODO:编译运行之前需要开放平台参数.  
SdkData.AppKey = "111111";  
SdkData.AppSecret = "1111111111";  
SdkData.RedirectUri = "http://www.weibo.com";
```

注意：WeiboSdkSample 里面的示例项目也需要配置这些信息才能正确使用。

2. OAuth 授权

开发者只需创建一个 OAuthClient 的实例对象，并调用 BeginOAuth 方法即可调出授权界面。如果需要回去授权信息还需注册 OAuthClient 对象的 LoginCallback 回调函数获取授权信息或者错误信息。并针对返回的信息做相应的处理。

- 界面效果：

← 正在连接到服务

 新浪微博
weibo.com

注册

授权 Alexis 访问你的微博帐号

登录

- 示例代码:

```
var oauthClient = new WeiboSDKForWinRT.ClientOAuth();
// 判断是否已经授权或者授权是否过期.
if (oauthClient.IsAuthorized == false)
{
    oauthClient.LoginCallback += (isSuccess, err, response) =>
    {
        if (isSuccess)
        {
            // TODO: deal the OAuth result.
            this.statusRun.Text = "Congratulations, Authorized successfully!";
            this.ResultRun.Text = string.Format("AccesssToken:{0}, ExpriesIn:{1}, Uid:{2}",
                response.AccessToken, response.ExpiresIn, response.Uid);
        }
        else
        {
            // TODO: handle the err.
        }
    };
    oauthClient.BeginAuth();
}
```

3. 常用微博请求

使用SdkCmdBase的子类作为参数包传入接口所需参数。

示例代码：

```
var engine = new WeiboSDKForWinRT.SdkNetEngine();
WeiboSDKForWinRT.SdkCmdBase cmdBase = new WeiboSDKForWinRT.cmdTimelines()
{
    Count = "20",
    Page = "1"
};
var result = await engine.RequestCmd(WeiboSDKForWinRT.SdkRequestType.FRIENDS_TIMELINE, cmdBase);

if (result.errCode == WeiboSDKForWinRT.SdkErrCode.SUCCESS)
{
    this.ResultRun.Text = result.content;
    this.statusRun.Text = "Fetch friend's Timeline succeeded!";
}
else
{
    // TODO: deal the error.
    this.statusRun.Text = "Fetch friend's Timeline failed!";
}
```

4. 其他数据请求

对于现有封装接口不能满足特定需求的用户，对此我们深感抱歉。本 SDK 的目的为了满足大部分第三方案接入微博，所以就不可能将微博开放平台的所有接口都封装和实现。不过开发者仍可以通过如下步骤自行添加对所需接口的支持：

1. 根据微博开放平台提供的接口文档添加所需接口的参数类，继承

CustomCmdBase。

2. 实现 ConvertToRequestParam 方法，参考如下代码（关注列表请求参数类）：

3. 测试新添加的接口实现，直至测试通过。

4. 类似常用微博请求调用 SdkNetEngine 的 SendRequest 方法获取数据。注意此时 SdkRequestType 枚举值应为 OtherAPI。

```
public class cmdFriendList : CustomCmdBase
{
    private string _uid = string.Empty;
    public string Uid[...]

    private string _cursor = string.Empty;
    public string Cursor[...]

    private string _trimStatus = string.Empty;
    public string TrimStatus[...]

    private string count = string.Empty;
    public string Count[...]

    public void ConvertToRequestParam(RestRequest request)
    {
        request.Resource = "/friendships/friends.json";
        request.Method = Method.GET;
        if (Uid.Length > 0)
        {
            request.AddParameter("uid", Uid);
        }
        if (Cursor.Length > 0)
        {
            request.AddParameter("cursor", Cursor);
        }
        if (Count.Length > 0)
        {
            request.AddParameter("count", Count);
        }
        if (TrimStatus.Length > 0)
        {
            request.AddParameter("trim_status", TrimStatus);
        }
    }
}
```

六、返回数据处理及错误处理

1. 错误类型

```
/// <summary>
/// 微博SDK自定义的ErrCode.
/// </summary>
public enum SdkErrCode
{
    /// <summary>
    /// 参数错误.
    /// </summary>
    XPARAM_ERR = -1,
    /// <summary>
    /// 成功.
    /// </summary>
    SUCCESS = 0,
    /// <summary>
    /// 网络不可用.
    /// </summary>
    NET_UNUSUAL,
    /// <summary>
    /// 服务器返回异常.
    /// </summary>
    SERVER_ERR,
    /// <summary>
    /// 访问超时.
    /// </summary>
    TIMEOUT,
    /// <summary>
    /// 用户请求被取消.
    /// </summary>
    USER_CANCEL
}
```

如果 errCode 为 `SdkErrCode.SERVER_ERR` 时，访问服务器时发送错误。此时错误码会保存在 `specificCode` 字段。错误码的详细意义请参见：

<http://open.weibo.com/wiki/Help/error>

2. 处理返回数据示例代码

```
var engine = new WeiboSDKForWinRT.SdkNetEngine();
WeiboSDKForWinRT.SdkCmdBase cmdBase = new WeiboSDKForWinRT.cmdTimelines()
{
    Count = "20",
    Page = "1"
};
var result = await engine.RequestCmd(WeiboSDKForWinRT.SdkRequestType.FRIENDS_TIMELINE, cmdBase);
switch (result.errCode)
{
    case WeiboSDKForWinRT.SdkErrCode.NET_UNUSUAL:
        // 网络状况不好，检查网络情况。
        break;
    case WeiboSDKForWinRT.SdkErrCode.SERVER_ERR:
        // 服务器返回错误，可以通过result.specificCode获得错误
        // 根据错误码查到错误信息。
        break;
    case WeiboSDKForWinRT.SdkErrCode.XPARAM_ERR:
        // 参数传递错误，检查传入的参数。
        break;
    case WeiboSDKForWinRT.SdkErrCode.SUCCESS:
        // 请求正常，数据正常返回。根据需求处理数据。
        break;
    default:
        break;
}
```

七、 问题反馈

如果您在使用此 SDK 上有问题，请先参考我们提供的调用范例 WeiboSdkSample 工程代码，并且 WeiboSDK 工程本身我们是开源的，这样方便大家了解这个 SDK 的工作原理。我们不推荐您修改 WeiboSDK.WinRT 工程中的代码，因为随着 SDK 开发包的升级维护，我们会对这个工程代码进行同步升级。如果有问题可通过以下联系方式联系我们：

- 微博：<http://e.weibo.com/u/2776760443>
- 邮箱：weibo_api_tech@sina.com
- 开放平台问答系统：<http://open.weibo.com/qa>