1. **概述**

**随着电商的发展，商家已经开始向多个平台发展。多个平台与商家自身管理系统（ERP）的数据对接就显得尤为重要。**

**AlibabaSDK 提供与阿里开放平台对接的Api，为商家在1688上的数据（商品，订单等）的对接、管理等提供解决方案。**

1. **阿里开放平台URL**

**开放平台的URL分为两种体系**

* 1. **预授权体系**

**此体系下的URL构成如下：**

${RequestSchema}://auth.1688.com/{apiRoot}/{protocol}/{apiVersion}/{apiNamespace}/{apiName}/{clientId}

取值示例如下(以获取accesstoken为例)，当然具体要依据不同的功用与阿里给出的文档对应。

${RequestSchema}的取值为 “http” 或 “https” ( Com.Alibaba.UriSchema )

{apiRoot}的取值为 “openapi”

{protocol}取值为 “http” (此处可取值为 枚举类型Com.Alibaba.Protocol 的值)

{apiVersion}的取值为1

{apiNamespace}取值为 “system.oauth2”

{apiName}取值为 “getToken”

{clientId}为对应店铺的授权信息 appKey

以post 方法提交 ，并且有 参数 如下：

var data={

grant\_type=”authorization\_code”,

need\_refresh\_token=”true”,

client\_id=${appKey}, //此处为应店铺的授权信息 appKey

client\_secret=${appSecret},//此处为应店铺的授权信息 appSecret

redirect\_uri=${redicectUri},////此处为接收回调的地址

code=${pre-auth-code}//此处为已经得到的预授权码

}

* 1. **实际调用体系**

**此体系下的URL构成如下：**

${RequestSchema}://gw.open.1688.com/{requestPolicy.ApiRoot}/{requestPolicy.Protocol}/{requestPolicy.ApiVersion}/{requestPolicy.ApiNamespace}/{requestPolicy.ApiName}/{requestPolicy.ClientId}

取值示例如下(以获取订单(卖家视图) 为例)，当然具体要依据不同的功用与阿里给出的文档对应。

${RequestSchema}的取值为 “http” 或 “https” ( Com.Alibaba.UriSchema )

{requestPolicy.ApiRoot}的取值为 “openapi”

{requestPolicy.Protocol}取值为 “param2” (此处可取值为 枚举类型Com.Alibaba.Protocol 的值)

{requestPolicy.ApiVersion}的取值为1

{requestPolicy.ApiNamespace}取值为 “com.alibaba.trade”

{requestPolicy.ApiName}取值为 “alibaba.trade.get.sellerView”

{requestPolicy.ClientId}为对应店铺的授权信息 appKey

以post 方法提交 ，并且有 参数 如下：

var data={

website=”1688”,

orderId=${orderId},

includeFields=” GuaranteesTerms,NativeLogistics,RateDetail,OrderInvoice”,//参看对应的阿里文档

access\_token=${access\_token},

\_aop\_signature=${\_aop\_signature }//此处通过url算出的签名

}

1. **API授权说明**
   1. **授权方式**

**目前开放平台提供了三种授权方式：单用户授权，简单多用户授权，OAuth授权，具体内容如下：**

* + 1. **单用户授权**

**单用户授权意味着您的应用程序只能被一个阿里巴巴用户授权，适用于企业对接场景，企业只需要在1688上使用一个账号进行商品采购或者商品售卖，授权流程可以简单归纳为：**

**（1）对接应用审核通过后开发者可以在控制台添加授权用户，默认为开发者自己，如果授权用户有登录邮箱，系统会自动发送邀请邮件，开发者账号除外；**

**（2）授权用户拿到邮件中或者开发者提供的授权确认链接，然后访问此链接确认授权；**

**（3）开发者可以参考 API调用说明了解如何调用API，注意不需要传access\_token，因为这样调用api操作的就是唯一授权用户的数据。**

* + 1. **多用户授权**

**多用户授权适用于企业对接场景，并且企业在1688有多个店铺或者买家账号，授权流程可以简单归纳为：**

**（1）开发者在控制台添加授权用户，默认为开发者自己，如果授权用户有登录邮箱，系统会自动发送邀请邮件，开发者账号除外；**

**（2）授权用户拿到邮件中或者开发者提供的授权确认链接，然后访问此链接确认授权；**

**（3）开发者在控制台拿到持久的访问令牌（accessToken）；**

**（4）开发者可以参考 API调用说明了解如何调用API，注意调用API时传入此accessToken，这样操作的是此token对应授权用户的数据。**

* + 1. **OAuth授权**

**阿里巴巴开放平台采用OAuth 2.0作为授权协议，授权流程可以简单归纳为：**

**（1）获取临时令牌；**

**（2）用临时令牌换取长时令牌以及访问令牌；**

**（3）访问令牌过期后用长时令牌刷新访问令牌。**

* + 1. **本方案采用 “多用户授权”方式**

1. **Com.Alibaba.Open**

**1.获得授权URL**

var clientId = "your app key";//

var authUrl = Com.Alibaba.Open.OAuthAPIs.OAuthApi.GetAuthorizeUrl(clientId, state: "your state");

**state 参数中的值 1688会原样 返回，此值可以与表中某字段对应，以作过滤之用**

**2.获得token**

**var client\_id = "your app key";**

**var client\_secret = "your app secret";**

**var redirect\_uri = @"http://localhost:12305";**

**var pre\_auth\_code = "\*\*\*\*-\*\*\*\*-\*\*\*\*-\*\*\*\*";**

**var accessToken = Com.Alibaba.Open.OAuthAPIs.OAuthApi.GetAccessToken(**

**clientId: client\_id,**

**clientSecret: client\_secret,**

**redirectUrl: redirect\_uri,**

**preAuthCode: pre\_auth\_code,**

**sessionType: SessionType.Prod);**

**3.刷新 token**

**var client\_id = "your app key";**

**var client\_secret = "your app secret";**

**var refresh\_token = "\*\*\*\*-\*\*\*\*\*\*-\*\*";**

**var accessToken = Com.Alibaba.Open.OAuthAPIs.OAuthApi.RefreshAccessToken(**

**clientId: client\_id,**

**clientSecret: client\_secret,**

**refreshToken: refresh\_token,**

**sessionType: SessionType.Prod,**

**requestSchema: UriSchema.https);**

1. **Com.Alibaba.Trade**

**1.获得订单列表（卖家视图）**

**var requestPolicy = new Com.Alibaba.Entities.Request.RequestPolicy**

**{**

**ClientId = "your app key",//**

**ClientSecret = "your app secret",//**

**AccessToken = "\*\*\*\*-\*\*\*-\*\*\*-\*\*\*\*",**

**ApiRoot = "openapi",**

**ApiVersion = "1",**

**ApiNamespace = "com.alibaba.trade",**

**ApiName = "alibaba.trade.getSellerOrderList",**

**Protocol = Com.Alibaba.Protocol.param2,**

**SessionType = SessionType.Prod,**

**RequestSchema = UriSchema.https**

**};**

**Com.Alibaba.Trade.Containers.AccessTokenContainer.Register(requestPolicy.ClientId, requestPolicy.ClientSecret, requestPolicy.AccessToken, 36000, 36000, "refresh-token", "ret\_url");**

**var reqModel = new Com.Alibaba.Trade.Entities.Request.ReqModelOrderListForSeller();**

**var result = Com.Alibaba.Trade.TradeAPIs.OrderApi.OrderListOnSellerViewRetrieveRowString(**

**requestPolicy,**

**reqModel);**

**2.获得订单详情(卖家视图)**

**var requestPolicy = new Com.Alibaba.Entities.Request.RequestPolicy**

**{**

**ClientId = "your app key",//**

**ClientSecret = "your app secret",//**

**AccessToken = "\*\*\*\*-\*\*\*\*",**

**ApiRoot = "openapi",**

**ApiVersion = "1",**

**ApiNamespace = "com.alibaba.trade",**

**ApiName = "alibaba.trade.get.sellerView",**

**Protocol = Com.Alibaba.Protocol.param2,**

**SessionType = SessionType.Prod,**

**RequestSchema = UriSchema.https**

**};**

**Com.Alibaba.Trade.Containers.AccessTokenContainer.Register(**

**requestPolicy.ClientId,**

**requestPolicy.ClientSecret,**

**requestPolicy.AccessToken,**

**36000,**

**36000,**

**"\*\*\*-\*\*\*",**

**"ret-url");**

**var reqModel = new Com.Alibaba.Trade.Entities.Request.ReqModelOrderDetailForSeller();**

**reqModel.orderId = your order id;**

**var result = Com.Alibaba.Trade.TradeAPIs.OrderApi.OrderDetailOnSellerViewRetrieveRowString(**

**requestPolicy,**

**reqModel);**