.NET C#支付宝条码支付接口详解

支付宝条码支付接口使用，供大家参考，具体内容如下

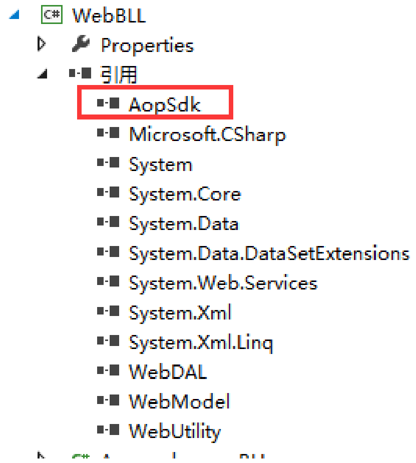
应用场景实例

收银员使用扫码设备读取用户支付宝钱包“付款码”后，将二维码或条码信息通过本接口上送至支付宝发起支付。

SDK下载

支付宝提供3种开发语言的SDK，选择自己的开发语言下载，项目中会有很多示例。本文选择.NET2010版本。

将SDK项目中的AopSdk.dll文件引用到自己的项目中。



```
WebBLL
  Properties
  引用
    AopSdk
    Microsoft.CSharp
    System
    System.Core
    System.Data
    System.Data.DataSetExtensions
    System.Web.Services
    System.Xml
    System.Xml.Linq
    WebDAL
    WebModel
    WebUtility
```

支付类代码

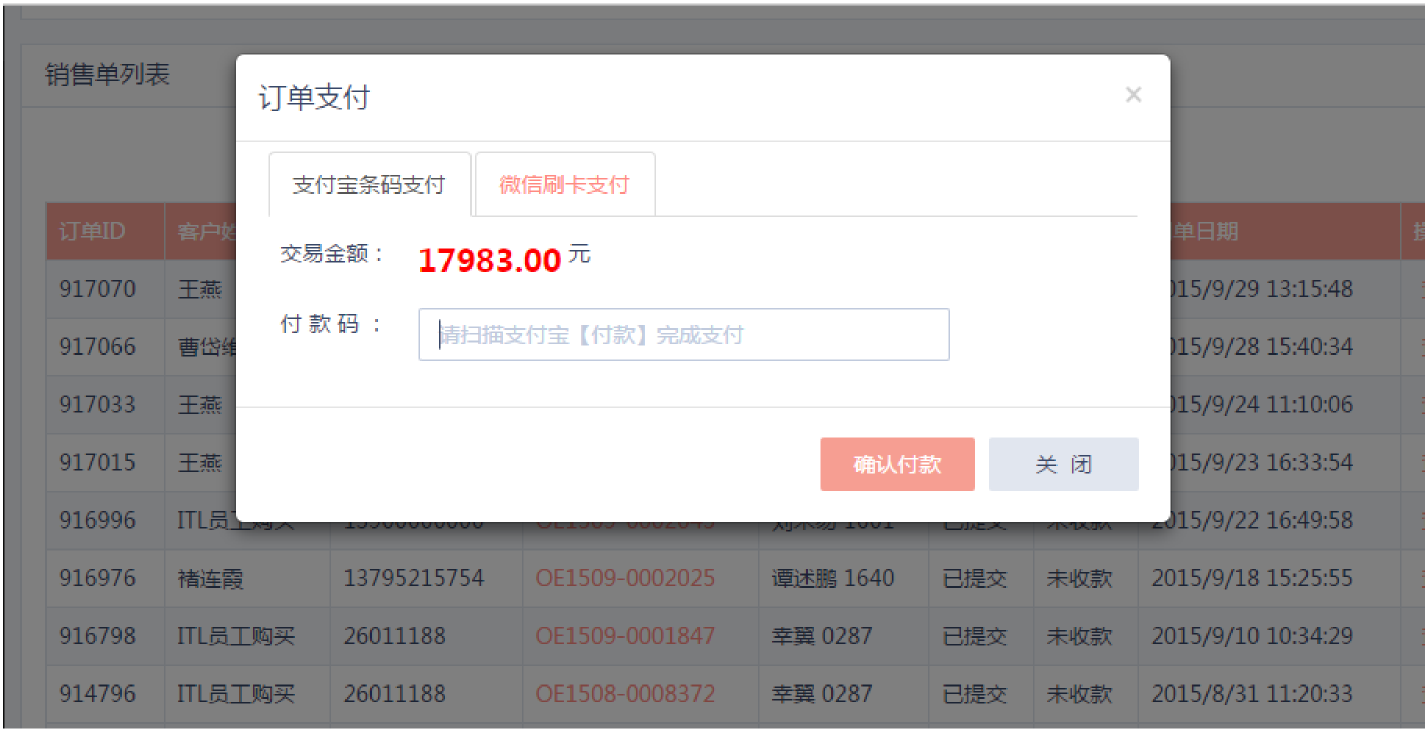
简略版 数据需自行获取

```
public class ToAlipayBLL
{
private static readonly ToAlipayDAL dal = new ToAlipayDAL();

static IAopClient client = null;

public static bool CheckValidationResult(object sender, X509Certificate certificate, X509Chain chain, SslPolicyErrors errors)
{
//直接确认，否则打不开
return true;
}

/// <summary>
/// 支付宝条码支付
/// </summary>
/// <param name="osaleh_osalehID"></param>
/// <param name="txtPaymentCode">付款码</param>
/// <param name="orderType"></param>
/// <param name="user"></param>
/// <returns></returns>
public static ResponseDTO GetAlipayRequestExecute(string osaleh_osalehID, string str_osaled_osaledID, string
txtPaymentCode, string orderType, UsersEntity user)
{
ResponseDTO resp = new ResponseDTO();
try
{

//请根据实际请求需要biz_content参数
#region biz_content参数

OrderListModel orderList = new OrderListModel();
List<ProductDetailModel> goodsList = new List<ProductDetailModel>();
StringBuilder param = new StringBuilder();

//商户订单号
string out_trade_no = System.DateTime.Now.ToString("yyyyMMddHHmmss") + "0000" + osaleh_osalehID; //商户唯一订单号
orderList.out_trade_no = out_trade_no;
```

```
//支付场景
orderList.scene = "bar_code";
//支付授权码
orderList.auth_code = txtPaymentCode;
//卖家支付宝用户ID
orderList.seller_id = "";
//订单总金额
//orderList.total_amount = osaled_amount.ToString("#0.00");

//商品明细
string goodsName = string.Empty;
bool IsFlag = true;

foreach (var item in OsaledList)
{

ProductDetailModel detailModel = new ProductDetailModel();
detailModel.goods_id = "0";
detailModel.goods_name = "default";
detailModel.quantity = ((int)item.osaled_qty).ToString();
detailModel.price = item.osaled_amount.ToString("#0.00");
detailModel.goods_category = "";
goodsList.Add(detailModel);
}
orderList.goods_detail = goodsList;

//订单标题
orderList.subject = goodsName;
//订单描述
orderList.body = "";
//商户操作员编号
orderList.operator_id = user.user_employeeNo;
//商户门店编号
orderList.store_id = "";
//支付宝店铺编号
orderList.alipay_store_id = "";
//机具终端编号
orderList.terminal_id = "";
//支付超时时间
string expire_time = System.DateTime.Now.AddMinutes(30).ToString("yyyy-MM-dd HH:mm:ss");
orderList.time_expire = expire_time;

#endregion biz_content参数

string biz_content = FormatToJson.Serialize(orderList);

StoreEntity store = StoreBLL.GetStoreEntityByStore_Code(user.user_company);
//开发者的AppId
string alipay_appId = "";
ConfigHelper.alipay_merchant_private_key = string.Format(ConfigHelper.alipay_merchant_private_key, store.Area_Code);
ConfigHelper.alipay_public_key = string.Format(ConfigHelper.alipay_public_key, store.Area_Code);

ServicePointManager.ServerCertificateValidationCallback = new
System.Net.Security.RemoteCertificateValidationCallback(CheckValidationResult);

client = new DefaultAopClient(ConfigHelper.alipay_serverUrl, alipay_appId, ConfigHelper.alipay_merchant_private_key, "json",
ConfigHelper.alipay_version,
ConfigHelper.alipay_sign_type, ConfigHelper.alipay_public_key, ConfigHelper.alipay_charset);

AlipayTradePayResponse payResponse = Pay(biz_content);

string result = payResponse.Body;

if (payResponse != null)
{

switch (payResponse.Code)
{
case ToAlipayResultCode.SUCCESS://支付成功
//业务处理
resp.Message = "支付成功";
resp.Status = true;
break;
case ToAlipayResultCode.INRROCESS://业务处理中

StringBuilder sb1 = new StringBuilder();
sb1.Append("{\"out_trade_no\":\"" + out_trade_no + "\"}");

//调用查询接口需要进行轮询订单支付结果
AlipayTradeQueryResponse queryResponse = LoopQuery(sb1.ToString()); //用订单号trade_no进行轮询也是可以的。
if (queryResponse != null)
{
if (queryResponse.Code == ToAlipayResultCode.SUCCESS)
{
//支付成功或交易结束
if (queryResponse.TradeStatus == "TRADE_SUCCESS" || queryResponse.TradeStatus == "TRADE_FINISHED")
{
//业务处理

resp.Message = "支付成功";
resp.Status = true;
}
else
{
//支付超时
resp.Message = "支付超时";
resp.Status = false;
}
}
else
{
//支付失败
resp.Message = "支付失败，错误代码：" + queryResponse.SubCode + "。描述：" + queryResponse.SubMsg;
resp.Status = false;
}
}

break;
case ToAlipayResultCode.FAIL:
StringBuilder sb2 = new StringBuilder();
sb2.Append("{\"out_trade_no\":\"" + out_trade_no + "\"}");
AlipayTradeCancelResponse cancelResponse = Cancel(sb2.ToString());

if (!string.IsNullOrEmpty(cancelResponse.SubCode))
{
resp.Message = "支付失败，错误代码：" + cancelResponse.SubCode + "。描述：" + cancelResponse.SubMsg;
}
else
{
resp.Message = "支付失败，错误代码：" + payResponse.SubCode + "。描述：" + payResponse.SubMsg;
}
//支付失败
resp.Status = false;

break;
}
}
}
else
{
resp.Message = "操作失败，未查询到订单信息，请联系管理员！";
resp.Status = false;
}
}
catch (Exception ex)
{
ExceptionLog.ToAlipayLog(ex.Message);//记录日志
resp.Message = ex.Message;
resp.Status = false;
}
return resp;
}
```

```
private static AlipayTradePayResponse Pay(string biz_content)
{
AlipayTradePayRequest payRequst = new AlipayTradePayRequest();
payRequst.BizContent = biz_content;

AlipayTradePayResponse payResponse = client.Execute(payRequst);
return payResponse;
}

private static AlipayTradeCancelResponse Cancel(string biz_content)
{
AlipayTradeCancelRequest cancelRequest = new AlipayTradeCancelRequest();
cancelRequest.BizContent = biz_content;
AlipayTradeCancelResponse cancelResponse = client.Execute(cancelRequest);

if (null != cancelResponse)
{
if (cancelResponse.Code == ToAlipayResultCode.FAIL && cancelResponse.RetryFlag == "Y")
{
// 新开一个线程重试撤销
ParameterizedThreadStart ParStart = new ParameterizedThreadStart(cancelOrderRetry);
Thread myThread = new Thread(ParStart);
object o = biz_content;
myThread.Start(o);
}
}

return cancelResponse;

}

private static void cancelOrderRetry(object o)
{
int retryCount = 10;

for (int i = 0; i < retryCount; ++i)
{

Thread.Sleep(5000);
AlipayTradeCancelRequest cancelRequest = new AlipayTradeCancelRequest();
cancelRequest.BizContent = o.ToString();

AlipayTradeCancelResponse cancelResponse = client.Execute(cancelRequest);

if (null != cancelResponse)
{
if (cancelResponse.Code == ToAlipayResultCode.FAIL)
{
if (cancelResponse.RetryFlag == "N")
{
break;
}
}
if ((cancelResponse.Code == ToAlipayResultCode.SUCCESS))
{
break;
}
}

}
}

private static AlipayTradeQueryResponse LoopQuery(string biz_content)
{

AlipayTradeQueryRequest payRequst = new AlipayTradeQueryRequest();
payRequst.BizContent = biz_content;

Dictionary<string, string> paramsDict = (Dictionary<string, string>)payRequst.GetParameters();
AlipayTradeQueryResponse payResponse = null;

for (int i = 1; i <= 6; i++)
{
Thread.Sleep(5000);

payResponse = client.Execute(payRequst);
if (string.Compare(payResponse.Code, ToAlipayResultCode.SUCCESS, false) == 0)
{
if (payResponse.TradeStatus == "TRADE_FINISHED"
|| payResponse.TradeStatus == "TRADE_SUCCESS"
|| payResponse.TradeStatus == "TRADE_CLOSED")
// return payResponse;
break;
}

}

//未付款交易超时或等待超时。
if (payResponse.Code == ToAlipayResultCode.FAIL || payResponse.TradeStatus == "TRADE_CLOSED" ||
payResponse.TradeStatus == "WAIT_BUYER_PAY")
{
//撤销订单
StringBuilder param = new StringBuilder();
param.Append("{\"out_trade_no\":\"" + payResponse.OutTradeNo + "\"}");
biz_content = param.ToString();
Cancel(biz_content);
}

return payResponse;

}
}
```

前端效果图



销售单列表

订单支付

交易金额： 17983.00 元

付 款 码 ： 请扫描支付宝【付款】完成支付

订单ID	客户姓						单日期
917070	王燕						015/9/29 13:15:48
917066	曹岱缉						015/9/28 15:40:34
917033	王燕						015/9/24 11:10:06
917015	王燕						015/9/23 16:33:54
916996	ITL员工购买						2015/9/22 16:49:58
916976	褚连霞	13795215754	OE1509-0002025	谭述鹏 1640	已提交	未收款	2015/9/18 15:25:55
916798	ITL员工购买	26011188	OE1509-0001847	幸翼 0287	已提交	未收款	2015/9/10 10:34:29
914796	ITL员工购买	26011188	OE1508-0008372	幸翼 0287	已提交	未收款	2015/8/31 11:20:33

扫描枪自动提交，input输入框内“onkeyup=()”方法即可。

您可能感兴趣的文章:.NET/C#如何判断某个类是否是泛型类型或泛型接口的子类型详解C#支付宝新版支付请求接口调用C# WebApi 接口传参详解C# WebApi 接口返回值不困惑：返回值类型详解详解C# WebApi 接口测试工具：WebApiTestClientC# API中模型与它们的接口设计详解C#中IDispose接口的实现及为何这么实现详解C# 开发(创蓝253)手机短信验证码接口的实例C#调用webservice接口的最新方法教程理解C#编程中的静态类和静态成员以及密封类轻松学习C#的密封类C#中的应用程序接口介绍及实现,密封类与密封方法