## ZYTK3.5 ThirdWebService对接开发操作步骤 V1.0

# 对接概述

WebService是一种通过HTTP协议，以web应用服务器为依托，利用XML平台做交互的方式，实现跨平台的功能。一卡通提供WebService的对接接口，支持第三方软件系统与一卡通系统实现信息获取、身份验证和支付交易等功能。

# 对接需求

1. 第三方需提供访问WebService的IP，并在一卡通系统中注册站点（专用对接机）
2. 如果使用的接口中需要用到操作员和商户，需要第三方向一卡通系统申请操作员号（OptNum）和商户号（DealerNum）
3. 调用接口前，必须先调用签到认证，获取站点号和认证安全头需要的数据，该数据当天有效，跨天需要重新签到认证获取

# 参数描述说明

1. 数据集格式1：采用标准的Json格式存放

例如：{Table1:[{AREANUM:'1',AREANAME:'单位本部',REMARK:'不允许删除',RESERVED1:'001'}]}

1. 数据集格式2 ：采用标准的Xml格式存放

例如：

<NewDataSet>

<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">

<xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">

<xs:complexType>

<xs:choice minOccurs="0" maxOccurs="unbounded">

<xs:element name="Table">

<xs:complexType>

<xs:sequence>

<xs:element name="AREANUM" type="xs:decimal" minOccurs="0" />

<xs:element name="AREANAME" type="xs:string" minOccurs="0" />

<xs:element name="REMARK" type="xs:string" minOccurs="0" />

<xs:element name="RESERVED1" type="xs:string" minOccurs="0" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:choice>

</xs:complexType>

</xs:element>

</xs:schema>

<Table>

<AREANUM>1</AREANUM>

<AREANAME>正元智慧</AREANAME>

<REMARK>不允许删除</REMARK>

<RESERVED1>001</RESERVED1>

</Table>

</NewDataSet>

1. 照片数据：采用二进制数据存放

单个卡户照片获取时，传回的二进制数据就是照片

多个卡户照片获取时，按4字节卡户帐号+4字节照片数据长度+照片数据存放

简介：

正元智慧一卡通系统提供ThirdWebService接口供第三方应用程序调用，具体接口见对接开发指南，其流程和示例代码(**C#**)如下：

* + **登录与退出系统**



* + **身份验证流程**



* + **消费扣款流程**



uint nStaSid;   
string Secret2;   
// 1、添加web引用之后，实例化ThirdWebservice   
ThirdWebservice tws = new ThirdWebservice();   
// 2、签到   
tws.SignIn(out stanum, out ThirdType, out Secret1, out Secret2);//签到   
// 3、实例化安全头   
tws.SecurityHeaderValue = new SecurityHeader();   
// 4、保存加密因子   
tws.SecurityHeaderValue.ThirdType = ThirdType;   
tws.SecurityHeaderValue.Secret1 = Secret1;   
tws.SecurityHeaderValue.Secret2 = Secret2;

**NOTE：**声明ThirdWebS变量：

private ThirdWebService\_Test.ThirdWebService.ThirdWebservice ThirdWebS;

### 第1步：第三方签到认证

* 接口定义

public string SignIn(out nStaNum,out nThirdType,out nSecret1,out sSecret2);

* 功能说明

向一卡通签到，并获取一卡通分配的站点编号，**以及提供其他接口调用时需要传递安全头的加密因子，**

**NOTE：加密因子每天会变，换天后必须重新签到获取。**

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| nstanum | Integer | 传出 | 认证站点编号 |  |  |
| nThirdType | Integer | 传出 | 第三方类型 |  | 存放于SecurityHeader中 |
| nSecret1 | Integer | 传出 | 加密因子1 |  | 存放于SecurityHeader中 |
| sSecret2 | String | 传出 | 加密因子2 |  | 存放于SecurityHeader中 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

示例代码：

private void button2\_Click(object sender, EventArgs e)

{

if (tbWebServiceUrl.Text != string.Empty)

ThirdWebS.Url = tbWebServiceUrl.Text;

int nStaNum = 0;

int nThirdType, nSecret1;

string sSecret2;

string sReturn = ThirdWebS.SignIn(out **nStaNum**, out nThirdType, out nSecret1, out sSecret2);

if (sReturn == "1")

{

ThirdWebS.SecurityHeaderValue = new ThirdWebService.SecurityHeader();

ThirdWebS.SecurityHeaderValue.ThirdType = nThirdType;

ThirdWebS.SecurityHeaderValue.Secret1 = nSecret1;

ThirdWebS.SecurityHeaderValue.Secret2 = sSecret2;

tbStaNum.Text = nStaNum.ToString();

MessageBox.Show("签到成功 站点号：" + nStaNum.ToString());

}

else

MessageBox.Show("签到失败 返回:" + sReturn);

StaNum = nStaNum;

}

说明：注意传出的**nStaNum**为站点编号，它为后续操作传入的参数。

### 第2步：验证账户状态

* 接口定义

public string GetAccStatus(string sIDNo, int nIDType, out bool bValid);

* 功能说明

从一卡通平台获取指定卡户的照片信息。

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| bValid | bool | 传出 | 是否有效 |  |  |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

示例代码：

/// <summary>

/// 获取帐户状态

/// </summary>

/// <param name="sIDNo">卡户识别号</param>

/// <param name="nIDType">识别号类别</param>

/// <param name="bValid">有效为true，无效为false</param>

/// <returns>成功返回"1"，失败返回错误码</returns>

private void button25\_Click(object sender, EventArgs e)

{

string sIDNo = tbAccID.Text;

int nIDType = cbAccIDType.SelectedIndex+1;

bool bValid = false;

string sReturn = ThirdWebS.GetAccStatus(sIDNo, nIDType, out bValid);

if (sReturn == "1")

{

if (bValid)

tbIDAuAccStatus.Text = "有效卡";

else

tbIDAuAccStatus.Text = "无效卡";

MessageBox.Show("获取卡户状态成功");

}

else

MessageBox.Show("获取卡户状态失败 返回:" + sReturn);

}

### 第3步：判断卡户余额是否足够

* 接口定义

public string GetAccDBMoney(string sIDNo, int nIDType, int nEWalletNum, out int nMoney);

* 功能说明

从一卡通平台获取指定卡户的库余额；

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| nEWalletNum | Integer | 传入 | 钱包号 |  | 主钱包为1，共8个钱包 |
| nMoney | Integer | 传出 | 卡户余额 |  | 单位：分 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

示例代码：

/// <summary>

/// 获取卡户库余额

/// </summary>

/// <param name="sIDNo">卡户识别号</param>

/// <param name="nIDType">识别号类别（1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号）</param>

/// <param name="nEWalletNum">钱包号</param>

/// <param name="nMoney">卡户库余额（以分为单位）</param>

/// <returns>成功返回"1"，失败返回错误编码</returns>

private void button17\_Click(object sender, EventArgs e)

{

string sIDNo = tbAccID.Text;

int nIDType = cbAccIDType.SelectedIndex + 1;

int nEWalletNum = Convert.ToInt32(tbEwalletNum.Text);

int nMoney = 0;

string sReturn = ThirdWebS.GetAccDBMoney(sIDNo, nIDType, nEWalletNum,out nMoney);

if (sReturn == "1")

{

tbPayAccMon.Text = nMoney.ToString();

MessageBox.Show("获取卡户库余额成功 余额:" + nMoney.ToString());

}

else

MessageBox.Show("获取卡户库余额失败 返回:" + sReturn);

}

说明：将用户的交易金额与nMoney进行比较。支付交易

### 第4步获取卡户库余额

* 接口定义

public string GetAccDBMoney(string sIDNo, int nIDType, int nEWalletNum, out int nMoney);

* 功能说明

从一卡通平台获取指定卡户的库余额；

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| nEWalletNum | Integer | 传入 | 钱包号 |  | 主钱包为1，共8个钱包 |
| nMoney | Integer | 传出 | 卡户余额 |  | 单位：分 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

### 第5步：商务收费

* 接口定义

public string Payment(string sIDNo, int nIDType, int nStaNum, int nOptNum, int nDealerNum,

int nEWalletNum, int nMonDeal,out uint nStaSID);

* 功能说明

对一卡通卡户进行商务消费交易；

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| nStaNum | Integer | 传入 | 一卡通操作站点 |  | 通过签到获取站点 |
| nOptNum | Integer | 传入 | 一卡通操作员编号 |  | 由一卡通手工分配给第三方 |
| nDealerNum | Integer | 传入 | 一卡通商户编号 |  | 由一卡通手工分配给第三方 |
| nEWalletNum | Integer | 传入 | 交易钱包号 |  | 主钱包为1，共8个钱包 |
| nMonDeal | Integer | 传入 | 交易金额 |  | 单位：分 |
| nStaSID | uInteger | 传出 | 站点流水号 |  | 一卡通站点流水号 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

示例代码：

/// <summary>

/// 商务交易

/// </summary>

/// <param name="sIDNo">卡户识别号</param>

/// <param name="nIDType">识别号类别（1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号）</param>

/// <param name="nStaNum">站点号</param>

/// <param name="nOptNum">操作员号</param>

/// <param name="nDealerNum">商户号</param>

/// <param name="nEWalletNum">钱包号</param>

/// <param name="nMonDeal">交易金额（以分为单位）</param>

/// <param name="nStaSID">站点流水号(流水唯一标示)</param>

/// <returns>成功返回"1"，失败返回错误编码</returns>

private void button42\_Click(object sender, EventArgs e)

{

string sIDNo = tbAccID.Text;

int nIDType = cbAccIDType.SelectedIndex + 1;

int nStaNum = Convert.ToInt32(tbStaNum.Text);

int nOptNum = Convert.ToInt32(tbOptNum.Text);

int nDealerNum = Convert.ToInt32(tbDealerNum.Text);

int nEWalletNum = Convert.ToInt32(tbEwalletNum.Text);

int nMonDeal = Convert.ToInt32(tbPayMondeal.Text);

uint nStaSID = 0;

string sReturn = ThirdWebS.Payment(sIDNo, nIDType, nStaNum, nOptNum, nDealerNum, nEWalletNum, nMonDeal, out nStaSID);

if (sReturn == "1")

MessageBox.Show("商务交易成功，流水号：" + nStaSID.ToString());

else

MessageBox.Show("商务交易失败 返回："+sReturn);

}

**注意点**：需要一卡通分配操作员号：nOptNum，商户号：nDealerNum，交易钱包：默认是1。

#### 5.1如果在交易时需要验证交易密码，则用如下接口：

* 接口定义

public string ValidatePayPassword(string sIDNo, int nIDType, string Password);

* 功能说明

验证指定卡户的交易密码；

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| password | string | 传入 | 卡户交易密码 |  |  |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

示例代码：

/// <summary>

/// 验证交易密码

/// </summary>

/// <param name="sIDNo">卡户识别号</param>

/// <param name="nIDType">识别号类别</param>

/// <param name="Password">交易密码</param>

/// <returns>成功返回"1"，失败返回错误码</returns>

private void button52\_Click(object sender, EventArgs e)

{

string sIDNo = tbAccID.Text;

int nIDType = cbAccIDType.SelectedIndex + 1;

string Password = tbIDAuPayPwd.Text;

string sReturn = ThirdWebS.ValidatePayPassword(sIDNo, nIDType, Password);

if (sReturn == "1")

MessageBox.Show("交易密码验证成功");

else

MessageBox.Show("交易密码验证失败");

}

#### 5.2收费冲正

* 接口定义

public string AntiPayment(string sIDNo, int nIDType, int nStaNum, int nOptNum, int nDealerNum,

int nEWalletNum, int nMonDeal,out uint nStaSID);

* 功能说明

对一卡通的商务消费交易进行冲正；

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| nStaNum | Integer | 传入 | 一卡通操作站点 |  | 通过签到获取站点 |
| nOptNum | Integer | 传入 | 一卡通操作员编号 |  | 由一卡通手工分配给第三方 |
| nDealerNum | Integer | 传入 | 一卡通商户编号 |  | 由一卡通手工分配给第三方 |
| nEWalletNum | Integer | 传入 | 交易钱包号 |  | 主钱包为1，共8个钱包 |
| nMonDeal | Integer | 传入 | 交易金额 |  | 单位：分 |
| nStaSID | uInteger | 传出 | 站点流水号 |  | 一卡通站点流水号 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

示例代码：

/// <summary>

/// 收费冲正，校验交易许可，通过后产生正确的临时流水记录返回

/// </summary>

/// <param name="sIDNo">卡户识别号</param>

/// <param name="nIDType">识别号类别（1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号）</param>

/// <param name="nStaNum">站点号</param>

/// <param name="nOptNum">操作员号</param>

/// <param name="nDealerNum">商户号</param>

/// <param name="nEWalletNum">钱包号</param>

/// <param name="nMonDeal">交易金额（以分为单位）</param>

/// <param name="nStaSID">站点流水号(流水唯一标示)</param>

/// <returns>成功返回"1"，失败返回错误编码</returns>

private void button3\_Click(object sender, EventArgs e)

{

string sIDNo = tbAccID.Text;

int nIDType = cbAccIDType.SelectedIndex + 1;

int nStaNum = Convert.ToInt32(tbStaNum.Text);

int nOptNum = Convert.ToInt32(tbOptNum.Text);

int nDealerNum = Convert.ToInt32(tbDealerNum.Text);

int nEWalletNum = Convert.ToInt32(tbEwalletNum.Text);

int nMonDeal = Convert.ToInt32(tbPayMondeal.Text);

uint nStaSID = 0;

string sReturn = ThirdWebS.AntiPayment(sIDNo, nIDType, nStaNum, nOptNum, nDealerNum, nEWalletNum, nMonDeal, out nStaSID);

if (sReturn == "1")

MessageBox.Show("收费冲正成功，流水号：" + nStaSID.ToString());

else

MessageBox.Show("收费冲正失败 返回:" + sReturn);

}

#### 5.3无卡充值（带交易时间）

public string DepositA(string sIDNo, int nIDType, int nStaNum, int nOptNum,

int nEWalletNum, int nMonDeal, string sDealTime, out uint nStaSID);

* 功能说明

对一卡通的指定卡户进行充值；

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| nStaNum | Integer | 传入 | 一卡通操作站点 |  | 通过签到获取站点 |
| nOptNum | Integer | 传入 | 一卡通操作员编号 |  | 由一卡通手工分配给第三方 |
| nEWalletNum | Integer | 传入 | 交易钱包号 |  | 主钱包为1，共8个钱包 |
| nMonDeal | Integer | 传入 | 交易金额 |  | 单位：分 |
| sDealTime | String | 传入 | 交易时间 |  | 格式：yyyyMMddHHmmss |
| nStaSID | uInteger | 传出 | 站点流水号 |  | 一卡通站点流水号 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

#### 5.4获取交易流水

* 接口1定义

public string GetPaymentBooks(string sIDNo, int nIDType, string sBeginDate, string sEndDate, int nStaNum, int nTerminalNum, int nDealerNum, out string Data);

* 接口2定义

public string GetPaymentBooksForXml(string sIDNo, int nIDType, string sBeginDate, string sEndDate, int nStaNum, int nTerminalNum, int nDealerNum, out string Data);

* 功能说明

从一卡通平台获取指定该站点的交易流水。

* 接口1参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| sBeginDate | String | 传入 | 开始日期 |  | 格式：YYYYMMDD |
| sEndDate | String | 传入 | 结束日期 |  | 格式：YYYYMMDD |
| nStaNum | Integer | 传入 | 一卡通站点编号 |  | 通过签到获取站点 |
| nTerminalNum | Integer | 传入 | 终端号 |  | 传0表示所有终端号 |
| nDealerNum | Integer | 传入 | 商户编号 |  | 传0表示所有商户 |
| Data | string | 传出 | 流水记录集 |  | Json格式 |

* 接口2参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sIDNo | string | 传入 | 对接点识别号 | 64字节 |  |
| nIDType | Integer | 传入 | 识别号类别 |  | 1帐号、2卡号、3卡内编号、4个人编号、5第三方对接号 |
| sBeginDate | String | 传入 | 开始日期 |  | 格式：YYYYMMDD |
| sEndDate | String | 传入 | 结束日期 |  | 格式：YYYYMMDD |
| nStaNum | Integer | 传入 | 一卡通站点编号 |  | 通过签到获取站点 |
| nTerminalNum | Integer | 传入 | 终端号 |  | 传0表示所有终端号 |
| nDealerNum | Integer | 传入 | 商户编号 |  | 传0表示所有商户 |
| Data | string | 传出 | 流水记录集 |  | Xml格式 |

* 返回值

成功返回1，失败返回错误号，详细内容通过辅助接口查询

* 数据集字段说明

|  |  |
| --- | --- |
| **流水数据** | |
| **字段名** | **字段对应名称** |
| RN | 记录号 |
| RECID | 系统流水号 |
| PAYAREANUM | 交易园区号 |
| STANUM | 站点号 |
| STANAME | 站点名称 |
| ACCAREANUM | 卡户园区号 |
| ACCNUM | 卡户帐号 |
| WRITEFLAG | 写卡标示 |
| FEENUM | 交易类型编号 |
| EWALLETNUM | 交易钱包号 |
| MONDEAL | 交易金额（元） |
| MONCARD | 卡余额 |
| MONDBCURR | 库余额 |
| DEALERNUM | 商户编号 |
| DEALTIME | 交易时间 |
| RECTIME | 入账时间 |
| OPTNUM | 操作员号 |
| TERMINALNUM | 终端号 |
| STASID | 站点流水号（交易时返回的流水号） |
| TERSID | 终端流水号 |
| ACCNAME | 卡户姓名 |
| PERCODE | 卡户个人编号 |
| DEPNAME | 卡户所在部门 |
| CARDID | 卡户卡号 |
| FEENAME | 交易类型名称 |
| DEALERNAME | 商户名称 |

## 辅助接口

### 获取错误码描述

* 接口定义

public string GetErrTxt(string sErrCode, out string sErrTxt)

* 功能说明

获取异常错误内容，调用时可以不通过签到认证

* 参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数类型** | **传值方向** | **参数说明** | **取值范围** | **备注** |
| sErrCode | String | 传入 | 错误码 |  |  |
| sErrTxt | String | 传出 | 错误描述 |  |  |

* 返回值

成功返回 1，失败返回错误号

# 附表

## 错误号列表

|  |  |
| --- | --- |
| **错误号** | **错误释义** |
| 00000000 | 异常错误编码 |
| 00000001 | 入口参数错误 |
| 00000003 | 未通过加密因子校验 |

## 卡户信息记录集字段对应表

|  |  |
| --- | --- |
| **卡户信息** | |
| **字段名** | **字段对应名称** |
| AccNum | 卡户帐号 |
| CardID | 卡序列号 |
| CardCode | 卡内编号 |
| AccStatus | 卡户状态 |
| AccType | 卡户类型 |
| SourceAccNum | 主卡账号 |
| PerCode | 个人编号 |
| AreaNum | 园区号 |
| AccName | 卡户姓名 |
| DepNum | 卡户部门 |
| DepName | 部门名称 |
| ClsNum | 卡户身份号 |
| ClsName | 卡户身份名称 |
| AccSex | 性别 |
| mobileCode | 手机号码 |
| EMail | 邮箱 |
| PostCode | 邮政编号 |
| CertCode | 证件号 |
| CertTypeNum | 证件类型编号 |
| CertTypeName | 证件类型名称 |
| ArcClassNum | 档案身份编号 |
| ArcClassName | 档案身份名称 |
| NationNum | 民族编号 |
| NationName | 民族名称 |
| PolityNum | 政治面貌编号 |
| PolityName | 政治面貌名称 |
| DutyNum | 职务编号 |
| DutyName | 职务名称 |
| TechNum | 职称编号 |
| TechName | 职称名称 |
| PostNum | 岗位编号 |
| PostName | 岗位名称 |
| EnterTime | 入单位日期 |
| BirthDay | 生日 |
| PostDate | 配卡日期 |
| LostDate | 失效日期 |
| PreDelDate | 预撤户日期 |
| DelDate | 撤户日期 |
| Remark | 备注 |
| MonOpen | 开户金额 |
| Reserved1 | 籍贯 |
| Reserved2 | 联系地址 |
| Reserved3 | 预留字段 |
| Reserved4 | 第三方关键字 |
| FixID | 信息版本号 |