

中国银联外部积分接入 技术规范（正式版）

V2.0.0

中国银联
云闪付部
2022. 03

● 版本修订历史

版本号	修订点	修订人	修订时间
V1.0.0	按照银联平台通用技术规范修改接口规范		
V2.0.0	更新银联平台通用技术规范修改接口规范	付鹤	2022.3

中国银联 严格保密

目 录

前 言.....	2
1 范围.....	3
2 术语和定义.....	3
3 接口规范.....	3
4 加密/签名.....	5
5 请求参数传递规则.....	6
6 数据元定义.....	8
6.1 说明.....	8
7 外部机构基础接口定义.....	8
7.1 可用权益查询.....	8
7.2 可用权益扣减.....	11
7.3 权益撤销或退货.....	13
7.4 权益交易查询.....	15
8 外部机构兑换接口定义.....	19
8.1 兑换活动信息查询.....	19
8.2 活动积分兑换查询.....	20
8.3 权益兑换接口.....	21
8.4 权益兑换查询.....	22
附录一：错误码.....	24
附录二：RSA 签名验签方法.....	24
附录三：SM4 加解密方法.....	25

前 言

《中国银联外部积分接入技术规范》对银联受理外部积分接入作具体要求，业务规则按规定执行。本规范由中国银联云闪付部制定并负责解释、修订。

中国银联 严格保密

中国银联银行积分接入

接口规范

1 范围

本规范适用于所有接入中国银联营销网络的外部入网机构，主要包括外部积分机构。

2 术语和定义

1、积分兑换

通过云闪付APP或外部APP，使用外部权益（如积分、会员等）兑换银联红包、优惠券等标的物。

2、转接消费

使用云闪付发起银联消费时，通过云闪付平台使用外部机构权益（积分、会员等）进行消费扣减。

3 接口规范

3.1 基础规范

银联云闪付与外部机构之间的积分交易，调用接口必须遵循如下规则：

传输方式	为保证交易安全性，采用 HTTPS 传输（专线可使用 HTTP）
提交方式	采用 POST 方法提交
Content-Type	application/json;charset=utf-8
数据格式	请求数据以及响应数据为 JSON 格式
字符编码	报文使用 UTF-8 编码方式，采用 Unicode 字符集
签名算法	RSA 或 SM2（详细代码参照附录二、附录四）
签名要求	对消息头中公共参数及请求体中业务参数进行加签
加密算法	SM4（详细代码参照附录三：SM4 加解密方法）
加密要求	对请求体业务参数敏感信息进行加密
判断逻辑	判断响应参数字段返回 resultCode

http 消息头公共参数，请求报文和应答报文中均需要，具体定义如下表所示。

3.2 请求报文头

数据元中文名	标识	类型	出现条件	备注
版本号	version	String(8)	M	格式: xx.yy.zz 取值举例: 1.0.0
发送方索引类型	appType	String(2)	M	00: 银联索引
发送方索引标识码	appId	String(32)	M	银联发起请求时，默认填写银联代码(00010000)
接口类型	bizMethod	String(64)	M	接口名称至少包含两级。取值: 宜全英文，不应出现中文 取值举例: pay.query.tradequery
签名	sign	String(344)	M	由请求或应答的发送方根据报文签名方法生成
证书 id	signId	String(128)	M	请填写接入方 ID
签名或摘要方式	signMethod	String(20)	M	RSA2/SM2
发送方流水号	reqId	String(64)		由请求的发送方自行生成，保证每笔请求不重复，但非交易主键。同步应答原样返回，异步应答由应答的发送方重新生成新值。

3.3 应答报文头

数据元中文名	标识	类型	出现条件	备注
版本号	version	String(8)	M	格式: xx.yy.zz 取值举例: 1.0.0
发送方索引类型	appType	String(2)	M	01: 机构索引
发送方索引标识码	appId	String(32)	M	机构发起应答或请求时，默认填写银联分配给机构的接入方 ID
接口类型	bizMethod	String(64)	M	接口名称至少包含两级。取值: 宜全英文，不应出现中文 取值举例: pay.query.tradequery

签名	sign	String(344)	M	由请求或应答的发送方根据报文签名方法生成
证书 id	signId	String(128)	M	请填写接入方 ID
签名或摘要方式	signMethod	String(20)	M	RSA2/SM2
发送方流水号	reqId	String(64)		由请求的发送方自行生成，保证每笔请求不重复，但非交易主键。同步应答原样返回，异步应答由应答的发送方重新生成新值。

4 加密/签名

为了防止请求报文在传输的过程中被窃取或者篡改，需要对指定敏感参数信息进行加密，并对请求头中公共参数和请求体中业务参数进行签名。

外部机构申请接入时，银联会为外部机构生成一对(RSA/SM2)公私钥及 SM4 加密密钥,并将公钥及 SM4 加密密钥交给外部机构保存。银联调用外部机构系统时，将根据对应 SM4 加密密钥对敏感信息加密，对应私钥进行(RSA/SM2)签名。返回响应数据时，由于不涉及敏感信息，故无需加密。如果外部系统需要访问银联接口，需提供(RSA/SM2)公钥交由银联进行配置，银联将根据公钥进行验签。

4.1 签名方式

接口对敏感信息采用 SM4 加密的方式，加密算法为 SM4/ECB/PKCS5Padding；请求报文采用 (RSA/SM2)签名的方式，将待签名字段封装成 JSON 格式字符串，针对敏感信息加密成密文，并以密文参与生成签名。签名时，所有上送字段，无论是否为空，均应参与签名与验签。

测试环境签名密钥：联调时提供

生产环境签名密钥：后续线下提供

4.2 待签名串生成方式

报文中的数据元按照如下顺序排列，并按照 Key1=Value1&Key2=Value2...KeyN=ValueN 方式拼接成字符串，需要去除前后空白符。数据元顺序及对应标识如下表所示。

顺序	位置	标识	备注
1	http 头	version	
2	http 头	appId	
3	http 头	bizMethod	

4	http 头	reqId	
5	http 体	body	指整个 http 报文体 json 串（ASCII 码升序）

示例：待签名串

```
version=1.0.0&appId=8888888&bizMethod=pay.query&reqId=8164ILTAVBCH16C12502SI8ZN134567&body={"aa":"1111","bb":"2222"}
```

5 请求参数传递规则

请求参数以 JSON 字符串的形式提交，Post 请求 MIME 类型为 application/json;charset=UTF-8，其中待加密以及待签名字段详见各接口章节。

如可用权益查询接口（详见 7.1 章节），请求报文如下：

```
POST https://sample.95516.com/api/trans.do http/1.1
Date: Tue,17 Nov 2020 10:10:10 GMT
Accept: application/json
Connection: close
User-Agent: curl/7.60.0
Content-Type: application/json
Connect-Length:80
version: 1.0.0
appType:00
appId: 00010000
bizMethod: pay.query.point
signId: 888888
signMethod: RSA2
reqId: 8164ILTAVBCH16C12502SI8ZN134567V
{"orderNo":"21060109091528877740060000129124","transChnl":"","acctEntityId":"6217000000000000000","acctEntityTp":"UP02","activityNo":"5199210531179928","orderAmt":"","pointType":"01","minTransPoint":"10","bussTp":"1002","transSeq":"21060109091528877740060000292320","accessId":"104007","appId":"","reqReserved":"","sceneId":"","maxTransPoint":"30"}
```

5.1 将请求体中敏感信息进行 SM4 加密

```
{"orderNo":"21060109091528877740060000129124","transChnl":"","acctEntityId":"DAyidnbz2jdI8PdL6roIglVvQPD4Nqxz3h8YxDQv+e8=","acctEntityTp":"UP02","activityNo":"5199210531179928","orderAmt":"","pointType":"01","minTransPoint":"10","bussTp":"1002","transSeq":"21060109091528877740060000292320","accessId":"104007","appId":"","reqReserved":"","sceneId":"","maxTransPoint":"30"}
```

5.2 生成待加签串 paramString

```
version=1.0.0&appId=8888888&bizMethod=pay.query.poin&reqId=8164ILTAVBCH16C12502SI8ZN134567&body={"accessId":"104007","acctEntityId":"DAyidnbz2jdI8PdL6roIgIVvQPD4Nqxz3h8YxDQv+e8=","acctEntityTp":"UP02","activityNo":"5199210531179928","appId":"","bussTp":"1002","maxTransPoint":"30","minTransPoint":"10","orderAmt":"","orderNo":"21060109091528877740060000129124","pointType":"01","reqReserved":"","sceneId":"","transChnl":"","transSeq":"21060109091528877740060000292320"}
```

5.3 将 paramString 用接入方私钥加签

示例 sign 加签结果：

```
QXzWuwpabTa2GS+GABRD3OxqCZ+JZul2qAXZecWwGII/FR/jUQUyJr7fxPHx/OHTnPZI67oKfwdCaZXJFe51J+6YFvvgWJRhwfFZmqAJ60ItWWXmq9fNhIAbK04z/x9hN8Bpff4qagyoi1YcHNAf7ohzRmuUa9qDYK5e/iB3exo=
```

5.4 将签名放入 http 请求头中

最终请求报文：

```
POST https://sample.95516.com/api/trans.do http/1.1
Date: Tue,17 Nov 2020 10:10:10 GMT
Accept: application/json
Connection: close
User-Agent: curl/7.60.0
Content-Type: application/json
Connect-Length:80
version: 1.0.0
appType:00
appId: 00010000
bizMethod: pay.query.point
signId: 888888
sign: QXzWuwpabTa2GS+GABRD3OxqCZ+JZul2qAXZecWwGII/FR/jUQUyJr7fxPHx/OHTnPZI67oKfwdCaZXJFe51J+6YFvvgWJRhwfFZmqAJ60ItWWXmq9fNhIAbK04z/x9hN8Bpff4qagyoi1YcHNAf7ohzRmuUa9qDYK5e/iB3exo=
signMethod: RSA2
reqId: 8164ILTAVBCH16C12502SI8ZN134567V
{"orderNo":"21060109091528877740060000129124","transChnl":"","acctEntityId":" DAyidnbz2jdI8PdL6roIgIVvQPD4Nqxz3h8YxDQv+e8=","acctEntityTp":"UP02","activityNo":"5199210531179928","orderAmt":"","pointType":"01","minTransPoint":"10","bussTp":"1002","transSeq":"21060109091528877740060000292320","accessId":"104007","appId":"","reqReserved":"","sceneId":"","maxTransPoint":"30"}
```

6 数据元定义

6.1 说明

表 1 数据元条件符号

符号	含义	
数据元数值符号	M	必须填写
	C	某条件成立时必须填写
	C+	某条件成立时增加
	C-	某条件成立时删除
	M+	必须增加
	O	受理方和发卡方自选填写
	→	转发
		必须去除的域

表 2 报文域数据类型和长度

符号	含义
a	字母字符, A至Z, a至z
n	数值, 0至9
s	汉字和特殊符号
an	字母和数字字符,
MM	月份, 01至12
DD	日期, 01至31
YYYY	年份, 00至99
hh	时, 00至23
mm	分, 00至59
ss	秒, 00至59
n12	12字符固定长度的数字
an3	3字符固定长度的字母和数字字符
an3..17	最小3, 最大17个字符的可变长度字母和数字字符

7 外部机构基础接口定义

发送方: 银联积分平台 —— 接收方: 外部机构

7.1 可用权益查询

接口名: **org.point.query**

7.1.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
接入方 ID	accessId	16	M	配置的接入方 ID

交易流水号	transSeq	64	M	用于标识唯一一笔操作，全局唯一
账户主体类型	acctEntityTp	4	M	UP01-手机号 UP02-卡号 UP03-银联统一用户 userId
账户主体标识	acctEntityId	128	M	银行传输卡号；商户传输手机号 加密传输
业务类型	bussTp	4	M	积分消费： 1002-积分兑换 1003-转接消费
权益类型	pointType	2	M	不填默认 01-普通积分， 01-普通积分 02-权益活动
订单金额	orderAmt	10	C	传整笔订单的金额，用于查询计算最大可抵扣的积分（转接消费时必传）
订单号	orderNo	60	M	填写上游订单号或系统自动生成，关联查询和消费的同一订单； 同一订单号可能会发起多笔查询
活动 ID	activityNo	64	C	活动 ID，选填；积分兑换时是兑换活动 ID；积分抵现时是抵现活动 ID
交易权益上限	maxTransPoint	10	C	部分银行必输，只针对兑换
交易权益下限	minTransPoint	10	C	部分银行必输，只针对兑换
交易 APP 标识	appld	64	C	选填，判断交易渠道时填写
交易渠道	transChnl	4	C	选填，判断交易渠道时填写
交易场景 ID	sceneld	2	C	选填，判断交易渠道时填写
请求保留域	reqReserved	512	C	{"field1":"value1","field2":"value2", "field...":"value..."}

7.1.2 应答报文格式

参数名	英文名 KEY	最大长度	必填	备注
响应码	resultCode	6	M	000000：成功；其他均为失败
响应信息	resultMsg	100	M	
权益总额	ttlBalance	12	C	只有当 resultCode 为 000000，该字段非空
可用权益数量	avlBalance	12	M	只有当 resultCode 为 000000，该字段非空（返回该笔订单最大可用于抵扣的积分值）
可抵扣金额	discountAmt	10	M	选填，若无判断，则不返回。
权益类型	pointType	2	M	选填，不填默认 01-普通积分 01-普通积分 02-权益活动
即将过期权益额	pointRectExpire	12	C	有则填，否则填 0

即将过期日期	dateExpire	14	C	有则填，否则填 0
即将过期说明	expireDesc	512	C	用于在前端界面中展示即将过期积分的过期原因说明。
永久有效权益额	permanentScore	12	C	有则填
应答保留域	rspReserved	512	C	

7.1.3 报文示例

1) 积分兑换报文示例

请求报文

```
{
  "orderNo": "8877740060000795144",
  "transChnl": "",
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQTKlJYqE13+",
  "acctEntityTp": "UP02",
  "activityNo": "5199210531179928",
  "orderAmt": "",
  "pointType": "01",
  "minTransPoint": "10",
  "bussTp": "1002",
  "transSeq": "21062916201728877740060000821894",
  "accessId": "1000000",
  "appld": "",
  "reqReserved": "",
  "sceneld": "",
  "maxTransPoint": "30"
}
```

2) 转接消费报文示例

请求报文

```
{
  "orderNo": "8877740060000795144",
  "transChnl": "0001",
  "acctEntityTp": "UP02",
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQTKlJYqE13+",
  "activityNo": "5199210531179966",
  "orderAmt": "10000",
  "pointType": "01",
  "minTransPoint": "",
  "bussTp": "1003",
  "transSeq": "21062916201728877740060000821531",
  "accessId": "1000000",
  "appld": "4",
  "reqReserved": "",
  "sceneld": "21",
  "maxTransPoint": ""
}
```

}

7.2 可用权益扣减

接口名: org.point.deduct

7.2.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
接入方 ID	accessId	16	M	配置的接入方 ID
账户主体类型	acctEntityTp	4	M	UP01-手机号 UP02-卡号 UP03-银联统一用户 userId
账户主体标识	acctEntityId	128	M	加密
业务类型	bussTp	4	M	积分消费: 1002-积分兑换 1003-转接消费
交易流水号	transSeq	64	M	用于标识唯一一笔操作, 全局唯一
订单号	orderNo	60	C	填写上游订单号或系统自动生成, 关联查询和消费的唯一订单; 同一订单号可能会发起多次支付请求
商户代码	mchntCd	15	C	
境内外交易标识	inOutTransFlag	1	C	
终端号	termId	8	C	
地区代码	regionCd	4	C	
币种	transCurrCd	4	C	
交易 APP 标识	appld	64	C	
交易渠道	transChnl	4	C	
支付方式	payMethod	16	C	
支付产品类型	payProductTp	16	C	
交易场景 ID	sceneld	2	C	
渠道商户号	chnlMchntCd	15	C	
活动 ID	activityNo	64	C	活动 ID, 选填
权益类型	pointType	2	M	
扣减权益数量	pointAt	12	M	
交易时间	transTime	14	M	格式为 yyyyMMddHHmmss
交易描述	transDesc	100	C	
备注	remark	512	C	
请求保留域	reqReserved	512	C	{"field1":"value1","field2":"value2", "field...":"value..."}

7.2.2 应答报文格式

参数名	英文名 KEY	最大长度	必填	备注
响应码	resultCode	6	M	000000: 成功; 其他均为失败
响应信息	resultMsg	100	M	
扣减权益数	pointAt	12	C	只有当 resultCode 为 000000, 该字段非空
支付交易凭证	transVoucher	50	C	只有当 resultCode 为 000000, 该字段非空
支付结算日期	settleDate	8	C	只有当 resultCode 为 000000, 该字段非空
备注	remark	512	C	
应答保留域	rspReserved	512	C	

7.2.3 报文示例:

1) 积分兑换报文示例

```

{
  "transChnl": "",
  "acctEntityTp": "UP02",
  "remark": "",
  "inOutTransFlag": "0",
  "transSeq": "8121062916132518",
  "accessId": "1000000",
  "termId": "",
  "payProductTp": "",
  "payMethod": "",
  "appld": "",
  "transCurrCd": "",
  "reqReserved": "",
  "sceneld": "",
  "orderNo": "21062916130128877740060000672436",
  "transTime": "20210629161307",
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtkIJYqE13+",
  "activityNo": "5199210531179928",
  "pointType": "01",
  "bussTp": "1002",
  "regionCd": "",
  "transDesc": "",
  "chnlMchntCd": "",
  "pointAt": "10",
  "mchntCd": ""
}

```

2) 转账消费报文示例

```

{

```

```

"transChnl": "0001",
"acctEntityTp": "UP02",
"remark": "",
"inOutTransFlag": "0",
"transSeq": "8121062916132518",
"accessId": "1000000",
"termId": "D123456",
"payProductTp": "M6",
"payMethod": "9999",
"appId": "4",
"transCurrCd": "156",
"reqReserved": "",
"sceneId": "21",
"orderNo": "21062916130128877740060000672436",
"transTime": "20210629161307",
"acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtk1JYqE13+",
"activityNo": "5199210531179928",
"pointType": "01",
"bussTp": "1003",
"regionCd": "0156",
"transDesc": "",
"chnlMchntCd": "lokiu3254698751",
"pointAt": "10",
"mchntCd": "4444"
}

```

7.3 权益撤销或退货

接口名：org.point.refund

7.3.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
接入方 ID	accessId	16	M	配置的接入方 ID
账户主体类型	acctEntityTp	4	M	UP01-手机号 UP02-卡号 UP03-银联统一用户 userId
账户主体标识	acctEntityId	128	M	加密
权益类型	pointType	2	M	<u>不填默认 01-普通积分,</u> <u>01-普通积分</u> <u>02-权益活动</u>
业务类型	BussTp	4	M	1004-交易撤销 1005-交易退货
原交易业务类型	orgBussTp	4	M	1002-积分兑换 1003-转接消费

原交易流水号	orgTransSeq	64	M	
交易流水号	transSeq	64	M	用于标识唯一一笔操作，全局唯一
退回权益数	pointAt	12	M	
交易时间	transTime	14	M	格式为 yyyyMMddHHmmss
退货或撤销原因	refundReason	100	M	
请求保留域	reqReserved	512	C	{"field1":"value1","field2":"value2", "field...":"value..."}

7.3.2 应答报文格式

参数名	英文名 KEY	最大长度	必填	备注
响应码	resultCode	6	M	000000: 成功; 其他均为失败
响应信息	resultMsg	100	M	
支付交易凭证	transVoucher	50	C	用于唯一标识一笔退款，只有当 resultCode 为 000000, 该字段非空
退回权益数	pointAt	12	C	实际退或撤销积分数，只有当 resultCode 为 000000, 该字段非空
剩余未退权益数	remainPoint	12	C	根据支付的总积分数，尚未退完的积分数，在部分退的情况下该字段不为 0, , 其余时候为 0, 只有当 resultCode 为 000000, 该字段非空
支付结算日期	settleDate	8	C	用于标识该笔交易出现在哪天的对账文件里，格式为 yyyyMMdd, 只有当 resultCode 为 000000, 该字段非空
应答保留域	rspReserved	512	C	

7.3.3 报文示例

1) 积分兑换报文示例

兑换撤销请求报文

```
{
  "accessId": "1000000",
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtklJYqE13+",
  "acctEntityTp": "UP02",
  "BussTp": "1004",
  "orgBussTp": "1002",
  "orgTransSeq": "21202106301238493902020422",
  "pointAt": "10",
  "pointType": "01",
  "refundReason": "红包赠送失败撤销积分",
  "reqReserved": "",

```

```
"transSeq": "21202106301238496789345466260",  
"transTime": "20210630101010"  
}
```

2) 转接消费报文示例

消费撤销

```
{  
  "accessId": "1000000",  
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtkIJYqE13+",  
  "acctEntityTp": "UP02",  
  "BussTp": "1004",  
  "orgBussTp": "1003",  
  "orgTransSeq": "21202106301238493902020400",  
  "pointAt": "10",  
  "pointType": "01",  
  "refundReason": "消费撤销",  
  "reqReserved": "",  
  "transSeq": "21202106301238496789345466241",  
  "transTime": "20210630101010"  
}
```

消费退货

```
{  
  "accessId": "1000000",  
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtkIJYqE13+",  
  "acctEntityTp": "UP02",  
  "BussTp": "1005",  
  "orgBussTp": "1003",  
  "orgTransSeq": "21202106301238493902020400",  
  "pointAt": "10",  
  "pointType": "01",  
  "refundReason": "消费退货",  
  "reqReserved": "",  
  "transSeq": "21202106301238496789345466221",  
  "transTime": "20210630101010"  
}
```

7.4 权益交易查询

接口名：org.trans.query

7.4.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
接入方 ID	accessId	16	M	配置的接入方 ID
交易流水号	transSeq	64	M	用于标识唯一一笔操作，全局唯一

业务类型	BussTp	4	M	5002-交易查询 5003-明细查询 标记是查原交易，还是查明细
账户主体类型	acctEntityTp	4	M	UP01-手机号 UP02-卡号 UP03-银联统一用户 userId
账户主体标识	acctEntityId	128	C	加密
开始时间	beginTime	14	C	查询类型为 5003 时必填 格式为 yyyyMMddHHmmss
结束时间	endTime	14	C	查询类型为 5003 时必填 格式为 yyyyMMddHHmmss
原交易流水号	orgTransSeq	64	C	查询类型为 5002 时，必填。 支付或者退款时传入的事务流水号， 用于标识唯一的原一笔支付或者退款
原交易业务类型	orgBussTp	4	C	查询类型为 5002 时，必填。 1002-积分兑换 1003-转接消费 1004-交易撤销 1005-交易退货
原交易时间	orgTransDt	14	C	查询类型为 5002 时必填 格式为 yyyyMMddHHmmss
请求保留域	reqReserved	512	C	{"field1":"value1","field2":"value2", "field...":"value..."}

7.4.2 应答报文格式

参数名	英文名 KEY	最大长度	必填	备注
响应码	resultCode	6	M	000000: 成功; 其他均为失败
响应信息	resultMsg	100	M	
支付交易凭证	tranVoucher	50	C	查询类型为 5002 时，必填; 其他不填 返回查询的那笔交易的支付凭证
支付结算日期	settleDate	8	C	用于标识该笔交易出现在哪天的对账文件里，格式为 yyyyMMdd
原交易时间	transTime	14	M	格式为 yyyyMMddHHmmss
交易权益数	pointAt	12	C	用于标识一笔支付或者退款的交易积分
历史积分变动	variationScore	List	C	查询类型为 5003 时，必填; 其他不填 为一个 list，包含如下参数： 月份 month，卡号 cardNo，积分变动值 pointAt

查询交易状态	seqStatus	2	C	查询类型为 5002 时, 必填, 其他不填 标记查询的原交易的交易状态 000000 成功; 其他失败
应答保留域	rspReserved	512	C	

variation_score 参数

名称	简要描述	类型	填写说明
monthdata1	变动月份 1	O	Array
monthdata2	变动月份 2	O	Array
monthdata3	变动月份 3	O	Array
monthdata4	变动月份 4	O	Array
monthdata5	变动月份 5	O	Array
monthdata6	变动月份 6	O	Array

每个 monthdata 包含参数如下:

respData参数

名称	简要描述	类型	填写说明	备注
Month	月份	String6	C	YYYYMM, 自然月。可以是当前月
gain_score	该月获取积分	String1..8	C	
cost_score	该月支出积分	String1..8	C	若有支出等原因减少的积分, 以负值表示

7.4.3 报文示例

积分兑换交易查询

```
{
  "accessId": "1000000",
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQTKlJYqE13+",
  "acctEntityTp": "UP02",
  "beginTime": "",
  "BussTp": "5002",
  "endTime": "",
  "orgBussTp": "1002",
  "orgTransDt": "20210630101010",
  "orgTransSeq": "2121062973776355778567886563",
  "reqReserved": "",
  "transSeq": "2121062973776355778567886811"
}
```

转接消费交易查询

```
{
  "accessId": "1000000",
  "acctEntityId": "8qYnyojcUhWmrXi8TsRrVQTKlJYqE13+",
  "acctEntityTp": "UP02",
```

```
"beginTime": "",
"BusstTp": "5002",
"endTime": "",
"orgBusstTp": "1003",
"orgTransDt": "20210630101010",
"orgTransSeq": "2121062973776355778567880364",
"reqReserved": "",
"transSeq": "2121062973776355778567886421"
```

```
}
```

撤销交易查询

```
{
```

```
"accessId": "1000000",
"acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtk1JYqE13+",
"acctEntityTp": "UP02",
"beginTime": "",
"BusstTp": "5002",
"endTime": "",
"orgBusstTp": "1004",
"orgTransDt": "20210630101010",
"orgTransSeq": "2121062973776355778567883663",
"reqReserved": "",
"transSeq": "2121062973776355778567881101"
```

```
}
```

退货交易查询

```
{
```

```
"accessId": "1000000",
"acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtk1JYqE13+",
"acctEntityTp": "UP02",
"beginTime": "",
"BusstTp": "5002",
"endTime": "",
"orgBusstTp": "1005",
"orgTransDt": "20210630101010",
"orgTransSeq": "21210629737763557785678821345",
"reqReserved": "",
"transSeq": "21210629737763557785678890367"
```

```
}
```

2) 明细查询报文示例

积分明细查询请求报文

```
{
```

```
"accessId": "1000000",
"acctEntityId": "8qYnyojcUhWmrXi8TsRrVQtk1JYqE13+",
"acctEntityTp": "UP02",
```

```

    "beginTime": "20210601101010",
    "BusstTp": "5003",
    "endTime": "20210630101010",
    "orgBusstTp": "",
    "orgTransDt": "",
    "orgTransSeq": "",
    "reqReserved": "",
    "transSeq": "2121062973776355778567886811"
}

```

8 外部机构兑换接口定义

发送方：外部积分发行机构；接收方：银联

8.1 兑换活动信息查询

接口地址：<https://point.95516.com/callOpen/activity/exchange/query>

8.1.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
cmd	cmd	20	M	queryActivitySvc
接入方 ID	accessId	16	M	配置的接入方 ID
应用 ID	appld	16	M	接入方申请的 appld
查询交易流水号	transSeq	64	M	
业务类型	BusstTp	4	M	1002-积分兑换
活动编号	activityNo	16	C	
活动积分 ID	pointId	16	C	外部积分 ID
发行机构号	issInsCd	8	M	
活动开始日期	beginAt	14	C	yyyyMMddHHmmss
活动结束日期	endAt	14	C	yyyyMMddHHmmss
时间戳	timestamp		M	生成签名的时间戳

8.1.2 响应报文格式

参数名	英文名 KEY	必填	备注
应答码	respCd	M	000000: 成功 其它:失败
错误码	errCd		参考附录一
错误信息	respMsg		
查询交易流水号	transSeq	M	
业务类型	activityType	M	1002-权益兑换
活动编号	activityList-> activityNo		
活动名称	activityList-> activityNm		

启用状态	activityList-> status		0: 停用 1: 启用
兑换比例	activityList-> targetRate		
兑换乘数	activityList-> multiplyNum		
标的物总数	activityList-> totalNum		
剩余标的物数	activityList-> surplusNum		
开始时间	activityList-> beginAt		yyyyMMddHHmmss
结束时间	activityList-> endAt		yyyyMMddHHmmss
兑换下限	activityList-> minTargetNum		
兑换上限	activityList-> maxTargetNum		
起兑物名称	activityList-> deductNm		
标的物名称	activityList-> targetNm		
标的物图片	activityList-> targetImgPath		
创建时间	activityList-> crtTime		yyyyMMddHHmmss
最后修改时间	activityList-> uptTime		yyyyMMddHHmmss
保留字段	rspReserved		

8.2 活动积分兑换查询

接口地址: <https://point.95516.com/callOpen/activity/exchange/point/query>

8.2.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
cmd	cmd	20	M	queryActPointSvc
接入方 ID	accessId	16	M	配置的接入方 ID
应用 ID	appId	16	M	接入方申请的 appId
查询交易流水号	transSeq	64	M	
账户类型	acctEntityTp	4	M	UP01: 手机号 UP02: 卡号 UP03: userId
账户标识	acctEntityId	64	M	
活动号	activityNo	16	M	
时间戳	timestamp		M	生成签名的时间戳

8.2.2 响应报文格式

参数名	英文名 KEY	必填	备注
应答码	respCd	M	000000: 成功 其它:失败
错误码	errCd	C	参考附录一

错误信息	respMsg	C	
查询交易流水号	transSeq	M	
账户类型	acctEntityTp		UP01: 手机号 UP02: 卡号 UP03: userId
账户标识	acctEntityId	M	
积分类型 ID	pointTpId	M	
积分余额	avlPoint		
使用规则	useRule		
保留字段	rspReserved		

8.3 权益兑换接口

接口地址: <https://point.95516.com/call0pen/activity/exchange>

8.3.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
cmd	cmd	20	M	exchangeSvc
接入方 ID	accessId	16	M	配置的接入方 ID
应用 ID	appld	16	M	接入方申请的 appld
交易跟踪号	traceld	16	M	
交易流水号	transSeq	64	M	
账户类型	acctEntityTp	4	M	UP01: 手机号 UP02: 卡号 UP03: userId
账户标识	acctEntityId	64	M	
活动编号	activityNo	16	M	
积分类型 ID	pointTpId	16	M	
可用积分余额	avlPoint		M	
兑换数量	targetNum		M	兑换上下限之间
交易时间	transTime		M	yyyyMMddHHmmss
时间戳	timestamp		M	生成签名的时间戳

8.3.2 响应报文格式

参数名	英文名 KEY	必填	备注
应答码	respCd	M	000000: 成功 其它:失败
错误码	errCd	C	参考附录一
错误信息	respMsg	C	

交易流水号	transSeq		
账户类型	acctEntityTp		UP01: 手机号 UP02: 卡号 UP03: userId
活动编号	activityNo		
兑换数量	targetNum		
保留字段	rspReserved		

8.4 权益兑换查询

接口地址: <https://point.95516.com/call0pen/activity/exchange/trans/query>

8.4.1 请求报文格式

参数名	英文名 KEY	最大长度	必填	备注
cmd	cmd	20	M	transQuerySvc
接入方 ID	accessId	16	M	配置的接入方 ID
应用 ID	appld	16	M	接入方申请的 appld
原交易类型	orgBussTp	4	M	1002: 权益兑换
原交易流水号	orgTransSeq	64	M	
交易流水号	transSeq	64	M	
账户类型	acctEntityTp	4	M	UP01: 手机号 UP02: 卡号 UP03: userId
账户 ID	acctEntityId	64	M	
交易时间	transTime		M	yyyyMMddHHmmss
时间戳	timestamp		M	生成签名的时间戳

8.4.2 响应报文格式

参数名	英文名 KEY	必填	备注
应答码	respCd	M	000000: 成功 其它:失败
错误码	errCd	C	参考附录一
错误信息	respMsg	C	
账户类型	acctEntityTp		UP01: 手机号 UP02: 卡号 UP03: userId
账户 ID	acctEntityId		
活动号	activityNo		
交易凭证	tranVoucher		
结算日期	settleDate		
交易时间	transTime		

交易权益数	pointAt		
兑换数量	targetNm		
查询交易状态	seqStatus		标记查询的原交易的交易状态 000000 成功; 其他失败
保留字段	rspReserved		

中国银联 严格保密

附录一：错误码

银联请求机构错误应答码：

错误码	描述
IS0001	重复提交
IS0002	验签失败
IS0003	敏感信息解密失败
IS0004	请求参数不合法
IS0005	非法账户信息
IS0006	余额查询异常
IS0007	积分扣减异常
IS0008	未找到原交易
IS0009	积分撤销异常
IS0010	其他错误

机构请求银联错误应答码：

错误码	描述
UP0800	非法接入方
UP0801	无接口权限
UP0802	验签失败
UP0803	敏感信息解密失败
UP0804	参数不合法
UP0805	用户未实名认证
UP0806	积分余额查询失败
UP0807	积分余额不足
UP0808	不符合活动规则
UP0809	兑换失败
UP0810	系统异常

附录二：RSA 签名验签方法

加签方法

```
public static final String SIGN_ALGORITHMS = "SHA256WithRSA";

public static String sign(String value, String privateKey) throws Exception {
    byte[] keyBytes = Base64Utils.decodeFromString(privateKey);
    PKCS8EncodedKeySpec keySpec = new PKCS8EncodedKeySpec(keyBytes);
    KeyFactory keyf = KeyFactory.getInstance("RSA");
    PrivateKey priKey = keyf.generatePrivate(keySpec);
```

```

Signature signature = Signature.getInstance(SIGN_ALGORITHMS);
signature.initSign(priKey);
signature.update(value.getBytes("UTF-8"));
byte[] signed = signature.sign();
String result = Base64Utils.encodeToString(signed);
return result;
}

```

验签方法

```

public static boolean verify(String value, String publicKey,String sign) throws Exception{
    byte[] keyBytes = Base64.decodeBase64(publicKey);
    X509EncodedKeySpec keySpec = new X509EncodedKeySpec (keyBytes);
    KeyFactory keyf = KeyFactory.getInstance("RSA");
    PublicKey pubkey = keyf.generatePublic(keySpec);
    Signature signature = Signature.getInstance("SHA256WithRSA");
    signature.initVerify(pubkey);
    signature.update(value.getBytes("UTF-8"));
    boolean result = signature.verify(Base64.decodeBase64(sign.getBytes()));
    return result;
}

```

附录三：SM4 加解密方法

敏感字段加密采用 SM4 加密算法，java 版本的 SM4 加密方法示例

```

import org.bouncycastle.jce.provider.BouncyCastleProvider;
import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;
import java.io.*;
import java.nio.charset.StandardCharsets;
import java.security.*;
import java.util.Arrays;
import java.util.Base64;

public class SM4Utils {

    static {
        Security.addProvider(new BouncyCastleProvider());
    }

    public static final String ALGORITHM_NAME = "SM4";
    public static final String ALGORITHM_NAME_ECB_PADDING = "SM4/ECB/PKCS5Padding";

    /**
     * SM4 算法目前只支持 128 位（即密钥 16 字节）
     */
}

```

```

*/
public static final int DEFAULT_KEY_SIZE = 128;

/**
 * 生成密钥
 *
 * @return
 */
public static String generateKey() {
    String key = "";
    try {
        KeyGenerator generator = KeyGenerator.getInstance(ALGORITHM_NAME,
BouncyCastleProvider.PROVIDER_NAME);
        SecureRandom secureRandom = SecureRandom.getInstance("SHA1PRNG");
        generator.init(DEFAULT_KEY_SIZE, secureRandom);
        SecretKey secretKey = generator.generateKey();
        key = byteToHexString(secretKey.getEncoded());
    } catch (Exception e) {
        e.printStackTrace();
    }
    return key;
}

/**
 * 敏感信息加密
 *
 * @param sourceData 明文
 * @param key 密钥
 * @return
 */
public static String encryptMessageBySM4(String sourceData, String key) {
    try {
        SecretKeySpec secretKey = getKey(key);
        Cipher cipher = Cipher.getInstance(ALGORITHM_NAME_ECB_PADDING);
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        return Base64.getEncoder().encodeToString(cipher.doFinal(sourceData.getBytes("UTF-8")));
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

/**

```

```

* 敏感信息解密
*
* @param encryptData 密文
* @param key          密钥
* @return
* @throws Exception
*/
public static String decryptMessageBySM4(String encryptData, String key) {
    try {
        SecretKeySpec secretKey = getKey(key);
        Cipher cipher = Cipher.getInstance(ALGORITHM_NAME_ECB_PADDING);
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        return new String(cipher.doFinal(Base64.getDecoder().decode(encryptData)));
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

/**
* 加密文件
*
* @param filePath 明文文件路径
* @param encFilePath 加密文件路径
* @param sm4key    密钥
* @return
*/
public static boolean encryptFileBySM4(String filePath, String encFilePath, String sm4key) {
    try (FileInputStream is = new FileInputStream(filePath);
        FileOutputStream out = new FileOutputStream(mkdirFiles(encFilePath))) {
        SecretKeySpec secretKey = getKey(sm4key);
        Cipher cipher = Cipher.getInstance(ALGORITHM_NAME_ECB_PADDING);
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        CipherInputStream cis = new CipherInputStream(is, cipher);
        byte[] buffer = new byte[1024];
        int r;
        while ((r = cis.read(buffer)) > 0) {
            out.write(buffer, 0, r);
        }
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

```

    }

    /**
     * 解密文件
     *
     * @param filePath 明文文件路径
     * @param encFilePath 加密文件路径
     * @param sm4key 密钥
     * @return
     */
    public static boolean decryptFileBySM4(String filePath, String decFilePath, String sm4key) {
        try (FileInputStream is = new FileInputStream(filePath);
            FileOutputStream out = new FileOutputStream(mkdirFiles(decFilePath))) {
            SecretKeySpec secretKey = getKey(sm4key);
            Cipher cipher = Cipher.getInstance(ALGORITHM_NAME_ECB_PADDING);
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            CipherOutputStream cos = new CipherOutputStream(out, cipher);
            byte[] buffer = new byte[1024];
            int r;
            while ((r = is.read(buffer)) > 0) {
                cos.write(buffer, 0, r);
            }
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }

    /** 根据 filePath 创建相应的目录
     * @param filePath 要创建的文件路径
     * @return file 文件
     * @throws IOException io 出现异常
     */
    private static File mkdirFiles(String filePath) throws IOException {
        File file = new File(filePath);
        if (!file.getParentFile().exists()) {
            file.getParentFile().mkdirs();
        }
        file.createNewFile();
        return file;
    }

    private static SecretKeySpec getKey(String myKey) throws Exception {

```

```

        byte[] key = myKey.getBytes(StandardCharsets.UTF_8);
        MessageDigest sha = MessageDigest.getInstance("SHA-1");
        key = sha.digest(key);
        key = Arrays.copyOf(key, 16);
        return new SecretKeySpec(key, ALGORITHM_NAME);
    }

    /**
     * 转 16 进制
     * @param bytes
     * @return
     */
    private static String byteToHexString(byte[] bytes) {
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < bytes.length; i++) {
            String strHex = Integer.toHexString(bytes[i]);
            if (strHex.length() > 3) {
                sb.append(strHex.substring(6));
            } else {
                if (strHex.length() < 2) {
                    sb.append("0" + strHex);
                } else {
                    sb.append(strHex);
                }
            }
        }
        return sb.toString();
    }
}

```

附录四：SM2 签名验签方法

```

import org.apache.commons.codec.binary.Base64;
import org.bouncycastle.asn1.gm.GMObjectIdentifiers;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.util.Base64Utils;

import java.nio.charset.StandardCharsets;
import java.security.*;

```

```

import java.security.spec.ECGenParameterSpec;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;
import java.util.HashMap;
import java.util.Map;

public class SM2Util {

    private static final String PRIVATE_KEY = "privateKey";
    private static final String PUBLIC_KEY = "publicKey";
    private static final String STD_NAME = "sm2p256v1";
    private static final String ALGORITHM = "EC";

    public static final Logger LOGGER = LoggerFactory.getLogger(SM2Util.class);

    /**
     * 生成密钥
     * @return
     */
    public static Map<String, String> generate() {
        Map<String, String> keyMap = new HashMap<String, String>();
        try {
            ECGenParameterSpec sm2Spec = new ECGenParameterSpec(STD_NAME);
            KeyPairGenerator kpg = KeyPairGenerator.getInstance(ALGORITHM, new
BouncyCastleProvider());
            kpg.initialize(sm2Spec);
            KeyPair keyPair = kpg.generateKeyPair();
            PublicKey publicKey = keyPair.getPublic();
            PrivateKey privateKey = keyPair.getPrivate();
            keyMap.put(PRIVATE_KEY, Base64Utils.encodeToString(privateKey.getEncoded()));
            keyMap.put(PUBLIC_KEY, Base64Utils.encodeToString(publicKey.getEncoded()));
        } catch (Exception e) {
            LOGGER.error(e.getMessage());
        }
        return keyMap;
    }

    /**
     * 加签
     * @param value 明文
     * @param privateKeyB64 私钥(base64 格式)
     * @return
     */
    private static String sign(String value, String privateKeyB64) {

```

```

String result = "";
try {
    Signature signature = Signature.getInstance(GMObjectIdentifiers.sm2sign_with_sm3.toString(),
new BouncyCastleProvider());
    KeyFactory keyFactory = KeyFactory.getInstance(ALGORITHM, new
BouncyCastleProvider());
    PrivateKey privateKey = keyFactory.generatePrivate(new
PKCS8EncodedKeySpec(Base64Utils.decodeFromString(privateKeyB64)));
    signature.initSign(privateKey);
    signature.update(value.getBytes(StandardCharsets.UTF_8));
    byte[] signed = signature.sign();
    result = Base64Utils.encodeToString(signed);
} catch (Exception e) {
    LOGGER.error(e.getMessage());
}
return result;
}

/**
 * 验签
 * @param value 明文
 * @param sign 签名
 * @param publicKeyB64 公钥(base64 格式)
 * @return
 */
private static boolean verify(String value, String sign, String publicKeyB64) {
    boolean result = false;
    try {
        Signature signature = Signature.getInstance(GMObjectIdentifiers.sm2sign_with_sm3.toString(),
new BouncyCastleProvider());
        KeyFactory keyFactory = KeyFactory.getInstance(ALGORITHM, new
BouncyCastleProvider());
        PublicKey publicKey = keyFactory.generatePublic(new
X509EncodedKeySpec(Base64Utils.decodeFromString(publicKeyB64)));
        signature.initVerify(publicKey);
        signature.update(value.getBytes(StandardCharsets.UTF_8));
        result = signature.verify(Base64.decodeBase64(sign.getBytes()));
    } catch (Exception e) {
        LOGGER.error(e.getMessage());
    }
    return result;
}
}

```