

联合登录接口文档1.7

关于联合登录合作的说明

联合登录接口说明

接口

1. 免登录接口
2. 银行上下线列表接口
3. 用户订单查询接口

附录

PHP相关方法参考

访问参数签名算法:

加密算法

测试环境说明

测试环境的账号和密钥

涉及加密数据处理的实例参考

1. 免登录接口参数处理过程
2. 用户订单查询接口参数处理过程

接口提供方用户申卡状态与描述之间的对应关系说明

其他地址参考

版本修改说明

1.7

联合登录接口文档1.7

关于联合登录合作的说明

【1】.联合登录即当用户在第三方平台登录后，免去在我方登录的流程，因此需第三方在用户点击我方页面时先进行登录校验，若该用户未登录，则应调起第三方登录页面。

【2】.联合登录调用接口一方需要有人注册成为合伙人，之后的平台可以使用带有其标记的链接，此处为绑定关系流程，若绑定错误或未能绑定，后期则无法结算，第三方自行承担。

【3】.暂时只开放办卡、办卡进度查询、银行上下线及订单状态接口，若有其他需求请联系。

【4】.为安全考虑，需第三方给出用户每天到我们页面的UV（调用接口次数）上限，提供一个数值，每自然日超过该上限，则会在进入我方页面时调起我们的登录页面。当第三方预测其量有大幅度增加超过之前所提供的上限时，请及时联系更改上限数值。

【5】.双方的结算方式按照卡社区的结算方式进行，不进行特殊处理，若第三方平台有自己内部结算方式请自行开发。

【6】.银行会不定期进行上下线，请第三方自行规定调用api/hrcooperation/supportbank接口频率，若第三方未能及时进行调用该接口，则导致的订单无效问题由第三方自行承担；对于单银行链接，由于银行及其他原因下线，目前会自动跳转到交通银行进行兜底，若需其他方案则第三方自己制定兜底策略。

【7】.订单状态由订单接口返回，需要第三方自行拉取，接口提供方不会主动回传。

联合登录接口说明

请求域基址：<https://m.hhrcard.com>

实际接口请求地址：{请求域基址}+{接口名称}

测试环境：请联系卡社区技术支持人员

接口

1. 免登录接口

接口名称：api/hrcooperation/login

请求协议：HTTPS

请求方式：POST

请求所需参数：

参数	类型	长度	是否必选	备注	示例
mobile	string	11	是	用户手机号（加密后）	例如：13041244499 加密后：uEgtjmbpuOEksJZZ2rt+nw==
tid	string	4	是	第三方平台id，对接后由接口提供方分配	例如：50（测试可使用）
tuid	string	32	是	用户uid，第三方平台的uid	例如：330033
timestamp	int	10	是	当前时间戳	例如：1542944265
token	string	32	是	签名参数，由接口提供方约定算法生成。第三方对接需要自己实现。见附件签名算法	例如：A967189BB9D7F25D7FABDEE4BCE13775

实际请求方式

请求参数json后的字符串转换为十六进制值，并使用针对url的base64编码后传输

实际请求参数

参数	说明	示例
data	请求参数经以上方式整理后的字符串	

返回参数

参数	说明
error	接口返回状态，0 成功 其他值失败 具体返回码见附件
msg	接口返回信息
data	免登录地址

返回示例

```
1 {
2     "error":0,
3     "msg":"success",
4     "data": {
5
6         "redirect_url":"https%3A%2F%2Fm.hhrcard.com%2Fhhrcoperation%2Faccess%3Faccess%3D16f64569b4537ffc40e89095bf1970b9%26tid%3D50"
7         //urlencode
8     }
9 }
```

登录方式说明

在接口正确返回的情况下，跳转至接口返回地址完成用户的免登录访问，测试期间接口免登录地址有效期为1天

(目前该免登陆地址验证通过后将跳转至卡社区平台的全银行列表页)

免登陆地址支持自定义参数说明

参数	说明	示例
inviteCode	邀请码【接口提供方平台新户将绑定为该邀请码用户的下级】	30005707
bankid	银行的id（请使用接口2银行上下线列表来获取）【使用该自定义参数，免登录后会跳转至相应银行的卡列表页】	7（交通银行）

2. 银行上下线列表接口

接口名称：api/hhrcoperation/supportbank

请求协议：HTTPS

请求方式：POST

请求所需参数：

参数	类型	长度	是否必选	备注	示例
tid	string	4	是	第三方平台id，对接后由接口提供方分配	例如：50（测试可使用）
timestamp	int	10	是	当前时间戳	例如：1542944265
token	string	32	是	签名参数，由接口提供方约定算法生成。第三方对接需要自己实现。见附件签名算法	例如：A967189BB9D7F25D7FABDEE4BCE13775

实际请求方式

请求参数json后的字符串转换为十六进制值，并使用针对url的base64编码后传输

实际请求参数

参数	说明	示例
data	请求参数经以上方式整理后的字符串	

返回参数

参数	说明
error	接口返回状态，0 成功 其他值失败 具体返回码见附件
msg	接口返回信息
data	银行上下线列表

返回示例

```
1 {
2   "error":0,
3   "msg":"success",
4   "data": {
5     // 上线的银行列表
6     "online": [
7       {
8         "id": "13",
9         "name": "兴业银行"
10      },
11      {
12        "id": "10",
13        "name": "平安银行"
14      },
15      ...
16    ],
17    // 下线的银行列表
18    "offline": [
19      {
20        "id": "16",
21        "name": "中信银行"
22      }
23    ]
24  }
25 }
```

3. 用户订单查询接口

接口名称: api/hhrcooperation/order

请求协议: HTTPS

请求方式: POST

说明:

该接口提供给合作方用来查询第三方用户在平台的申卡订单数据以及状态，订单数据分页返回，按时间倒叙排序。支持批量的用户查询，建议批量查询的用户数量在100以内

默认只提供1个月（接口提供方可能会做调整）内的订单数据拉取，历史数据的获取请使用历史订单数据接口

请求所需参数:

参数	类型	长度	是否必选	备注	示例
tid	string	4	是	第三方平台id, 对接后由接口提供方分配	例如: 50 (测试可使用)
tuid	array		是	值为第三方平台的用户uid, string类型	例如: ["330033", "330034", "330035"]
timestamp	int	10	是	当前时间戳	例如: 1542944265
page	int	10	是	页码	例如: 1
pagesize	int	10	是	数据分页大小 (支持100-1000)	例如: 500
earliest	int	10	否	返回数据的最早时间戳, 默认返回1个月之内的数据	例如: 1543907735
token	string	32	是	以上参数的签名, 由接口提供方约定算法生成。第三方对接需要自己实现。见附件签名算法	例如: A967189BB9D7F25D7FABDEE 4BCE13775

实际请求方式

请求参数json后的字符串转换为十六进制值，并使用针对url的base64编码后传输

实际请求参数

参数	说明	示例
data	请求参数经以上方式整理后的字符串	

返回参数

参数	说明
error	接口返回状态, 0 成功 其他值失败 具体返回码见附件
msg	接口返回信息
data	正确返回情况下包含用户的订单信息 查看订单接口返回的data字段说明

返回示例

```
1 {
2   "error": 0,
3   "msg": "success",
4   "data": {
5     "orders": [ // 订单数据
6       {
7         "order_id": 127019308,
8         "tuid": "10000",
9         "suid": 32921701,
10        "bank_id": 13,
11        "card_name": "兴业万事达欧冠白金卡",
12        "price": 82,
13        "apply_time": 1545062400,
14        "apply_status": 6,
15        "desc": "审批成功, 等待激活"
16      },
17      {
18        "order_id": 127019310,
19        "tuid": "10000",
20        "suid": 32921701,
21        "bank_id": 3,
22        "card_name": "光大招财猫主题信用卡",
23        "price": 1010,
24        "apply_time": 1545062400,
25        "apply_status": 4,
26        "desc": "非首次申请"
27      },
28      ....
29    ],
30    "page": 1 // 当前接口返回数据的页码
31  }
32 }
```

用户订单接口返回参数data中的字段说明:

字段	说明
order_id	订单唯一ID
orders	当前页码的订单列表
tuid	第三方请求的用户ID
suid	该tuid在接口提供方平台的上级ID, 等于零的时候表示该tuid没有上级
bank_id	接口提供方定义的银行ID
card_name	用户申请的信用卡中文名称
price	用户申请卡片的结算价
apply_time	用户申请卡片的时间
apply_status	用户当前申请的信用卡状态 点击查看信用卡申请状态与描述对应的说明
desc	接口提供方对当前申请卡片的状态描述
page	当前页码

附录

PHP相关方法参考

PHP二进制字符串转十六进制函数: `bin2hex`

针对url的base64加密与解密:

```
1 function base64_url_encode($data) {
2   return strtr(base64_encode($data), '+/=', '-_');
```

```

3 }
4 function base64_url_decode($data) {
5     return base64_decode(strtr($data, '-_', '+/'));
6 }

```

访问参数签名算法：

```

1 <?php
2
3 /**
4  * 参数签名
5  */
6 class Signature {
7     const SALT = 'ogbdgejg'; // 测试使用
8
9     public static function sign($args) {
10         if (!$args) return '';
11         $salt = self::SALT;
12         ksort($args);
13         $signStr = '';
14         foreach ($args as $k => $v) {
15             if ($v || is_numeric($v)) {
16                 if (is_array($v)) {
17                     $v = json_encode(array_map('strval', $v));
18                 }
19                 $signStr .= $k.$v;
20             }
21         }
22         return strtoupper(md5(md5($signStr) . $salt));
23     }
24 }

```

加密算法

```

1 <?php
2
3 /**
4  * 加密算法
5  */
6 class HhrCrypt {
7     const KEY = 'UA3MzZiMTgwZDB2B'; // 测试使用
8     const IV = 'iqZVGCCFVMkjMYUE'; // 测试使用
9
10     public static function encrypt($content)
11     {
12         $key = self::KEY;
13         $content = $content . "\0\0\0\0\0";
14         $iv = self::IV;
15         $data = openssl_encrypt($content, 'aes-128-cbc', $key, OPENSSL_ZERO_PADDING, $iv);
16         $error = openssl_error_string();
17         if($error) {
18             return false;
19         }
20         return $data;
21     }
22
23     public static function decrypt($encryData){
24         $key = self::KEY;
25         $iv = self::IV;
26         $data = openssl_decrypt($encryData, 'aes-128-cbc', $key, OPENSSL_ZERO_PADDING, $iv);
27         $data=substr($data,0,11);
28         return $data;
29     }
30 }

```

测试环境说明

1. 绑HOST的方式：将访问基址m.hhrcard.com绑定到IP：124.251.102.83
2. 或使用我们的测试域名：测试调试时将访问基址m.hhrcard.com 替换成 test.hhrcard.com

测试环境的账号和秘钥

1. 请求参数

```
1 tid: 50
```

2. 加密算法

```
1 KEY: UA3MzZiMTgwZDB2B
2 IV: iqZVGCCFVMkjMYUE
```

3. 签名算法

```
1 SALT: ogbdgejg
```

涉及加密数据处理的实例参考

1. 免登录接口参数处理过程

1. 手机号加密 [查看加密方法](#)

手机号 13141240049 加密后 UGegpugfbpVngifmjf7jog==

2. 整理初始参数:

```
1 array (
2   'mobile' => 'UGegpugfbpVngifmf7jog==',
3   'tid' => 50,
4   'tuid' => '333030303',
5   'timestamp' => 1545624974,
6 )
```

3. 参数签名 [查看签名方法](#)

- 签名前拼接的字符串: `mobileUGegpugfbpVngifmjf7jog==tid50timestamp1545624974tuid33303303`
- 签名后的 `token`: `92B2227318BE8B524260453F2047D058`
- 签名后的参数为:

```
1 array (  
2   'mobile' => 'UGegpugfbpVngifmjf7jog=',  
3   'tid' => 50,  
4   'tuid' => '3330330303',  
5   'timestamp' => 1545624974,  
6   'token' => '92B2227318BE8B524260453F2047D058',  
7 )
```

4. 对签名后参数数组转json

```
1 {"mobile": "UGegpugfbpVngifmjf7jog==", "tid": 50, "tuid": "333030303", "timestamp": 1545624974, "token": "92B2227318BE8B524260453F2047D058"}
```

5. 转16进制字符串

1 7b226d6f62696c65223a225547656f7707567666270566e6769666d6a66376a6f673d3d222c22746964223a35302c2274756964223a223333333033303330333033222c2274696d657374616d70223a313534353632343937342c22746f6b656e223a2239324232323237333138424538423532343236303435333463230343744303538222d

6. 最后使用对url的base64方法

[illegible]

2. 用户订单查询接口参数处理过程

1. 整理初始参数

```
1 array (
2   'tid' => 50,
3   'tuid' => array ( "330033", "330034", "330035"),
4   'page' => 1,
5   'pagesize' => 100,
6   'timestamp' => 1545624160,
7 )
```

2. 参数签名查看签名方法:

- 签名前拼接的字符串: `page1pagesize100tid50timestamp1545624160tuid["330033","330034","330035"]`
- 签名后的 `token`: `0735957A9444A51563ACB9A95EFEE650`
- 签名后的参数为:

```
1 array (
2   'tid' => 50,
3   'tuid' => array ( "330033", "330034", "330035"),
4   'page' => 1,
5   'pagesize' => 100,
6   'timestamp' => 1545624160,
7   'token' => '0735957A9444A51563ACB9A95EFEE650',
8 )
```

3. 对签名后参数转json:

```
1 {"tid":50,"tuid":  
2 ["330033","330034","330035"],"page":1,"pagesize":100,"timestamp":1545624160,"token":"0735957A9444A51563ACB9A95FEE650"}  
// 注意tuid的格式
```

4. 转16进制字符串

```
1 7b22746964223a35302c2274756964223a5b2233333030333322c223333030333422c2233333030333525d2c2270616765223a312c22706167657369
  7a65223a3130302c2274696d657374616d70223a313534353632343136302c22746f6b656e223a22307335393537413934343441353135363341434239
  41393545464545363530227d
```

5. 最后使用对url的base64方法

[illegible]

请参考以上处理过程，检查每一步的处理结果是否一致

接口提供方用户申卡状态与描述之间的对应关系说明

状态 (status)	描述 (desc)	说明
2	待查询	用户从平台进件, 但是暂未有从银行的进度查询信息
3	查询完成, 等待审核	银行已经审批, 等待平台的进一步审核确认, 平台审核完成后状态会变更为已结算
3	审批成功, 等待结算	银行已经审批, 等待平台的奖励结算
4	已结算	银行已经审批, 平台的奖励已结算完成
4	非首次申请	银行已审批, 用户之前已有该银行的申卡成功记录
4	非本渠道申请	用户信用卡不是在本平台进的件
5	已失败	银行审批拒绝
6	审批成功, 等待激活	银行已审批, 平台已结算, 等待用户进一步的激活
7	待再查	银行审批中, 待用户进一步的进度查询

其他地址参考

1. 进度查询支持的银行列表 (包含所有支持的银行进度查询)

<http://m.hhrcard.com/coprogress/banklist?coopsource=2>

版本修改说明

1.7

- 1. [增加接口参数处理参考](#)
- 2. [说明文档更新](#)