

安智 App 接入

服务端

**SDK 接口开发参考说明书**

Version 3.2

2015-01-06

|                         |                             |     |     |          |
|-------------------------|-----------------------------|-----|-----|----------|
|                         | 安智 App 接入_服务端 SDK 接口开发参考说明书 |     |     |          |
| 文档名称                    |                             |     |     |          |
| 作 者                     |                             |     |     |          |
| 类 别                     |                             |     |     |          |
| 子 类 别                   |                             |     |     |          |
| 文件名称                    |                             |     |     |          |
| 摘 要                     |                             |     |     |          |
| 修订历史 (REVISION HISTORY) |                             |     |     |          |
| 版本号                     | 日期                          | 修改人 | 审阅人 | 摘 要      |
| 1.0                     | 2013-07-02                  |     |     | 创建       |
| 3.2                     | 2015-01-06                  |     |     | 修改礼券回调信息 |
|                         |                             |     |     |          |
|                         |                             |     |     |          |
|                         |                             |     |     |          |
|                         |                             |     |     |          |
|                         |                             |     |     |          |

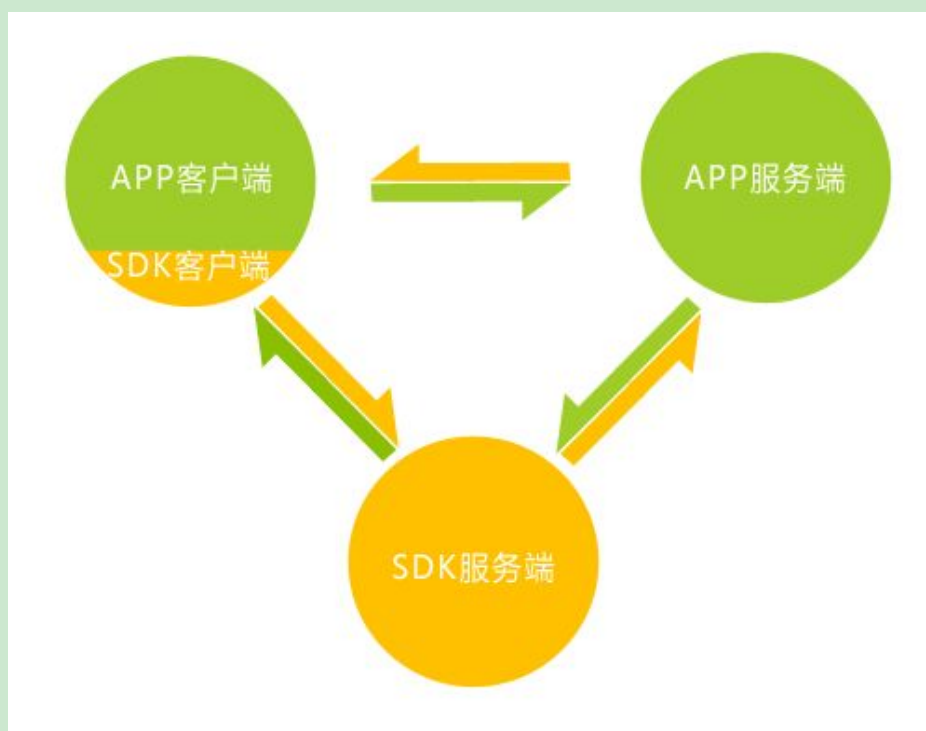
## 目录

|                            |    |
|----------------------------|----|
| 安智 App 接入.....             | 1  |
| 服务端.....                   | 1  |
| SDK 接口开发参考说明书.....         | 1  |
| 1. 接口应用场景.....             | 3  |
| 2. 整体流程图示.....             | 3  |
| 3. 接口通信方式.....             | 3  |
| 4. HTTP 通信接口.....          | 4  |
| 1) 登录(注册)回调 App 服务端接口..... | 4  |
| 2) 退出回调 App 服务端接口.....     | 5  |
| 3) 用户在线会话验证.....           | 6  |
| 4) 刷新用户在线时长.....           | 7  |
| 5) App 官方账号登录.....         | 8  |
| 6) 支付订单回调 APP 服务端接口.....   | 11 |
| 7) 查询订单.....               | 13 |
| 5 规范.....                  | 15 |
| 5.1 错误码返回格式: .....         | 15 |

## 1. 接口应用场景

App 接入业务需要与安智 SDK 服务端进行通信，完成业务逻辑。接口中主要名词解释请参考《安智 App 接入 开发文档参考说明书—总体介绍》名词解释一节。接口应用主要在 SDK 服务端与 App 服务端直接的通信和交互。

## 2. 整体流程图示



### 3. 接口通信方式

SDK 服务端与第三方服务端通过 HTTP POST 方式进行数据的交互。数据格式为 JSON，为保证信息通信安全，采用 Base64，加 URL 消息签名方式。

### 4. HTTP 通信接口

#### 1) 登录(注册)回调 App 服务端接口

| 接口              | 描述  |   |
|-----------------|---|---|
| 接口名称            | 登录（注册）回调 App 服务端接口（SDK 服务端→App 服务端）   |   |
| 接口说明            | 用于用户登录或者注册回调 App 服务端，回调地址需在应用申请时配置  |   |
| SDK 服务端<br>请求格式 | 完整消息  | <pre>msg= {   "uid":"xxx",   "nickName":"xxx",   "type":"0",   "time":"20130228101059123" } ext= {   "gameArea" : "xxx",   "userRole" : "xxx",   "gameLevel" : "xxx",   "memo" : "xxx",   "time" : "20130228101059123" } action=login</pre> |
| App 服务端<br>响应格式 | 完整消息  | 直接返回字符串 "success"   |
| 请求格式说明          | 1.msg->uid 安智账号 id<br>2. msg->nickname 用户昵称<br>3. msg->type 登录类型：0 普通登录，1 注册后登录 |   |

|        |  |
|--------|--|
|        | 4.msg->time 响应时间 yyyyMMddHHmmssSSS 精确到毫秒<br>例如: 20130703103856771<br>【App 类型为游戏】额外返回 Ext 参数<br>5.ext->gameArea 游戏分区<br>6. ext->userRole 用户角色<br>7. ext->gameLevel 游戏等级<br>8. ext->memo 备注<br>9. ext->time 响应时间 yyyyMMddHHmmssSSS 精确到毫秒<br>10.action 回调类型, login 为登录回调, logout 为退出回调  |
| 响应格式说明 |  |
| 备注     | 回调参数中 msg、ext 中的内容均用 Base64. decode ()解密后得到对应 json, 若用户存在用户游戏信息则返回 ext 参数, 反之则无。   |
| 请求样例   | msg=eydlawQnOicyMDEzMDcwODE4MjgzOWxzdlkyYmJsbnInLCd0aW1lJzonMjAxMzA3MDkxNDAwMzA4MTInLCduaWNrTmFtZSc6JycsJ3R5cGUoOicwJ30=<br>action=login<br>其中 msg 节点通过解密后的明文为<br>{<br>'uid': '201307081828391YvY2bblnb',<br>'time': '20130709140030812',<br>'nickName': '',<br>'type': '0'<br>}<br> |
| 返回样例   | success  |

## 2) 退出回调 App 服务端接口

| 接口              | 描述                                     |  |
|-----------------|--|--|
| 接口名称            | 退出回调 App 服务端 (SDK 服务端→ App 服务端)        |  |
| 接口说明            | 用于用户退出回调 App 服务端, 回调地址需在应用申请时配置        |  |
| SDK 服务端<br>请求格式 | 完整消息                                   | msg=<br>{<br>“uid” :” xxx” ,<br>“time” :” 20130228101059123”<br>}<br>action=logout |
| App 服务端<br>响应格式 | 完成消息                                   | 直接返回字符串 “success”  |
| 请求格式说明          | 1.msg->uid 安智账号 id<br>2.msg->time 响应时间 |  |

|        |  |
|--------|--|
|        | 3.action 回调类型, login 为登录回调, logout 为退出回调   |
| 响应格式说明 | 直接返回字符串 “success”  |
| 备注     | 回调参数中 msg 中的内容均用 Base64. decode ()解密后得到对应 json。  |
| 请求样例   | <pre>msg=eydlaWQnOicyMDEzMDcwODE4MjgzOWxzdlkyYmJsbnInLCd0aW1lJzonMjAxMzA3MDkxNDEyNDA5MzEnfQ==</pre> <pre>action=logout</pre> <p>其中 msg 节点通过解密后的明文为</p> <pre>{   'uid': '201307081828391YvY2bblnb',   'time': '20130709141240931' }</pre> |
| 返回样例   | success  |

### 3) 用户在线会话验证

| 接口              | 描述  |   |
|-----------------|---|---|
| 接口名称            | 用户在线会话验证 (App 服务端→ SDK 服务端)   |   |
| 接口说明            | 用于实时查询用户在线情况, 请求方式为 POST, 把所有数据, 包括 sign 用 key=value 并用&符号链接的方式发送, 请求地址为<br><a href="http://user.anzhi.com/web/api/sdk/third/1/quervislogin">http://user.anzhi.com/web/api/sdk/third/1/quervislogin</a> |   |
| App 服务端<br>请求格式 | 完整消息  | <pre>time=20130228101059123&amp; appkey=fds12121&amp; sid=xxxxxx&amp; sign=xxxx</pre>                                 |
| SDK 服务端<br>响应格式 | 完整消息  | <pre>{   "sc": "1",   "st": "成功(sid 有效)",   "time": "20130228101059123"   "msg": {     "uid": "123456789"   } }</pre> |
| 请求格式说明          | 1. time 请求时间 yyyyMMddHHmmssSSS 精确到毫秒<br>例如: 20130703103856771<br>2. appkey 应用 key<br>3. sid 当前登录用户会话 id<br>4. sign 签名串, 格式<br>Base64.encodeToString (appkey+sid+appsecret);                             |   |
| 响应格式说明          | 1. sc 状态码<br>2. st 状态描述   |   |

|      |  |
|------|--|
|      | 3. time 响应时间<br>4. msg->nickname 用户昵称<br>5. msg->uid 安智用户 id   |
| 备注   | 回调参数中 msg 中的内容均用 Base64.Decode () 解密后得到对应 json。  |
| 请求样例 | 请求参数 (post提交)<br>appkey = "c318br6RLex12IeBs0Ta6wo1";<br>sid = "MjAxMzA3MDgxODI4MzlsWXZZMmJi<br>bG5iXzEzNzMzNTE5OTJfMQ==";<br>sign = Base64.encodeToString(appkey+sid+appsecret);<br>time = "20130709161023120";           |
| 返回样例 | <pre>{   'time': '20130709150615195',   'msg': 'eydl aWQnOicyMDEzMDcwODE4MjgzOWxZdlkyYm JsbmInfQ==',   'sc': '1',   'st': '成功(sid 有效)'}</pre> <p>其中 msg 节点通过解密后的明文为</p> <pre>{   'uid': '20130708182839IYvY2bblnb' }</pre> |

#### 4) 刷新用户在线时长（暂时停用）

| 接口              | 描述   |  |
|-----------------|--|--|
| 接口名称            | 刷新用户在线时长 (App 服务端→ SDK 服务端)  |  |
| 接口说明            | 用于刷新用户在线的时间,生成返回新的会话 sid,请求方式为 POST,<br>把所有数据, 包括 sign 用 key=value 并用&符号链接的方式发送,<br>请求地址为<br><a href="http://user.anzhi.com/web/api/sdk/third/1/refreshlogin">http://user.anzhi.com/web/api/sdk/third/1/refreshlogin</a> |  |
| SDK 服务端<br>请求格式 | 完整消息   | time=20130228101059123&<br>appkey=fds12121&<br>account=xxxxx&<br>sid=xxxxxx&<br>sign=xxxxx                   |
| App 服务端<br>响应格式 | 完整消息   | <pre>{   "st": "成功(sid 有效)",   "sc": "1",   "time": "20130228101059123",   "msg": {     "sid": "xxxx",</pre> |

|        |  |  |
|--------|--|--|
|        |  | <pre> “uid” :” 123456789” } } </pre>   |
| 请求格式说明 |  | 1.time 请求时间 yyyyMMddHHmmssSSS 精确到毫秒<br>例如: 20130703103856771<br>2.appkey 应用 key<br>3.account 用户登录账号<br>4.sid 当前登录用户会话 id<br>5.sign 签名串, 格式<br>Base64.encodeToString (appkey+account+sid+appsecret);  |
| 响应格式说明 |  | 1.sc 状态码<br>2.st 状态描述<br>3.time 响应时间<br>4.msg->sid 当前登录用户会话 id<br>5.msg->uid 安智用户 id   |
| 备注     |  | 回调参数中 msg 中的内容均用 Base64.Decode () 解密后得到对应 json。  |
| 请求样例   |  | 请求参数 (post 提交)<br>appkey = "c318br6RLex12IeBs0Ta6wo1";<br>account = "13051730720";<br>sid = " MjAxMzA3MDgxODI4MzlsWXZ<br>ZMmJibG5iXzEzNzMzNTM3MTVfMQ==";<br>sign = Base64.encodeToString(appkey+account+sid+appsecret);<br>time = "20130709161023120";   |
| 返回样例   |  | <pre> {   'time':'20130709151837128',   'msg':{'eyd1aWQnOicyMDEzMDcwODE4MjgzOWxZdlkyYmJsbmInLC dzaWQnOidNakF4TXpBM01EZ3hPREk0TXpscldYWlpNbUppYkc1aV h6RXpOek16TIRRek1UZGZNUT09J30='},   'sc':'1',   'st':'成功(sid 有效)' } </pre> <p>其中 msg 节点通过解密后的明文为</p> <pre> msg: {   'uid':'20130708182839IYvY2bblnb',   'sid':'MjAxMzA3MDgxODI4MzlsWXZMmJibG5iXzEzNzMzNTQzMTdfMQ==' } </pre> |

## 5) App 官方账号登录

| 接口   | 描述                             |
|------|--------------------------------|
| 接口名称 | App 官方账号登录 (App 服务端 → SDK 服务端) |



|                 |   |   |
|-----------------|---|---|
| 接口说明            | 用于 App 官方账号登录,用户采用“App 老账号(App 官方账号)”方式登录过程,“App 服务器”完成账号登录验证后,调用本接口,通知“SDK 服务器”同步账号的登录状态。请求地址为<br>http://user.anzhi.com/web/api/sdk/1/user-create-bind   |   |
| App 服务端<br>请求格式 | 完整消息  | <pre> 'head':{   'appkey':'c318br6RLex12IeBs0Ta6wo1',   'version':'1.0',   'time':'201305291036036' }, 'body':{   'msg':{     'gameUser': '{       "id":1,       "loginname":"aaa"     }',     'time':'201305291036036'   },   'ext': {} } </pre> |
| SDK 服务端<br>响应格式 | 完整消息  | <pre> {   'time':'20130529115412431',   'sign':'',   'msg':'',   'sc':'200',   'st':'登录成功!' } </pre> <p>其中 msg, sign 的内容为加密串, 明文格式</p> <pre> {   'uid':'',   'sid':'' } </pre>  |
| 请求格式说明          | 1.head ->time 请求时间 yyyyMMddHHmmssSSS 精确到毫秒<br>例如: 20130703103856771<br>2.head->appkey 应用 key<br>3.head->version 版本号, 默认 1.0,<br>4.body->msg->gameUser->id 游戏账号 id<br>5.body->msg->gameUser->loginname 游戏账号<br>6. body->msg->time 请求时间 yyyyMMddHHmmssSSS 精确到毫秒<br>例如: 20130703103856771<br>7. body->ext 扩展信息 |   |
| 响应格式说明          | 1.sc 状态码<br>2.st 状态描述<br>3.time 响应时间<br>4.msg 响应数据解密用, 加解密 规则 Base64(3DES(msg)).  |   |

|      |  |
|------|--|
|      | <p>采用服务端 SDK 提供的类库. Des3Util.decrypt() 即可得到 json.</p> <p>5.sign 签名串采用 Des3Util.getSigned() 的签名串,可自行采用该方法加密响应数据进行验签</p>   |
| 备注   | <p>1、请求内容需加密后以输出流的形式 post 请求提交至 sdk 服务端并在请求对象 request 对象 header 中设置 appKey, 如: request.setHeader(“appKey”, “xxxx”);</p> <p>2、“App 服务器”在成功调用该接口后,从“SDK 服务器”获得了 App 老账号(App 官方账号)对应的安智账号,必须进行映射绑定,以能够将使用这两个账号登录的用户识别为同一个用户。</p>  |
| 请求样例 | <pre> 'head':{   'appkey':'c318br6RLex12IeBs0Ta6wo1',   'version':'1.0',   'time':'20130704150954076' }, 'body':{   'msg':{     'gameUser': '{       "id":3,       "loginname":"test123"     }',     'time':'20130704150954076'   },   'ext': {} } </pre>  |
| 返回样例 | <pre> {   'time':'20130704151000076',   'msg': 'isUey0AICaTwT7TGmuEvT0nRstvLN plve9lrt+c7dac5vqjwIfgUakqelrtJV9s0 5xHdpmdjp//9\r\n2AVVPyV+1xmbYY0amur oBSvZqNBhYRDSHAr8X9MKD1zVNGuc0egL',   'st': '注册成功!',   'sign': 'dcTT4dJTGP0zJ0+T2HiJng==',   'sc': '200' } </pre> <p>其中 msg 节点通过解密后的明文为</p> <pre> {   'uid': '201307041509557R7Xy5sx03',   'sid': 'MjAxMzA3MDQxNTA5NTU3UjdYeT VzeDAzXzEzNzI5MjE3OTlfMQ==' } </pre> |

## 6) 支付订单回调 APP 服务端接口

| 接口              |   | 描述   |
|-----------------|---|--|
| 接口名称            | 支付订单回调 App 服务端 (SDK 服务端 → App 服务端)  |  |
| 接口说明            | 用于支付订单回调 App 服务端，回调地址需在应用申请时配置  |  |
| SDK 服务端<br>请求格式 | 完整消息  | <pre>data = {   "uid": "xxx",   "orderId": "",   "orderAmount": "",   "orderTime": "",   "orderAccount": "",   "code": "",   "msg": "",   "payAmount": "",   "cpInfo": "",   "notifyTime": "",   "memo": "",   "redBagMoney": "" }</pre> |
| App 服务端<br>响应格式 | 完整消息  | “success”  |
| 请求格式说明          | <p>1. uid 安智账号 id <b>uid 在匿名支付的时候为空</b></p> <p>2. orderId 订单号</p> <p>3. orderAmount 实际付费金额(单位为分)</p> <p><b>订单回调通知发送到 CP 服务端时应以此作为实际购买道具的金额，订单金额可能大于实际的付费金额，多出道具金额的部分会转化为安智币保存在用户账户内。在使用充值卡，短代付费时订单金额 (payAmount) 与实际付费金额 (orderAmount) 可能存在差异，如道具 30 (orderAmount) 元，用户使用充值卡 100 (payAmount) 元进行付费，剩余的 70 元则转化为安智币，最终通知 CP 时包含 orderAmount (30 元) 与 payAmount (100 元)，CP 方则使用 orderAmount (30 元) 进行对账和发放道具。</b></p> <p><b>{payAmount:10000, orderAmount:3000}</b></p> <p>4. orderTime 支付时间</p> <p>5. orderAccount 支付账号</p> <p>6. code 订单状态 (1 为成功)</p> <p>7. payAmount 订单金额(单位为分)</p> <p><b>参数说明见 orderAmount</b></p> <p>8. cpInfo 回调信息 <b>用户自定义参数，此参数可作为厂商判别用户充值的身份标记(如保存用户账号 ID 和所在区分等身份信息)或其他用处。</b></p> |  |

|               |   |
|---------------|---|
|               | <p><b>注意：（一）</b> 若用户以此参数为自己内部自定义订单号，只作为内部参考，将不作为用户充值订单唯一性依据。</p> <p><b>（二）</b> 若保存了用户自定义订单号，在用户连续充值的时候，将与安智支付平台订单号保持一（厂商订单号）对多（安智订单号）的关系。厂商在判断用户是否充值时请依据安智平台发出的订单号作为唯一性条件,或者使用厂商+安智订单号组合进行唯一性验证。</p> <p><b>（三）</b> 账单核对一律以安智支付平台的订单号为依据。</p> <p><b>（四）</b> 部分充值渠道在充值完成后发送的通知可能会有延迟，安智支付平台依据第三方交易平台发出的交易通知作为充值成功与否的条件，此会影响对厂商的通知时间。</p> <p><b>（五）</b> 账单核对以 orderAmount 作为对账金额。</p> <p>9.notifyTime 通知时间</p> <p>10. memo 备注</p> <p>11.redBagMoney 礼券金额(单位为分)</p> <p><b>注意：（一）</b> 若用户没有使用礼券支付，通知没有该字段。</p> <p><b>（二）</b> 若用户使用包含礼券支付，金额是礼券金额。</p> <p>data 响应数据加解密 规则 Base64(3DES(data)).<br/>采用服务端 SDK 提供的类库. Des3Util.decrypt() 即可得到 json. （解密的 key 为接入应用的 appsecret）</p> |
| <b>响应格式说明</b> | App 服务端业务处理完成后 直接返回字符串 “success”  |
| <b>备注</b>     | <p>1、服务端数据解密的 key 均为 app 申请获得的 appsecret，直接将 demo 中的 key 换成 app 申请的 appsecret 即可，支付回调数据 php 解密算法在服务端文档中有示例。</p> <p>2、支付订单回调 APP 服务端接口中 uid 在匿名支付的时候为空，payAmount 用充值卡支付时可能为空。</p>  |
| <b>请求样例</b>   | <p>data=PUCJ03fzQZAZurLW/3phf5mHgFx6881ECDyJe3/1jmfLsxCZc5gX8dtEw0eRd+o+e0GLMarY+TjN5mwFY4S8aZZ5GFPxgbxj50LXZDuQ7sQIQF+y50Qg2wVpUYJWCJ3U7mkGwlvbht0v0dhIDZDGzEsgJuc8U4zrGbqy1v96YX8f0kLh7PVbbAnmDIniAVFinLsHcyyTHF+iT5+3798SoGwdsCGnoduJ12iTcccHQ0IkAA0gFDh1i763XAI0+vWcs4KXKfA5WJJT3C/n0ItAQIoq7a4qigKedGdN9fxJebllzfknVZe0TtMnEIqyjx1Er0xTLF04KTs=</p> <p>data 通过解密后的明文格式为</p> <pre>{   "payAmount": "10",   "uid": "201307081828391YvY2bblnb",   "notifyTime": 1373338174,   "cpInfo": "回调信息",   "memo": null,</pre>   |

|      |   |
|------|---|
|      | <pre> "orderAmount": "10", "orderAccount": "13051730720", "code": 1, "orderTime": "2013-07-09 10:47:00", "msg": "", "orderId": "20130709104714493" } </pre> |
| 返回样例 | success   |

## 7) 查询订单

| 接口              | 描述   |   |
|-----------------|--|---|
| 接口名称            | 查询订单 (App 服务端 → SDK 服务端)   |   |
| 接口说明            | 用于查询订单信息，请求方式为 POST，把所有数据，包括 sign 用 key=value 并用 & 符号链接的方式发送，请求地址为 <a href="http://pay.anzhi.com/web/api/third/1/queryorder">http://pay.anzhi.com/web/api/third/1/queryorder</a> |   |
| SDK 服务端<br>请求格式 | 完整消息   | <pre> time=20130228101059123&amp; appkey=fds12121&amp; type=0&amp; tradenum=xxxx&amp; mintradetime=xxxxxx&amp; maxtradetime=xxxxxx&amp; sign=xxxx </pre>  |
| App 服务端<br>响应格式 | 完整消息   | <pre> {   "st": "200",   "sc": "查询订单成功",   "total": "100",   "time": "20130228101059123",   "msg": "" } </pre> <p>其中 msg 节点的内容明文格式为</p> <pre> {   [     {       "tradenum": "xxxx",       "tradetime": "123",       "tradeamount": "12.0",       "tradestatus": xxx     }   ] } </pre> <p>.....//多条订单记录</p> |

|        |  |
|--------|--|
|        | <pre>     ]   } </pre>   |
| 请求格式说明 | <p>1. time 请求时间 yyyyMMddHHmmssSSS 精确到毫秒<br/>例如: 20130703103856771</p> <p>2. appkey 应用 key</p> <p>3. type0 web 端查询, 1sdk 端查询</p> <p>4. tradenum 交易订单号</p> <p>5. mintradetime 支付订单开始时间</p> <p>6. maxtradetime 支付订单结束时间</p> <p>7. sign 签名串采用<br/>Base64.encodeToString(appkey+tradenum+mintradetime+maxtradetime+appsecret)加密</p> |
| 响应格式说明 | <p>1. sc 状态码</p> <p>2. st 状态描述</p> <p>3. time 响应时间</p> <p>4. total 订单记录总计</p> <p>5. msg 订单数据加密串, 采用 Base64. decode ();解密即可得到 json</p> <p>6. msg-&gt;tradenum 交易订单号</p> <p>7. msg-&gt;tradetime 订单时间</p> <p>8. msg-&gt;tradeamount 订单金额</p> <p>9. msg-&gt;tradestatus 订单状态 1 为成功</p>  |
| 备注     | <p>1、请求参数中 tradenum 为一组, mintradetime、maxtradetime 为一组, 两组参数至少得传一组, mintradetime、maxtradetime 两个参数需两个同时有值或无值, 且有值时时间间隔不能超过 24 小时</p>   |
| 请求样例   | <p>请求参数(post 提交):</p> <pre> appkey = "c318br6RLex12IeBs0Ta6wo1"; tradenum = "20130701180155090"; mintradetime = ""; maxtradetime = ""; time ="20130701120311"; sign="YzMxOGJyNlJMZXgxMk1lQnMwVGE2 d28xMjAxMzA3MDExODAxNTUwOTA1bXY 4NVFhM1l4MUtnMWZsQmhoRWZlM2Q=" </pre>  |
| 返回样例   | <pre> {   'total':1,   'time': '20130704172613225',   'msg': 'W3sndHJhZGVudW0nOicy MDEzMDcwMTE4MDE1NTA5MCcsJ3R yYWRldGltZSc6JzIwMTMtMDctMD EgMTg6MDM6MzInLCd0cmFkZWFTb3 VudCc6MS4wLCd0cmFkZXN OYXRlcyc6MX1d', </pre>   |

|  |  |
|--|--|
|  | <pre>'sc': '1', 'st': '查询订单成功' } 其中 msg 节点通过解密后的明文为 [ {   'tradenum': '20130701180155090',   'tradetime': '2013-07-0118:03:32',   'tradeamount': 1.0,   'tradestatus': 1 } ]</pre> |
|--|--|

## 5 规范

### 5.1 错误码返回格式：

```
{
  "sc": "204",
  "st": ".....",
  "time": "20130301101010"
}
```

### 5.2 返回状态码

| 返回码     | 说明            |
|---------|---------------|
| 200 或 1 | 请求（调用）成功      |
| 0       | 失败(sid 无效)    |
| 5       | Sign 无效       |
| 10      | 请求参数错误        |
| 202     | 登录失败，用户状态不正常  |
| 205     | 账号不存在         |
| 206     | 用户已经登录        |
| 207     | 用户已经退出        |
| 208     | 用户名、邮箱、手机号已存在 |
| 905     | 内部接口错误        |
| 906     | IO 异常         |
| 907     | SQL 异常        |
| 908     | Runtime 异常    |
| 999     | 其他错误          |

