密付支付平台开发者手册

密付支付平台给游戏，直播等互联网应用提供了一个基于区块链加密货币的支付系统，方便最终用户通过加密货币給各种互联网应用支付和兑出。互联网应用采用密付完成支付和兑出，具备高效，安全，跨区域，保护隐私和降低支付成本等一些列的优势。

技术对接流程

一,网站后台新建应用。

二,根据应用信息，调用不同业务接口。

三,调起支付访问页面。（参考API接口9）

四,等待支付成功回调。（参考API接口10）

一，新建应用

开发者门户网址：https://www.cryptocurpays.com:80/

开发者可以根据不用的业务，新建多个应用，进行使用。

开发者输入信息

|  |  |  |
| --- | --- | --- |
| **参数名称** | **是否必须** | **说明** |
| 应用名称 | 是 | 在密付平台显示的应用名称，唯一，不可重复 |
| 回调地址 | 是 | 密付平台回调商户后台的访问地址，要求完整url格式，协议，地址，端口。密付根据不同业务，会调用不同接口，例如，用户支付成功后，会调用deposit接口 |
| 应用类型 | 是 | 业务应用涉及的类型 |
| 货币 | 是 | 应用想要结算的货币类型 |
| 应用签名地址 | 否 | 应用服务器向支付网关withdraw请求的时候，需要使用ECDSA（Elliptic Curve Cryptography Algorithm）对于请求信息进行签名。应用服务器自行生成一个私钥（Private Key），然后把私钥所对应的地址（Address），上传到开发者门户。  (暂时不需要) |

开发商获取信息

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| appId | 新建应用的id，唯一，根据业务接口请求，需要添加到请求参数中 |
| 平台验证参数(GatewayAddress) | 支付网关签名地址。支付网关发送给应用服务器的deposite成功消息，会附件签名验证。使用该地址验证签名的合法性。 |
| 密钥(AppSecret) | 应用密钥。应用服务器发送給支付网关的部分restful请求需要附加appsecret参数来验证请求的合法性。 |

二,根据应用信息,调用不同业务接口。

api接口列表

1，Get Rates 获取加密货币汇率接口

2，Get Token 获取用户token接口

3，Get Address 获取用户加密货币收款地址接口

4，Get Unconfirmed Deposit 获取用户连上充值处理中状态信息接口

5，Check Deposit Data 开发商主动验证支付信息合法性接口

6，Get Balance 获取应用加密货币库存数量接口

7，Post New Withdraw Request 开发商提取加密货币库存接口

8，Get Withdraw Status 获取处理中提币请求的状态信息接口

9，Get Open OTC Url 获取打开支付界面(发起支付，调起支付界面)

需要开发商服务器实现的接口列表

10，Post A New Deposit 通知开发商充值成功信息(必须)

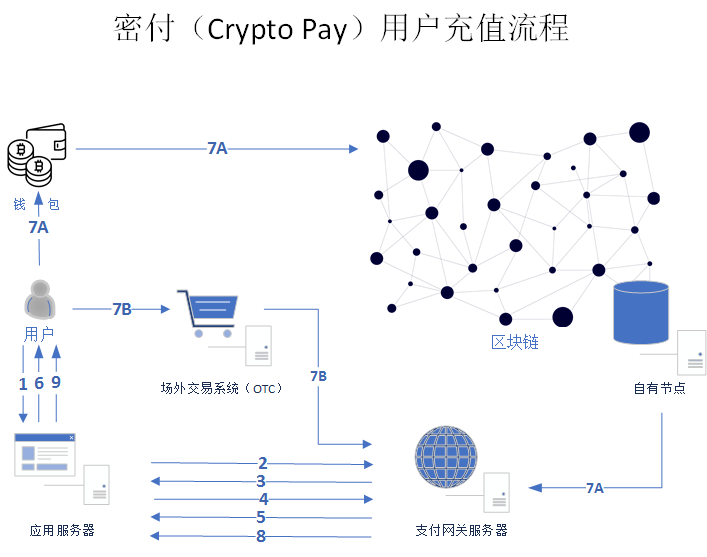
11，Withdraw Succeed 通知开发商提币成功信息(非必须)

简版支付调用流程(快速调起支付界面)

接口调用顺序：2->9>10

1，先根据接口2，请求到对应用户的token。

2，使用token到接口9请求回用户的OTC支付界面url，在url后，添加开发商想要支付的对应限制信息，打开支付界面

3，等待支付成功的消息通知

接口详情

返回参数数据的统一格式

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| type | Int | 请求的有效性，0，请求成功，非0，请求失败，失败原因，附异常编码 |
| data | String | 业务请求的有效数据信息，json格式 |

下文文档中，只包含data中的业务有效数据参数

1，Get Rates 获取加密货币汇率接口

因为加密货币和法币的兑换比例会随时浮动，所以最终用户使用加密货币支付可以兑换的虚拟货币也是一个浮动值。应用服务器如果需要展示当前订单需要支付加密货币的数量，可以调用下面的接口来获取最新汇率。

*请求路由*:Get: /rates/:fiatCurName

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| fiatCurName | String | 是 | 汇率对应的法币币种现在支持cny(人民币),usd(美金) |

请求参数:无

例如:<https://api.cryptocurpays.com/api/v1/rates/cny>

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| eth | float | 对eth的汇率 |
| usdt | float | 对usdt的汇率 |
| btc | float | 对btc的汇率 |
| ccn | float | 对ccn的汇率 |

例如:{"type":0,"data":{"eth":505.2672,"bch":893.3159999999999,"btc":6227.7222,"usdt":1.02}}

2，Get Token 获取用户token接口

支付网关为每个应用的每个最终用户生成一个访问令牌－Access Token。应用服务器对于最终用户的所有访问需要提供Access Token。在获取和最终用户相关的所有信息之前，应用服务器需要先获取该用户的Token。

游戏服务器向支付服务器请求某个的Access Token。游戏服务器使用本应用的App Secret来请求，支付网关下发Access Token給游戏服务器。如果支付网关发现请求的玩家是一个新玩家，则先在支付网关后台生成一个玩家的Access Token。

游戏服务器可以把这个Token下发給浏览器或者客户端应用程序，客户端或者浏览器可以使用这个Token直接访问支付网关，进行相关的操作

*请求路由:*Get: /:appId/token/:userId

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |
| userId | String | 是 | 开发商请求的用户id，是用户的唯一识别码，开发商后台唯一，原则上需要填写，不填写可能会造成支付流程复杂度提升，财务报表统计数据缺失 |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appsecret | String | 是 | 应用在后台生成的appsecret秘钥参数 |

例如:https://api.cryptocurpays.com/api/v1/122/token/Lisha?appsecret=GO2wW63AMEglXNmtQQeu9vYy5UwKrxmgc3DgSrpRdrf6fLY7cTqhTccQOlIdQiUvergHixziWbJsFcgaLY0zNB7oPEUXz7VBOGE0ksCPdmPHyuwVZ2yaGamYAKx33Emr46qREf9U0u3m7JFtOszsxlTIaq4ayxdYZD4WclTBMHZNtM3siqJqbbDaV4peFl5iNavxUffe

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| token | String | 用户的Token数值 |

例如:{"type":0,"data":{"token":" xgY0esCYVxoY4tDw7XBN6ZOMUP8L10 "}}

3，Get Address 获取用户加密货币收款地址接口

应用客户端或者应用服务器向支付网关服务器请求某应用某玩家的收款地址。

*请求路由:*Get : /{appid}/users/{userid}/address

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |
| userId | String | 是 | 开发商请求的用户id，是用户的唯一识别码，开发商后台唯一，原则上需要填写，不填写可能会造成支付流程复杂度提升，财务报表统计数据缺失 |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| token | String | 是 | 用户的Token数值 |

例如：https://api.cryptocurpays.com/api/v1/122/users/user1/address?token=xgY0esCYVxoY4tDw7XBN6ZOMUP8L10

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| addresses | String | json格式，内容包含密付支持的加密货币币种充值地址 |
| fiatRate | float | 币种的当前兑换汇率，对应法币为新建应用的时候，选择的结算币种类型 |

例如:

{"type":0,"data":{"addresses":{"eth":{"address":"0x48091678e2102a312e4e93be98e155bad41bb3c0","fiatRate":3235.2258816000003},"btc":{"address":"mxStsTm3uWrwjvUx6kQQvwzg7sADFAPm9J","fiatRate":39876.1052466},"usdt":{"address":"1HvwaQg56VRgxp1LPBS372nMFsZWMDLDUf","fiatRate":6.53106}}}}

4，Get Unconfirmed Deposit 获取用户连上充值处理中状态信息接口

如果最终用户通过自己的加密币钱包支付，加密币在提交到公链之后，有一个挖矿确认的过程，所以可能会有pending的deposits。应用客户端可以使用该API来查询最终用户是否有正在确认当中的支付交易。

*请求路由:*Get : /{appid}/users/{userid}/pendingDeposits

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |
| userId | String | 是 | 开发商请求的用户id，是用户的唯一识别码，开发商后台唯一，原则上需要填写，不填写可能会造成支付流程复杂度提升，财务报表统计数据缺失 |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| token | String | 是 | 用户的Token数值 |

例如:https://api.cryptocurpays.com/api/v1/122/users/user1/pendingdeposits?token=xgY0esCYVxoY4tDw7XBN6ZOMUP8L10

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| eth | String | json格式，内容包含eth币种的充值订单信息 |
| fiatRate | float | 币种的当前兑换汇率，对应法币为新建应用的时候，选择的结算币种类型 |
| 0x43644f650f4e500c38f634ebfda3b73f2f2824d3405fcca3f4ddd9a3eaa647f3 | String | 区块链上的txid信息 |
| from | String | 链上交易的fromaddress地址 |
| value | float | 这笔交易的加密货币数量 |
| confirmations | Int | 当前确认了多少个块数 |
| need | Int | 完成交易需要确认的块数 |

例如:

{"type":0,"data":{"eth":{"0x43644f650f4e500c38f634ebfda3b73f2f2824d3405fcca3f4ddd9a3eaa647f3":{"from":"0xdeaf4aec5b133b5e827d5dfb9a37d242e5e552a9","value":"0.05","confirmations":2,"need":8},"0x0f7f15fcbca01fa641da7f22737cf076dca9e10d43936dbf77142c45972257cf":{"from":"0xdeaf4aec5b133b5e827d5dfb9a37d242e5e552a9","value":"0.05","confirmations":1,"need":8}}}}

5，Check Deposit Data 开发商主动验证支付信息合法性接口

应用服务器需要验证收到的支付信息的有效性的时候，可以调用这个接口

*请求路由:*Get : /{appid}/tx/check/deposit

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appsecret | String | 是 | 应用在后台生成的appsecret秘钥参数 |
| txid | String | 是 | 开发者需要查询的txid数值，只有txid是该appid的合法有效数据，才会正常返回订单数据，否则返回失败 |

例如:https://api.cryptocurpays.com/api/v1/122/tx/check/deposit?appsecret=GO2wW63AMEglXNmtQQeu9vYy5UwKrxmgc3DgSrpRdrf6fLY7cTqhTccQOlIdQiUvergHixziWbJsFcgaLY0zNB7oPEUXz7VBOGE0ksCPdmPHyuwVZ2yaGamYAKx33Emr46qREf9U0u3m7JFtOszsxlTIaq4ayxdYZD4WclTBMHZNtM3siqJqbbDaV4peFl5iNavxUffe&txid=xgY0esCYVxoY4tDw7XBN6ZOMUP8L10

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| appId | Int | 此交易订单所属的应用的appid |
| cryptoType | String | 此交易订单的加密货币类型 |
| cryptoValue | Float | 此交易订单的加密货币数量 |
| fee | Float | 此交易订单的手续费 |
| fiatCurrency | String | 此交易订单的法币结算类型 |
| fiatRate | Float | 此交易订单的法币结算汇率 |
| fromAddress | String | 此交易订单的充值来源地址 |
| orderid | String | 此交易订单的商户订单编号 |
| timestamp | Int | 此交易订单的完成时间 |
| toAddress | String | 此交易订单的充值目标地址 |
| txId | String | 此交易订单的密购区块链单号，系统唯一，可作为对账的唯一识别 |
| userId | String | 此交易订单的开发商的用户id |
| signature | String | 此交易订单的加密验证字符串，加密验证部分，请参考加deposit密验证部分 |

例如:

{"type":0,"data":{"appId":"154","cryptoType":"ccn","cryptoValue":"100","fee":"-2","fiatCurrency":"cny","fiatRate":"1","fromAddress":"otc","orderid":"ord154355515518722","timestamp":"1543555357006","toAddress":"18Cs7nfrzJaRZaMwrp3ZWctS6X4TUPaFJN","txId":"0x013094ffc0ee3f1e74898fc2555368cf8b","userId":"2","signature":"H1uuI2SZYIdj4The8jXs8PUPQg9sdmTgYdESLE6ZBzjCXpjiaw88g4cDBcwBZF0Yw5U2HOYgUzfHnGLOWhsZcJ0="}}

6，Get Balance 获取应用加密货币库存数量接口

应用服务器向支付服务器请求该应用在支付服务器上不同加密货币的balance。

*请求路由:*Get: /{appid}/balance

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appsecret | String | 是 | 应用在后台生成的appsecret秘钥参数 |

例如:https://api.cryptocurpays.com/api/v1/122/balance?appsecret=GO2wW63AMEglXNmtQQeu9vYy5UwKrxmgc3DgSrpRdrf6fLY7cTqhTccQOlIdQiUvergHixziWbJsFcgaLY0zNB7oPEUXz7VBOGE0ksCPdmPHyuwVZ2yaGamYAKx33Emr46qREf9U0u3m7JFtOszsxlTIaq4ayxdYZD4WclTBMHZNtM3siqJqbbDaV4peFl5iNavxUffe

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| eth | String | 应用库存币种 |
| 30.39 | Float | 币种数量 |

例如:

{“type”:0,“data”:{"eth":30.39,"bch":39.01}}

7，Post New Withdraw Request 开发商提取加密货币库存接口(暂未开通)

支付网关API不但可以处理最终用户向应用服务器支付的请求，而且可以处理应用服务器提取加密货币的请求。

*请求路由*:Post: /{appid}/txs/withdraw/{cyptotype}

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |
| cyptotype | String | 是 | 要兑出的币种类型 |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| cyptoType | String | 是 | 要兑出的币种类型 |
| toAddress | String | 是 | 要对出的加密货币目标地址 |
| orderId | String | 是 | 代表唯一的withdraw请求。如果应用服务器提交了两次请求，两次请求使用相同的orderId，则支付网关当作相同的请求处理。 |
| cryptoValue | Float | 是 | 要兑出的加密货币数量 |
| signature | String | 是 | 应用服务器提交支付网关的Post Data需要包含Signature Key。使用对应的Private key使用ECDSA对下面的信息签名。 |

返回参数:

|  |  |  |
| --- | --- | --- |
| 参数名称 | 数据类型 | 说明 |
| orderId | String | 兑出订单号和请求参数相同 |
| withdrawId | String | 此兑出行为的系统内订单编号 |

例如:{type:0,data:{orderId:"10000000000037",withdrawId:9}}

8，Get Withdraw Status 获取处理中提币请求的状态信息接口(暂未开通)

游戏服务器或者游戏客户端向支付网关服务器更新自己请求正在Pending Withdraw的状态。

*请求路由*:Get: /{appid}/txs/withdraws/{cyptotype}

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |
| cyptotype | String | 是 | 要查询的兑出账单加密货币类型 |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| tx\_hash | String | 是 | 兑出行为的区块链交易hash值 |

例如:https://api.cryptocurpays.com/api/v1/1234434/withdraws/eth?tx\_hash=’fasdfsadf4rt34rfafsdafsdafd’

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| timestamp | Int | 交易时间 |
| txHash | String | 交易hash值 |
| blockHeight | Int | 交易所在区块链块上高度 |
| latestBlock | Int | 区块链最新交易高度 |
| confirmationNeed | Int | 确认交易完成需要的交易高度数量 |
| cyptoValue | Int | 加密货币交易数量 |
| fiatCur | String | 交易法币结算类型 |
| fiatRate | Float | 交易法币结算汇率 |

例如:

{"type":0,"data":{"timestamp":987653849,"txHash":"ujegxe4ce41d88bbb6ce734c1784ea5951f310242751df512bb6048ef2fb4ef49ec7e","blockHeight":345678,"latestBlock":45680,"confirmationNeed":10,"cyptoValue":0.12,"fiatCur":"CNY","fiatRate":0.001}}

9，Get Open OTC Url 获取打开支付界面(发起支付，调起支付界面)

游戏服务器需要向支付网关请求一个只针对当前游戏玩家的，打开购币OTC的链接地址，请求方式如下

*请求路由:*Get: /{appId}/otcurl/{userId}

路由参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| appId | Int | 是 | 请求应用的平台appid |
| userId | String | 是 | 开发商请求的用户id，是用户的唯一识别码，开发商后台唯一，原则上需要填写，不填写可能会造成支付流程复杂度提升，财务报表统计数据缺失 |

请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| token | String | 是 | 用户的Token数值 |
| userName | String | 是 | 用户的昵称，用做显示用 |

例如:

<https://api.cryptocurpays.com/api/v1/122/otcurl/user1?token=xgY0esCYVxoY4tDw7XBN6ZOMUP8L10&userName=张三>

返回参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| data | String | 打开otc界面基础url跳转链接 |

data中的链接，为登陆链接。

可以支持本用户登录购币充值平台进行购币操作，生成一次url地址，可供用户15分钟以内登录使用，过期需要获取新的登录地址信息。

如果需要引导用户购买对应币种或者币值的加密货币 可以再登陆链接中，添加参数

获取到支付基础URL之后，可以在此基础上，添加和支付相关的参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| matchone | Int | 意向引导的购买金额 |
| matchtype | String | 意向购买的加密货币种类,默认需要填写ccn |
| payways | JsonArray | [1,0](意向购买支付渠道 0 银行卡 1 支付宝 2 微信) |
| backopenurl | String | Imh0dHBzOi8vd3d3LmJhaWR1LmNvbT9tPTEyMyZuPTk4NzIi(为保证回跳链接附带参数的完整性，url地址请进行base64编码，此地址实际是<https://www.baidu.com>，用户下单完成后的跳回的商家界面地址，可选，如果不填，otc购币充值平台将跳过回跳逻辑。地址采用标准网站url地址格式，回跳时，除backopenurl完整地址，不附带其他形式的参数) |
| orderid | String | 开发者平台附带的订单编号，订单交易完成后，会随着deposit通知消息，一同通知到开发者后台。 |

例如：

http://www.cryptocurbuy.com:8080/#/buyer?appId=MTAwMDEz&userId=Mjk5OQ%3D%3D&acttime=1536921584948&userNmae=hello&address=%7B%22eth%22%3A%7B%22address%22%3A%220xe1a7a1f90a09eb9ea19790dd5c885722dbcf9038%22%7D%2C%22bch%22%3A%7B%22address%22%3A%221GjiF9KfEUHsjJrqgcr2QaYh1DYo3stuqR%22%7D%2C%22usdt%22%3A%7B%22address%22%3A%221GjiF9KfEUHsjJrqgcr2QaYh1DYo3stuqR%22%7D%2C%22btc%22%3A%7B%22address%22%3A%221GjiF9KfEUHsjJrqgcr2QaYh1DYo3stuqR%22%7D%7D&sign=8af796f0b3adb99c991c9a7ba93c00c9&matchone=192&matchtype=eth&backopenurl=aHR0cHM6Ly93d3cuYmFpZHUuY29tP209MTIzJm49OTg3Mg&orderid=201810241650

需要开发商服务器实现的接口列表

10，Post A New Deposit 通知开发商充值成功信息(必须)

当支付网关在自己管理的区块链节点上面发现可被确认的，给最终用户地址发送加密货币的transaction之后，支付网关通过该最终用户所属应用的Callback URL 通知应用服务器。

例如:Post : https://gameCallbackURL/deposit

请求参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| appId | Int | 此交易订单所属的应用的appid |
| cryptoType | String | 此交易订单的加密货币类型 |
| cryptoValue | Float | 此交易订单的加密货币数量 |
| fee | Float | 此交易订单的手续费 |
| fiatCurrency | String | 此交易订单的法币结算类型 |
| fiatRate | Float | 此交易订单的法币结算汇率 |
| fromAddress | String | 此交易订单的充值来源地址 |
| orderid | String | 此交易订单的商户订单编号 |
| timestamp | Int | 此交易订单的完成时间 |
| toAddress | String | 此交易订单的充值目标地址 |
| txId | String | 此交易订单的密购区块链单号，系统唯一，可作为对账的唯一识别 |
| userId | String | 此交易订单的开发商的用户id |
| signature | String | 此交易订单的加密验证字符串，加密验证部分，请参考加deposit密验证部分 |

例如：

{orderid:'amks9172eiu1jqk2'appId:1,cryptoType:'eth',cryptoValue:0.01,fee:0.0001,fiatCurrency:'cny',fiatRate:3374.3374596,fromAddress:'0xd54ee29702c3d11cc66ac0f0a35982bcd06b0cb4',timestamp:1529359814,toAddress:'0xc4726dc0f185b621218b709e48590b9701b305fa',txId:'0x820169b3e003777414e823e0405d9ae74674ab4329b445cae3001c20d9fae1a8',userId:9,signature:'HzPhJjzBfBPYeDl2KdxuDO9JAEknNbzI/qGgzMnIMui8Jo2QjhuR2ldbLeaccbyTkPj7KiQqwqkVJbReya0fLxU='}

接受通知消息后，需用加密算法验证数据的合法性，加密验证算法行参考后续加密验证部分。

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| message | String | 是 | 收到成功后的回复消息。 |

如果验证消息有效，请回复received，如果验证失败，请回复error

例如:{"message":"received!"}"

11，Withdraw Succeed 通知开发商提币成功信息(非必须)

支付网关对于已经广播出去的Withdraw的交易Transaction，定期检测该交易被挖矿的状态，如果挖矿的区块数量大于要求的数量，则給开发商的Callback URL发送信息，通知游戏开发商这个提币的请求已经完成。

例如:Post: https://callbackURL/withdraw

请求参数:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| cryptoType | String | 此交易订单的加密货币类型 |
| cryptoValue | Float | 此交易订单的加密货币数量 |
| fee | Float | 此交易订单的手续费 |
| txId | String | 此交易订单的密购区块链单号，系统唯一，可作为对账的唯一识别 |
| orderId | String | 此交易订单的商户订单编号 |
| withdrawType | Int | 此交易订单的兑出方式类型 |
| withdrawId |  | 此交易订单的兑出订单id |

例如：

{"txId":"0xf87b8f9539d66d26fba775e05bb77349bf7f0fe5b48f6f75f51896ed5afcd635",  "orderId":10000000044,  "cyptoType":"eth",  "cyptoValue":0.05,  "fee":0.0001,  "withdrawType":0,  “withdrawId”:10000000044,}

收到通知消息后，验证后需要返回消息。

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **是否必须** | **说明** |
| message | String | 是 | 收到成功后的回复消息。 |

如果验证消息有效，请回复received，如果验证失败，请回复error

例如:{"message":"received!"}"

ECDSA签名和验证

签名和验证流程描述

开发商应用在支付网关注册的过程中，开发商需要生成一个随机的私钥（Private Key From 1 to 2256），该Private Key可以演算出对应的Public Key和Base58Check Address。开发商将Address上传給支付网关，而私钥仅仅个人拥有，对任何第三方保密。如果开发商希望通过Restful API withdraw加密货币，该请求的信息需要使用private key签名。支付网关会通过地址验证签名的有效性。

开发商在支付网关新建应用之后，支付网关后台会生成相应的随机私钥，Private Key。Private Key不会透露給开发商，但是Private Key对应的Base58 Check的地址会提供给开发商。支付网关确认收到一个deposit之后，会通过callback URL通知应用服务器。该信息会使用private key进行签名。应用服务器应该使用address来验证该信息是否有效。

ECDSA签名的作用

信息不能被篡改

修改交易信息的任何部分都会造成签名验证失败。因此可通过验证签名来确认交易请求在网络传输过程种，没有被篡改。

信息不可抵赖

因为Receiver并不知道Sender的Private Key，所以Receiver不能杜撰一个交易请求。Sender发出来的交易请求是不可抵赖的。

加密验证Demo请参考Git目录:https://github.com/JeremywangCN/cryptopayapi

示例

Nodejs示例

使用的module

https://github.com/bitpay/bitcore-lib

https://github.com/bitpay/bitcore-message

安装bitcore-lib module

npm install bitcore-lib

npm install bitcore-message

在JS里面引用相关Module

var bitcore = require('bitcore-lib');

var Message = require('bitcore-message');

var Hash = require('bitcore-lib').crypto.Hash;

生成随机Private Key

var privateKey = new bitcore.PrivateKey();

计算Private Key对应的Base58check地址

var address = new Address(privateKey.toPublicKey()，bitcore.Networks.mainnet,'pubkeyhash');

生成签名

postObj.signature = Message(' This is a sample message ').sign(privateKey);

验证签名

var verified = Message(' This is a sample message ').verify(address, signature);

签名和验证签名的消息

支付网关会首先对于消息体进行sha256sha256的哈希计算，然后把计算出来的签名增加到消息体。在验证签名的时候，首先会把消息体里面的signature这个key取出来，然后把删除signature key的消息，进行sha256sha256的哈希计算。对于计算结果进行签名验证。

let message = Hash.sha256sha256(Buffer.from(JSON.stringify(postObj))).toString;

错误代码

|  |  |
| --- | --- |
| **编号** | **说明** |
| 0 | 消息返回成功 |
| 9001 | 系统繁忙 |
| 9007 | 错误的参数 |
| 10002 | 多余的应用信息 |
| 10003 | 不存在的应用信息 |
| 10004 | 不存在的用户id信息 |
| 10005 | 异常的用户token数据 |
| 10006 | 异常的汇率信息 |
| 10007 | 无效的加密货币类型 |
| 10008 | /错误的 地址信息 |
| 10009 | 不存在应用信息 |
| 10010 | 错误的withdraw 金额参数 |
| 10011 | 账户库存不足 |
| 10016 | 无效的txid信息 |