快币公司平台服务接入规范(V1.0)

快币（上海）网络技术有限公司

2016 年 7 月

文档修订历史			
版本号	**日期**	**说明**	**作者/审阅**
V1.0	2016/7/13	初稿	孙誉

第一章简介

一、目的

提供商户端通过快币公司提供的账单缴费接口开发指南

二、功能描述

按照本规范所描述的标准，商户查询上海（目前仅支持上海）的水电煤缴费服务。

接口采用签名+https 数据保护方式来保证商户与网上支付平台间的身份验证、中间信息传递的完整性，以便进行电子商务安全当中非常重要的交易身份辨识、不可抵赖、防止篡改等功能。

三、运行环境

请根据以下 API 文档，将请求通过 HTTP/HTTPS 发送至服务端即可。

四、配置测试应用

测试环境地址：https://cs.keycoin.cn/plat/
客户端简称：cs0001
测试环境私钥：
MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBANuwp0aY+rnL3Q2K899SLtzq
JWwFEOgz2zj3tbKGClvLFB/62tuIyHIu3oZs2o132FqcYEq+XJ2Q6FcEuYIEtBPoNdlUdHt4kNgRiG
L1EWr/tUFUg5qdmvgGqYNP/z7PWl7YgEczkKBNg8Ha1pMrTfLcbQj8Wgl5zRmjxOPQhtuXAgM
BAAECgYAawp7to47O+P05PBbOuiUJNeBPEyLnFgH7I3lXgKoj1R3jc1wbZ8L63aiveSz6FIIfL3Mv
R5bifpc+mk26f7YriLOuXd8ZqoIm4AUwtG1gAPIpqcGATYEE4jwBReflLTwskRqiKu+7Xiwt7UwJv
HveCnQ9cI/bJf1quREfrkTZ4QJBAP1mTB1PZVqcejaQLb9iWCUocTzqgEgBu4yARB85dEfo7WM
dY04YRbhMZ5K9/y/u+8WL5hq0MOR9xOSTOvHgGMcCQQDd8cxXAXECgIX1hRePFlnwQT2ju
wdyMWtSFFqs/8y800oaHSpU8TV7rCRDerScTTb1dJbINVQEuoMR3t4nhnaxAkEAk/iurdb1Oq
pxe3dKH55QFrK9HXNaobHRUUR9gzZLvso4BU7d9HacD/kd6kTkxZrqjvsYzbM3cdY3qbqIC0Jvd
QJAeK4IcOqsuzKNJHII3FnH75ZZ4NRKeTIb69Z7c4PnnGi66zYelW9OqH9yfGqVGEP39UE+13//v
vk54BFiNrSo1EQJBAJby2geOSMeVl2SpE7M2Br0CH1XlEHCfGp6XE/29N/MmjKeLh5/PDKC+SlzS
F8GXZeYaViREpfs4kZKj0UAUsBQ=
测试环境公钥：
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDbsKdGmPq5y90NivPfUi7c6iVsBRDoM9
s497WyhgiLyxQf+trbiMhyLt6GbNqNd9hanGBKvlydkOhXBLmCBLQT6DXZVHR7eJDYEYhi9RFq/
7VBVIOanZr4BqmDT/8+z1pe2IBHM5CgTYPB2taTK03y3G0I/FoJec0Zo8Tj0IbblwIDAQAB

第二章 接入说明

接入流程

配置参数

配置参数	参数描述
sign	请求的签名验证字符串
clientAbbr	快币给接入方分配的客户端简称（平台系统唯一）
Private key	快币给接入方分配生成的 RSA（MD5withRSA）私钥
Public key	快币给接入方分配生成的 RSA（MD5withRSA）公钥

签名生成规则

1，请求参数名按照字典法升序排列
2，依次取出参数值按 key=value 的形式以&符拼接成字符串（末尾若有“&”，则删除）
3，将以上字符串使用 RSA private key 加密 生成签名,生成的签名进行 url 编码，存入参数 sign
注：若将使用 get/post 参数则需将 sign 值进行 url 编码
若放入 body，则无需编码

参数列表

1. 根据城市代码查询支持的产品信息

接口地址	https://cs.keycoin.cn/plat/plat/getLiftFeeInfo				
参数列表	参数名称	别名	类型	必传	备注
	cityCode	城市代码	String	Y	目前只支持上海。应为：021
	clientAbbr	客户端简称	String	Y	
	Sign	签名	String	Y	

返回值
(字段说明参照实体类注释)
	{
	"data": {
	"gasBill": "1000000211", //燃气费
	"powerRate": "1000000210", //电费
	"waterRate": "1000000209" //水费
	},
	"code": "0",
	"msg": "请求成功"
	}

2. 产品机构列表查询交易

接口地址	https://cs.keycoin.cn/plat/plat/productQuery				
参数列表	参数名称	别名	类型	必传	备注
	cityCode	城市代码	String	Y	目前只支持上海。应为：021
	clientAbbr	客户端简称	String	Y	
	productId	产品 ID	String	Y	根据 1 的接口返回值，按需传入
	Sign	签名	String	Y	

返回值
(字段说明参照实体类注释)
	{
	"data": [//data 中的数据为 list，使用英文“-”隔开，为 ID - 名称
	"888880002102900-上海市自来水市南有限公司",
	"888880002202900-上海市松江自来水有限公司",
	"888880002222900-上海市松江东部自来水有限公司",
	"888880002232900-上海市松江西部自来水有限公司",
	"888880002302900-上海城投水务（集团）有限公司",
	"888880002402900-上海嘉定自来水有限公司",
	"888880002502900-上海浦东威立雅自来水有限公司 ",
	"888880002602900-上海浦东新区自来水有限公司 ",
	"888880002702900-上海市自来水市南有限公司（闵行）",
	"888880002802900-上海南汇自来水有限公司",
	"888880002812900-上海临港供排水发展有限公司",
	"888880002902900-上海市自来水奉贤有限公司",
	"888880003002900-上海封浜自来水有限公司"
	],
	"code": "0",
	"msg": "请求成功"
	}				

3. 机构支持账单查询方式应答参数说明

接口地址	https://cs.keycoin.cn/plat/plat/billConfQuery				
参数列表	参数名称	别名	类型	必传	备注
	billOrgId	机构 ID	String	Y	根据 2 的接口返回值，按需传入
	clientAbbr	客户端简称	String	Y	
	productId	产品 ID	String	Y	根据 1 的接口返回值，按需传入
	Sign	签名	String	Y	
返回值 (字段说明参照实体类注释)	{ "data": [{ "needPwd": "N", //没有使用， "searchType": "0", //0-使用户号查询，其他-使用条形码查询 "searchTypeName": "销根号", //因机构不同而显示的名称不同，可用可不用 "validationExp": "^\\d{9}$" //该号码的正则表达式验证 }], "code": "0", "msg": "请求成功" }				

4. 查询账单

接口地址	https://cs.keycoin.cn/plat/plat/queryByBillNo				
参数列表	参数名称	别名	类型	必传	备注
	billOrgId	机构 ID	String	Y	根据 2 的接口返回值，按需传入
	clientAbbr	客户端简称	String	Y	
	productId	产品 ID	String	Y/N	根据 1 的接口返回值，按需传入
	startMonth	账单月份	String	Y/N	格式 :yyyyMM 默认当前月
	billNo	户号	String	Y/N	二选一必传，根

	barcode	条形码	String	Y/N	据3的接口返回值类型传入
	searchType	查询类型	String	Y/N	根据3的接口返回值传入
	Sign	签名	String	Y	

返回值
(字段说明参照实体类注释)
	<pre>{
	"data": [
	{
	"barcode": "19010000117061000020230",
	"billAddr": "",
	"billAmt": 20.23,
	"billId": "201707101000242117",
	"billMonth": "201706",
	"billNo": "190100001",
	"billOrgId": "888880002302900",
	"billOrgName": "上海城投水务（集团）有限公司",
	"billRecordTimes": "01",
	"billStatus": "00",
	"isInsurance": "N",
	"overdueFee": 0
	}
	],
	"code": "0",
	"msg": "请求成功"
	}</pre>				

说明	条形码查询必传字段： clientAbbr 、barcode、billOrgId、sign 户号查询必传字段： billOrgId、clientAbbr、productId、startMonth、billNo、searchType、sign				

返回值说明：

字段名	中文名称	类型	说明
billId	账单代码	String	用户账单创建订单必填参数
billOrgId	出账机构代码	String	
billOrgName	出账机构名称	String	
billNo	账单号	String	
billMonth	账期	String	格式为 YYYYMM
billAmt	账单金额	number	单位为:元
billRecordTimes	抄次	String	
barcode	条码	String	
billAddr	账单地址		

billOwner	户名		
overdueFee	滞纳金	number	单位为:元
isInsurance	是否保险费		Y/N

5. 创建订单

接口地址	https://cs.keycoin.cn/plat/plat/createBillOrder				
参数列表	参数名称	别名	类型	必传	备注
	clientAbbr	客户端简称	String	Y	
	clientOrderNo	订单号	String	Y	
	notifyUrl	回调地址	String	Y	
	billId	账单代码	String	Y	
	billOrgId	出账机构的代码	String	Y	
	platType	记录类型	Integer	Y	1-水 2-电 3-煤
	barCode	条形码号	String	Y	户号和条码必须有一个不为空
	doorCode	户号	String	Y	
	Sign	签名	String	Y	
返回值 (字段说明参照实体类注释)	{ "data": { " clientOrderNo": "20170711103520544085", "thirdOrderNo":"20170711103520544085", " totalFee ":"20.4" }, "code": "0", "msg": "请求成功" }				
说明	创建订单需在查询账单 20 分钟内提交，否则需要重新查询账单				

6. 查询缴费记录

接口地址	https://cs.keycoin.cn/plat/plat/getRecords				
参数列表	参数名称	别名	类型	必传	备注
	clientAbbr	客户端简称	String	Y	
	clientOrderNo	接入方订单号	String	Y	二选一必传，亦可以都传（若都传入值，则会根据两个值进行全值匹配，若传入两个值不属于同一订单，则查询不到具体数据）
	thirdOrderNo	第三方订单号	String	Y	
	Sign	签名	String	Y	

返回值
(字段说明参照实体类注释)
	{
	"data": {
	" clientOrderNo": "20170711103520544085",
	"orderAmt": "20.23",
	" thirdOrderNo": "801707111000128464",
	"orderStatus": "01"
	},
	"code": "0",
	"msg": "请求成功"
	}				

7. 订单状态后台通知（http 协议）

接口地址	根据传入参数，确定回调地址				
参数列表	参数名称	别名	类型	必传	备注
	clientAbbr	客户端简称	String	Y	
	clientOrderNo	接入方订单号	String	Y	
	thirdOrderNo	第三方订单号	String	Y	
	orderStatus	订单状态	String	Y	交易状态 00 处理中 01 成功 02 失败

	Sign	签名	String	Y	通过分配的私钥加密，使用公钥验证
返回值 (字段说明参照实体类注释)	该请求不接收返回值，若处理不成功；请调用查询缴费记录接口				

错误码提示

0	请求成功
10001	加密验证失败
10002	客户端简称不正确
1004	系统异常
1005	参数错误
20001	订单不存在
554008	商户订单号已存在,无法重复下单
554009	查无此记录
554999	未查到符合条件的数据

第三章附录

RSA 加密及验证示例代码：

代码 JDK 版本为 1.7

```
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;
import java.security.KeyFactory;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;
```

```
/**
	* 根据私钥加密字符串,编码集为 UTF-8
	* @param data
	* @param privateKey
	* @param encyName		默认为 "MD5withRSA"
	* @return
	* @throws Exception
	*/
	public static String encodeByRSA(String data,String privateKey, String encyName) throws Exception {
		if (StringUtil.isBlank(encyName)) {
			encyName = "MD5withRSA";
		}
		BASE64Decoder base64Decoder = new BASE64Decoder();
		BASE64Encoder base64Encoder = new BASE64Encoder();
		KeyFactory keyf = KeyFactory.getInstance("RSA");
		PrivateKey privateKeyPair =
keyf.generatePrivate(new PKCS8EncodedKeySpec(base64Decoder.decodeBuffer(privateKey)));
		java.security.Signature signet = java.security.Signature.getInstance(encyName);
		signet.initSign(privateKeyPair);
		signet.update(data.getBytes("utf-8"));
		return base64Encoder.encode(signet.sign());
	}
```

```
/**
     * 根据公钥验证加密串 ,编码集为 UTF-8
     * @param data
     * @param sign
     * @param publicKeyString
     * @param encyName          默认为 "MD5withRSA"
     * @return
     * @throws Exception
     */
    public static boolean verify(String data, String sign, String publicKeyString, String encyName) throws Exception {
        if (StringUtil.isBlank(encyName)) {
            encyName = "MD5withRSA";
        }
        BASE64Decoder base64Decoder = new BASE64Decoder();
        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        PublicKey publicKeyStr =
keyFactory.generatePublic(new X509EncodedKeySpec(base64Decoder.decodeBuffer(publicKeyString)));
        java.security.Signature signet = java.security.Signature.getInstance(encyName);
        signet.initVerify(publicKeyStr);
        signet.update(data.getBytes("utf-8"));
        return signet.verify(base64Decoder.decodeBuffer(sign));
    }
```