文档编号	
文档版本	1.0
保密级别	

连连银通电子支付有限公司

连连支付

手机应用 WEB 应用开发指南

2013/4/10

目　录

文档修订记录

序号	日期	版本号	修订说明	修订人	审核人
1	2014-06-24	1.0	新建	钱栋	

1. 文档说明

1.1 面向读者

本文档主要面向需要接入连连银通 WEB 安全支付的商户技术人员。本文档以 JAVA 为例，其他语言开发者亦可参考。

1.2 读者所需技能

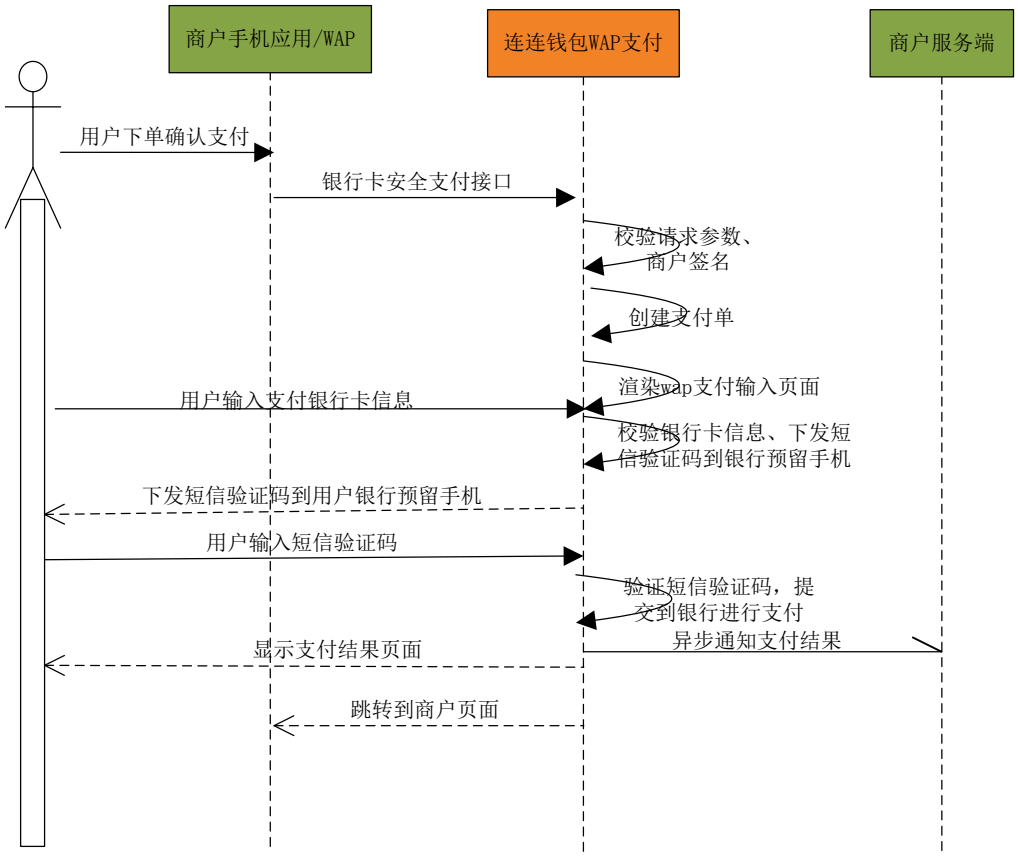
读者需有基本的程序开发背景，掌握 java 及 WEB 开发技术。

2. 连连支付 WEB 安全支付简介

2.1 WEB 安全支付服务介绍

连连支付安全支付 WEB 服务是提供给商户网站应用调用的 WEB 支付服务，主要用来向其它的应用程序提供便捷、安全以及可靠的 WEB 支付途径。

2.2 WEB 安全支付服务时序图



2.2.1 商户构造请求数据

商户根据连连支付安全支付 WEB 系统提供的本接口规则，通过程序生成得到签名结果及要传输给连连支付支付 WEB 的数据集合。

2.2.2 商户发送请求数据

商户把构造完成的数据集合，通过POST方式传递给连连支付WEB安全支付接口。

2.2.3 连连支付 WEB 对请求数据进行处理

连连支付WEB得到这些数据集合后，会先进行安全校验等验证，验证通过后就进入连连支付WEB收银台界面。

2.2.4 连连支付 WEB 返回支付完成的结果数据

对于支付完成的交易，连连支付会以两种方式把结果反馈给商户。

- 客户端WEB以GET方式直接返回支付结果（参数url_return，如果商户没设定，则不会进行该操作）。
- 连连支付服务器主动发起通知，调用商户在请求时设定好的通知地址（参数notify_url，如果商户没设定，则不会进行该操作）。

2.2.5 商户对获取的返回结果数据进行处理

商户在客户端（参数 url_return 指定的同步回显地址）或服务器异步通知地址（参数 notify_url 指定通知地址）获取连连支付支付平台返回的结果数据后，可以结合自身网站的业务逻辑进行数据处理（如：订单更新、自动充值到会员账号中等）。

3. WEB 安全支付接入流程

3.1 商户开户

根据签订的商户协议，由连连银通市场部完成商户开户流程。

3.2 密钥发放和配对

商户签约成功后，可以获得连连银通支付公钥。另外，商户需要生成商户公钥和商户私钥（具体步骤见本文档第 5 节安全签名机制），并且将商户公钥提交给连连银通支付。至此，接入前期准备工作完成，下一节将使用 demo 测试准备工作是否正确。

3.3 Demo

为了便于商户的接入，我们提供了安全支付 demo。通过本 demo，商户可测试 3.1 节的前期准备工作是否正确完成，同时还可参考 demo 的代码完成接入。本文档以 java 为例。

3.4 Demo 配置运行

3.4.1 步骤 一

解压 Demo 开发包 Webdemo.rar，将其导入 MyEclipse

3.4.2 步骤二

导入成功后打开 PartnerConfig.java，按照注释修改商户的配置信息。及 index.jsp 页面修改商户用户 ID 等信息。

```
public interface PartnerConfig{
    // 银通公钥
    String YT_PUB_KEY     = "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCSS/DiwdCf/aZsxxcacDnoc
    // 商户私钥
    String TRADER_PRI_KEY = "MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wggJbAgEAAoGBAMlGNh/WsyZSY
    // MD5 KEY
    String MD5_KEY        = "201306031000001013";
    // 接收异步通知地址
    String NOTIFY_URL     = "http://ip:port/wepdemo/notify.htm";
    // 支付结束后返回地址
    String URL_RETURN     = "http://ip:port/wepdemo/urlReturn.jsp";
    // 商户编号
    String OID_PARTNER    = "201306031000001013";
    // 签名方式 RSA或MD5
    String SIGN_TYPE      = "MD5";
    // 接口版本号，固定1.0
    String VERSION        = "1.0";

    // 业务类型，连连支付根据商户业务为商户开设的业务类型；  (101001：虚拟商品销售、109001：实物商品销售、10

    String BUSI_PARTNER   = "101001";
}
```

3.4.3 步骤三

使用 jboss/Tomcat 部署并运行

浏览器地址栏输入 http://localhost:8080/webdemo/

输入商户系统中用户唯一 ID，选择“普通接入”或“卡前置接入”，进入演示：

4.开发

本章指导在商户项目中集成 WEB 安全支付，关键代码以 Demo 为例。

4.1 3 个核心服务 servlet

ToPayServlet.java： 支付服务

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
{
    req.setCharacterEncoding("utf-8");
    resp.setCharacterEncoding("utf-8");
    // 创建订单
    OrderInfo order = createOrder(req);
    RequestDispatcher dispatcher = null;
    String paymod = req.getParameter("paymod");
    if ("plain".equals(paymod))
    {
        plainPay(req, order);
        dispatcher = req.getRequestDispatcher("/gotoPlainPay.jsp");
    } else
    {
        prepositPay(req, order);
        dispatcher = req.getRequestDispatcher("/gotoPrepositPay.jsp");
    }
    dispatcher.forward(req, resp);
}
```

ReceiveNotifyServlet.java： 异步通知服务

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
{
    resp.setCharacterEncoding("UTF-8");
    System.out.println("进入支付异步通知数据接收处理");
    RetBean retBean = new RetBean();
    String reqStr = LLPayUtil.readReqStr(req);
    if (LLPayUtil.isnull(reqStr))
    {
        retBean.setRet_code("9999");
        retBean.setRet_msg("交易失败");
        resp.getWriter().write(JSON.toJSONString(retBean));
        resp.getWriter().flush();
        return;
    }
    System.out.println("接收支付异步通知数据: 【" + reqStr + "】");
    try
    {
        if (!LLPayUtil.checkSign(reqStr, PartnerConfig.YT_PUB_KEY,
                PartnerConfig.MD5_KEY))
        {
            retBean.setRet_code("9999");
            retBean.setRet_msg("交易失败");
            resp.getWriter().write(JSON.toJSONString(retBean));
            resp.getWriter().flush();
            System.out.println("支付异步通知验签失败");
            return;
        }
    } catch (Exception e)
    {
        System.out.println("异步通知报文解析异常: " + e);
        retBean.setRet_code("9999");
        retBean.setRet_msg("交易失败");
        resp.getWriter().write(JSON.toJSONString(retBean));
        resp.getWriter().flush();
        return;
    }
    retBean.setRet_code("0000");
    retBean.setRet_msg("交易成功");
    resp.getWriter().write(JSON.toJSONString(retBean));
    resp.getWriter().flush();
    System.out.println("支付异步通知数据接收处理成功");
    // 解析异步通知对象
    PayDataBean payDataBean = JSON.parseObject(reqStr, PayDataBean.class);
    // TODO:更新订单，发货等后续处理
}
```

InforQueryServlet.java: 银行卡查询 ajax 服务

```
/**
 * 银行卡卡bin信息查询
 * @param req
 * @return
 */
private String queryCardBin(HttpServletRequest req)
{

    JSONObject reqObj = new JSONObject();
    reqObj.put("oid_partner", PartnerConfig.OID_PARTNER);
    reqObj.put("card_no", req.getParameter("card_no"));
    reqObj.put("sign_type", PartnerConfig.SIGN_TYPE);
    String sign = LLPayUtil.addSign(reqObj, PartnerConfig.TRADER_PRI_KEY,
            PartnerConfig.MD5_KEY);
    reqObj.put("sign", sign);
    String reqJSON = reqObj.toString();
    System.out.println("银行卡卡bin信息查询请求报文[" + reqJSON + "]");
    String resJSON = HttpRequestSimple.getInstance().postSendHttp(
            ServerURLConfig.QUERY_BANKCARD_URL, reqJSON);
    System.out.println("银行卡卡bin信息查询响应报文[" + resJSON + "]");
    return resJSON;
}

/**
 * 用户已绑定银行列表查询
 * @param req
 * @return
 */
private String queryBankcardList(HttpServletRequest req)
{
    JSONObject reqObj = new JSONObject();
    reqObj.put("oid_partner", PartnerConfig.OID_PARTNER);
    reqObj.put("user_id", req.getParameter("user_id"));
    reqObj.put("offset", "0");
    reqObj.put("sign_type", PartnerConfig.SIGN_TYPE);
    String sign = LLPayUtil.addSign(reqObj, PartnerConfig.TRADER_PRI_KEY,
            PartnerConfig.MD5_KEY);
    reqObj.put("sign", sign);
    String reqJSON = reqObj.toString();
    System.out.println("用户已绑定银行列表查询请求报文[" + reqJSON + "]");
    String resJSON = HttpRequestSimple.getInstance().postSendHttp(
            ServerURLConfig.QUERY_USER_BANKCARD_URL, reqJSON);
    System.out.println("用户已绑定银行列表查询响应报文[" + resJSON + "]");
    return resJSON;
}
```

4.2 处理连连支付支付结果

支付完成后，点击返回商户按钮，WEB以GET方式直接请求参数url_return指定的地址，如果商户没设定，则不会进行该操作。另外连连银通支付服务器会向参数notify_url指定的通知地址发送支付结果通知。

5.安全签名机制

MD5 安全签名机制说明

MD5安全签名机制是商户和连连银通约定一个签名key，每次在做签名时将key=value附在待签名字符串后面，然后经MD5加密运算后得到一个签名串，商户和连连银通在检验时也采用同样的方式得到签名串，经比对后确定是否一致，如果一致，则签名通过。

RSA 安全签名机制说明

在RSA签名时，需要私钥和公钥一起参与签名。私钥与公钥皆是客户通过OPENSSL来生成得出的。客户把生成出的公钥与连连银通的技术人员配置好的连连支付的公钥做交换。因此，在签名时，客户要用到的是客户的私钥及连连银通的公钥。

- 支付请求时签名

商户当拿到请求时的待签名字符串后，把待签名字符串与商户的私钥一同放入 RSA 签名函数中进行签名运算，从而得到签名结果字符串。

- 通知或返回时验证签名

商户当获得到通知或返回时的待签名字符串后，把待签名字符串、连连银通提供的公钥、连连支付通知返回参数中的参数sign的值三者一同放入RSA签名函数中进行非对称的签名运算，来判断签名是否验证通过。

RSA密钥生成工具及验证指南文档，可在本文档同目录下找到。

若需要签名和验签的方法代码，可从demo中提取。

需要参与签名的参数

直接把请求数据中的所有元素(除sign本身)按照“key值=value值”的格式拼接起来，并且把这些拼接以后的元素以“&”字符再连接起来（顺序按首字母升序排列，值为空的不参与签名），如：

```
busi_partner=101001&dt_order=20130516131212&info_order=用户13958069593购买了3桶羽毛球&money_order=210.97&name_goods=羽毛球&no_order=20130516000000001&notify_url=http://payhttp.xiaofubao.com/***/back.shtml&oid_partner=201304121000001004&sign_type=RSA
```

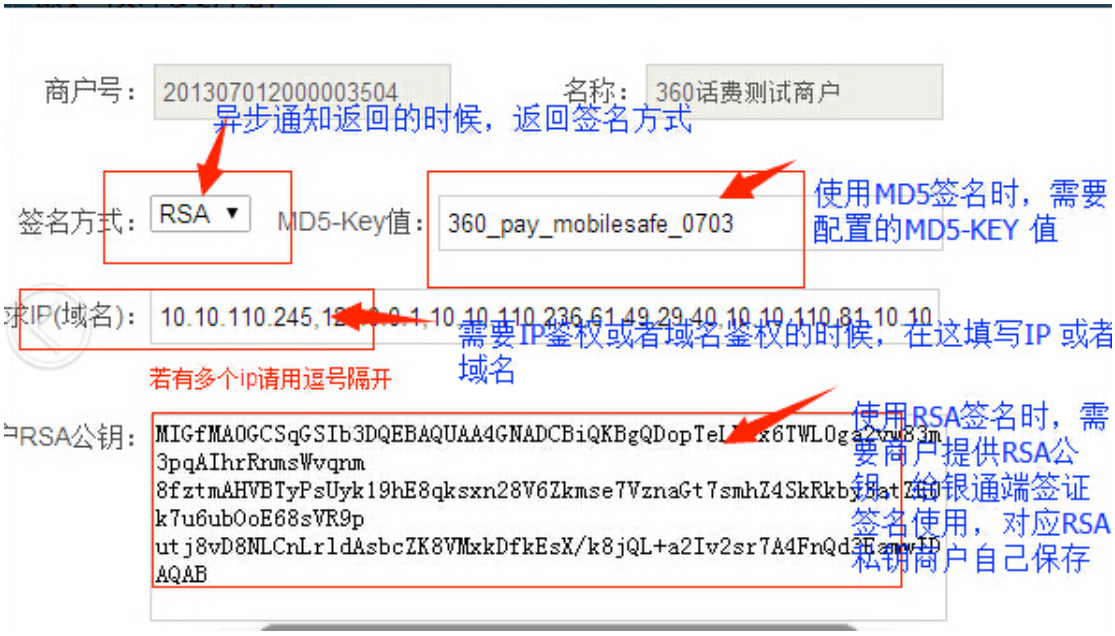
这段字符串即是商户支付请求时的待签名字符串。

如果为MD5加密的则如下：(参数按顺序按首字母升序排列，值为空的不参与签名，MD5的key值放在最后，其他待签名字符串同理可得)

```
busi_partner=101001&dt_order=20130516131212&info_order=用户13958069593购买了3桶羽毛球&money_order=210.97&name_goods=羽毛球&no_order=20130516000000001&notify_url=http://payhttp.xiaofubao.com/***/back.shtml&oid_partner=201304121000001004&sign_type=MD5&key=lianlian1234567890
```

6. 签名秘钥获取及 IP 域名配置

在正式开完商户站后，需要登录正式商户站 https://yintong.com.cn/merchant/trader/login.htm ，在商户站安全中心-》商户秘钥维护进行配置，界面入下图：



1、API 接口（https://yintong.com.cn/traderapi/ ）这种地址开头的接口需要在上面配置请求服务器 IP

2、PC 端 Web 接口（https://yintong.com.cn/payment/bankgateway.htm ）需要在上面配置请求域名。

7. 风控参数列表

商户用户信息				
*用户唯一标识	user_info_mercht_userno	是	String	客户在商户系统中生成的编号
用户登陆名	user_info_mercht_userlogin	否	String	客户在商户系统注册的登陆名(手机号、邮箱等标识)
绑定手机号	user_info_bind_phone	否	String	客户在商户系统绑定的手机号码
*客户姓名	user_info_full_name	是	String	商品类目为实名类业务必传 可以只提供姓氏隐藏名字 例如：李*
客户证件类型	user_info_id_type	否	String	0：身份证或企业经营证件 1：户口簿， 2：护照 3：军官证, 4：士兵证

				5： 港澳居民来往内地通行证 6：台湾同胞来往内地通行证 7： 临时身份证 8： 外国人居留证 9： 警官证 X：其他证件
*客户证件号码	user_info_id_no	是	String	商品类目为实名类业务必传， 证件号可以隐藏中间生日部分 例如：330825********3325
注册时间	user_info_dt_register	否	String(14)	YYYYMMDDH24MISS
是否实名	user_info_identify_state	是	String	1 实名 2 非实名
注册认证方式	user_info_identify_type	否	String	实名认证必填 1 银行卡认证 2 现场认证 3 身份证远程认证 4 其它认证
注册 IP	user_info_register_ip	否	String	
联系地址省级编码	user_info_addr_province	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
联系地址市级编码	user_info_addr_city	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
联系地址区/县级编码	user_info_addr_district	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
联系地址区/县级以下	user_info_addr_street	否	String	区/县以下级别的地址
联系人电话	user_info_phone	否	String	
联系人电话区号	user_info_phone_area_code	否	String	杭州市： 0571
联系人电话国家号	user_info_phone_county_code	否	String	跨境业务必填
联系地址省级编码	user_info_addr_province	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
***商户支付交易信息**				
*商品类目	frms_ware_category	否	String	见《商品类目代码表》
用户支付请求时间	frms_dt_request	否	String(14)	YYYYMMDDH24MISS
用户支付请求 IP	frms_ip_addr	否	String	
手机支付物理信息				
Imei 号	frms_imei	否	String(40)	Android 手机送 imei 号，iphone 送 ifda 手机唯一编号
Sim 号	frms_sim_id	否	String(40)	

机器编码	frms_mechine_id	否	String	Uuid(imei+imsi)
Mac 地址	frms_mac_addr	否	String(40)	
PC 安全控件支付物理信息				
cpu 信息	frms_cpu	否	String	
Mac 地址	frms_mac_addr	否	String	
磁盘信息	frms_disk	否	String	
版本号	frms_version	否	String	安全控件版本号
***商户端签约或支付验证信息（商户端自行验证必传）**				
*验证方式	verify_type	否	String	1、短信验证 2、密码 3、其他 4、无验证
*验证方式	verify_type	否	String	1、短信验证 2、密码 3、其他 4、无验证
*验证手机号	verify_phoneno	否	String	验证下发的手机号码
*验证时间	verify_dt	否	String(14)	YYYYMMDDH24MISS
商户订单物流信息参数				
收货地址全名	delivery_addr_full	否	String	地址全名
收货地址省级编码	delivery_addr_province	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
收货地址市级编码	delivery_addr_city	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
收货地址县/区级编码	delivery_addr_district	否	String	编码见最新县及县以上行政区划代码(截止 2012 年 10 月 31 日)
收货地址县/区以下级	delivery_addr_street	否	String	区/县以下级别的地址
*收货人姓名	delivery_full_name	是	String	实物类业务必传，可以只提供姓氏隐藏名字 例如：李*
收货人联系电话	delivery_phone	否	String	
收货人邮箱	delivery_mail	否	String	

备注：以上字段传送要求可能根据业务风险有所调整，以双方风控部门的最终意见为准。