



链滴

微信支付 SDK 文档 (V3.0.9)

作者: [chinaYoung](#)

原文链接: <https://ld246.com/article/1534734497833>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

微信支付 Java SDK

对[微信支付开发者文档](#)中给出的API进行了封装。

com.github.wxpay.sdk.WXPay类下提供了对应的方法：

方法名	说明
microPay	刷卡支付
unifiedOrder	统一下单
orderQuery	查询订单
reverse	撤销订单
closeOrder	关闭订单
refund	申请退款
refundQuery	查询退款
downloadBill	下载对账单
report	交易保障
shortUrl	转换短链接
authCodeToOpenid	授权码查询openid

- 注意:
- 证书文件不能放在web服务器虚拟目录，应放在有访问权限控制的目录中，防止被他人下载
- 建议将证书文件名改为复杂且不容易猜测的文件名
- 商户服务器要做好病毒和木马防护工作，不被非法侵入者窃取证书文件
- 请妥善保管商户支付密钥、公众帐号SECRET，避免密钥泄露
- 参数为 `Map<String, String>` 对象，返回类型也是 `Map<String, String>`
- 方法内部会将参数转换成含有 `appid`、`mch_id`、`nonce_str`、`sign_type` 和 `sign` 的XML
- 可选HMAC-SHA256算法和MD5算法签名
- 通过HTTPS请求得到返回数据后会对其做必要的处理（例如验证签名，签名错误则抛出异常）
- 对于downloadBill，无论是否成功都返回Map，且都含有 `return_code` 和 `return_msg`，若成功其中 `return_code` 为 `SUCCESS`，另外 `data` 对应对账单数据

示例

配置类MyConfig:

```
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;

public class MyConfig extends WXPayConfig {
```

```

private byte[] certData;

private String appid = "wxxsxxxxxxxxxxxx";
private String mch_id = "xxxxxxxx";
private String key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";

public MyConfig() {
    String certPath = "/path/to/apiclient_cert.p12";
    File file = new File(certPath);
    try(InputStream certStream = new FileInputStream(file)) {
        this.certData = new byte[(int) file.length()];
        certStream.read(this.certData);
        certStream.close();
    } catch (Exception e) {
    }
}

public String getAppid() {
    return appid;
}

public void setAppid(String appid) {
    this.appid = appid;
}

public String getMch_id() {
    return mch_id;
}

public void setMch_id(String mch_id) {
    this.mch_id = mch_id;
}

public void setKey(String key) {
    this.key = key;
}

@Override
String getAppID() {
    return this.appid;
}

@Override
String getMchID() {
    return this.mch_id;
}

@Override
String getKey() {
    return this.key;
}

@Override
InputStream getCertStream() {

```

```

        ByteArrayInputStream certBis = new ByteArrayInputStream(this.certData);
        return certBis;
    }

    @Override
    IWXPayDomain getWXPayDomain() {
        return new IWXPayDomain() {

            @Override
            public void report(String domain, long elapsedTimeMillis, Exception ex) {
            }

            @Override
            public DomainInfo getDomain(WXPayConfig config) {
                return new DomainInfo("api.mch.weixin.qq.com", true);
            }
        };
    }

    @Override
    public boolean shouldAutoReport() {
        return false;
    }
}

```

统一下单:

```

import com.github.wxpay.sdk.WXPay;

import java.util.HashMap;
import java.util.Map;

public class WXPayExample {

    public static void main(String[] args) throws Exception {

        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config);

        Map<String, String> data = new HashMap<String, String>();
        data.put("body", "腾讯充值中心-QQ会员充值");
        data.put("out_trade_no", "2016090910595900000012");
        data.put("device_info", "");
        data.put("fee_type", "CNY");
        data.put("total_fee", "1");
        data.put("spbill_create_ip", "123.12.12.123");
        data.put("notify_url", "http://www.example.com/wxpay/notify");
        data.put("trade_type", "NATIVE"); // 此处指定为扫码支付
        data.put("product_id", "12");

        try {
            Map<String, String> resp = wxpay.unifiedOrder(data);
            System.out.println(resp);
        }
    }
}

```

```
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

订单查询:

```
import com.github.wxpay.sdk.WXPay;

import java.util.HashMap;
import java.util.Map;

public class WXPayExample {

    public static void main(String[] args) throws Exception {

        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config);

        Map<String, String> data = new HashMap<String, String>();
        data.put("out_trade_no", "2016090910595900000012");

        try {
            Map<String, String> resp = wxpay.orderQuery(data);
            System.out.println(resp);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

退款查询:

```
import com.github.wxpay.sdk.WXPay;

import java.util.HashMap;
import java.util.Map;

public class WXPayExample {

    public static void main(String[] args) throws Exception {

        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config);

        Map<String, String> data = new HashMap<String, String>();
        data.put("out_trade_no", "2016090910595900000012");

        try {
            Map<String, String> resp = wxpay.refundQuery(data);
            System.out.println(resp);
        }
    }
}
```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

下载对账单:

```

import com.github.wxpay.sdk.WXPay;

import java.util.HashMap;
import java.util.Map;

public class WXPayExample {

    public static void main(String[] args) throws Exception {

        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config);

        Map<String, String> data = new HashMap<String, String>();
        data.put("bill_date", "20140603");
        data.put("bill_type", "ALL");

        try {
            Map<String, String> resp = wxpay.downloadBill(data);
            System.out.println(resp);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

其他API的使用和上面类似。

暂时不支持下载压缩格式的对账单，但可以使用该SDK生成请求用的XML数据:

```

import com.github.wxpay.sdk.WXPay;
import com.github.wxpay.sdk.WXPayUtil;

import java.util.HashMap;
import java.util.Map;

public class WXPayExample {

    public static void main(String[] args) throws Exception {

        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config);

        Map<String, String> data = new HashMap<String, String>();
        data.put("bill_date", "20140603");
        data.put("bill_type", "ALL");
    }
}

```

```

data.put("tar_type", "GZIP");

try {
    data = wxpay.fillRequestData(data);
    System.out.println(WXPayUtil.mapToXml(data));
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

收到支付结果通知时，需要验证签名，可以这样做：

```

import com.github.wxpay.sdk.WXPay;
import com.github.wxpay.sdk.WXPayUtil;

import java.util.Map;

public class WXPayExample {

    public static void main(String[] args) throws Exception {

        String notifyData = "...."; // 支付结果通知的xml格式数据

        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config);

        Map<String, String> notifyMap = WXPayUtil.xmlToMap(notifyData); // 转换成map

        if (wxpay.isPayResultNotifySignatureValid(notifyMap)) {
            // 签名正确
            // 进行处理。
            // 注意特殊情况：订单已经退款，但收到了支付结果成功的通知，不应把商户侧订单状态从
            款改成支付成功
        }
        else {
            // 签名错误，如果数据里没有sign字段，也认为是签名错误
        }
    }
}

```

HTTPS请求可选HMAC-SHA256算法和MD5算法签名：

```

import com.github.wxpay.sdk.WXPay;
import com.github.wxpay.sdk.WXPayConstants;

public class WXPayExample {

    public static void main(String[] args) throws Exception {
        MyConfig config = new MyConfig();
        WXPay wxpay = new WXPay(config, WXPayConstants.SignType.HMACSHA256);
    }
}

```

```
    // .....  
  }  
}
```

若需要使用sandbox环境:

```
import com.github.wxpay.sdk.WXPay;  
import com.github.wxpay.sdk.WXPayConstants;  
  
public class WXPayExample {  
  
    public static void main(String[] args) throws Exception {  
        MyConfig config = new MyConfig();  
        WXPay wxpay = new WXPay(config, WXPayConstants.SignType.MD5, true);  
        // .....  
    }  
  
}
```

资源下载

[WxPayAPI_JAVAzip](#)