

支付宝接入的那点小事儿

前言

本次的 Chat 算是一个新手入门指南，但在内容不会局限于入门导引，适应准备对接支付宝以及正在对接支付宝遇到问题的开发者。

本次 Chat 中涉及到的图片均来自于网络，截图均为 macOS 和 Chrome 以及手机上的截图，如有侵权请联系删除。

希望这次 Chat，能够帮助还没有接触过支付宝对接的开发者或者公司更快更方便的了解支付宝的支付方式(支付场景)以及对接流程。帮助已在对接但途中遇到棘手问题的开发者或者公司提供一些参考的解决方案。

如果你有什么问题，都可以在文章底部的评论框中留言，参与讨论。

[支付宝CSDN博客文章列表](#)

什么是支付宝



提供的解决方案

支付宝公司从2004年建立开始到目前为止到底提供了那些解决方案呢？在支付宝开发平台的行业解决方案栏目中不难发现涉及到了以下几个领域。

1. **商业消费**
2. **交通出行**
3. **政务名声**
4. **教育医疗**

本次 Chat 主要带着大家来聊聊支付宝为我们提供了哪些支付方式来解决以上领域的无现金支付以及如何快速接入这些支付方式。

以上问题的答案就在[开发平台的文档](#)中。好了，说了这么多的费话现在开始我们的探索。

GitChat



开始前必读

开始开发

设置开发者管理员

应用介绍

创建应用

使用应用

使用沙箱环境

使用金融云产品

支付能力

当面付

App支付

手机网站支付

电脑网站支付

花呗分期

开始前必读

开始开发

设置开发者管理员

应用介绍

创建应用

使用应用

使用沙箱环境

使用金融云产品

支付能力

当面付

App支付

手机网站支付

电脑网站支付

花呗分期

资金能力

单笔转账到支付宝账户

新当面资金授权

收发现金红包

开放平台概述

蚂蚁金服开放平台基于支付宝的海量用户，将强大的支付、营销、数据能力，通过接口等形式开放给第三方合作伙伴，帮助第三方合作伙伴创建更具竞争力的应用。

通过接入蚂蚁金服开放平台，第三方合作伙伴可以获得更多的流量、用户和收益，而第三方合作伙伴提供的服务也让用户获得了更丰富的体验，平台生态更加繁荣，最终实现多方共赢。

通过平台能做什么

开发者是开放生态的主要组成部分，通过与商家、消费者的合作，提升商家的服务效率和营收，降低运营成本，使消费者的体验更便捷、更愉悦、更完美。无论你是第三方开发者、系统商、线下推广组织还是商家，都能基于开放平台找到适合自己的角色和方向。

插件提供商：

通过我们提供的丰富的API接口，开发者可以开发并在应用市场上线各类插件，解决门店管理、支付核销、会员营销、数据分析等方面的问题。开放平台还为各种应用提供了清晰的盈利模式，通过应用市场的销售获得回报。

服务提供商：

在应用市场之外，合作伙伴还可以通过服务市场为商家提供店铺装修、拍摄修图、地面推广、门店代运营等服务。实现线上订购，线下服务。

场景分销商：

第三方APP或媒介可以借助蚂蚁金服开放平台的分销能力为商家提供基于门店、卡券、内容的分销服务，为商家的流量导入、品牌推广提供平台。分销商可以为所提供的服务获取收入。

平台的优势

业务优势

- 一站式商家服务：为商家提供支付、门店、营销、数据等一站式服务
- 海量的实名用户：4亿实名用户，迅速沉淀会员信息玩转会员体系
- 丰富的线下场景：超过50万家线下门店，包括餐饮、超市等多种行业

技术支持

- 接口支持：支付、分享、账户、营销、信用、服务窗等九大优质接口支持
- 开发工具：包括 Java、PHP、.Net等语言的SDK与Demo示例
- 服务保障：丰富的技术文档，开放的技术社区，高效的在线技术服务，是你开发的坚强后盾

盈利模式

- 通过应用获利：基于商家服务市场，为合作伙伴的插件及服务提供变现渠道
- 营销咨询：基于商家的营销需求，为合作伙伴提供多元的商家增值服务
- 代运营：合作伙伴提供商家运营托管服务，最大化商家的营销价值

新手入门

新手指南

热门FAQ

合作攻略

平台规则

服务支持

开发文档

业务文档

物料模板下载

蚂蚁开放日

客服帮助

技术支持中心

在线问答

客服：0571-88158090

(9:00-22:00)

合作洽谈

点此联系我

关注我们

关注开放平台生活号



打开支付宝扫一扫



注册开发平台推荐使用企业账户，因为后面使用到的一些支付方式仅对企业开放。

 支付宝 ALIPAY

注册

登录 | 注册 | 我的支付宝 | 服务大厅 | 提建议 | 更多

1 创建账户

2 填写账户信息

3 企业实名认证

4 注册成功

个人账户

企业账户

账户名

输入电子邮箱

验证码

请输入验证码

M15M

下一步

港澳台区海外企业注册

金融机构注册请致电 0571-88158090

注册前需要准备什么资料？

注册前需要准备：影印件必须为彩色原件的扫描件或数码照

• [营业执照影印件](#)

• 对公银行账户，可以是基本户或一般户

• [法定代表人的身份证影印件](#)

如果你是代理人，除以上资料，还需准备：

• [你的身份证影印件](#)

• [企业委托书](#)，必须盖有公司公章或者财务专用章，不能是合同/业务专用章 [查看模板](#)

如果你是代理人，除以上资料，还需准备：

- 你的身份证影印件
- 企业委托书，必须盖有公司公章或者财务专用章，不能是合同/业务专用章

企业实名认证时支付宝会在对公银行账户中打一笔小金额款项。认证完成之后此笔金额会转到你注册的支付宝账户的余额当中。

常见帮助问题可以参考[官方的FAQ](#)

账号注册已实名认证之后我们就开始开发。

开始开发

如果你不是刚才注册账号的拥有者，你来开发需要账号的拥有者添加开发者管理员才能进行下面的操作。[如何设置开发者管理员？](#)

创建应用与使用

具体流程这里就不截图了可以[参考这里](#)。这里我们使用沙箱环境的应用。[沙箱环境使用说明](#)

正式环境应用需要**创建应用**、**添加应用功能**、**配置应用环境**以及**应用申请上线**。

沙箱环境与正式环境应用都是需要配置应用环境其他步骤按照提示操作就行，下面我们重点来关注下如何配置应用的环境。

配置应用环境

配置参数说明

————— 配置应用环境 —————

Javen

APPID

2017102209460762

概览

开发者中心 / 网页&移动应用 / 应用列表 / Javen / 概览

提交审核 删除

1 完善基本信息，提交审核

2 等待审核结果

3 应用上线

基础信息

自用型应用

修改

应用名称

Javen

应用图标

功能选项

已经添加3项个人使用功能 + 继续添加

1项功能待签约后生效

立即签约

功能名称	功能介绍	是否需要签约	状态	操作
手机网站支付	商户在网页中调用支付宝提供的网页支付接口唤起支付宝客户端内的支付模块。通过网页跳转到支付宝中完成支付。	需签约	已生效	删除
APP支付	商户APP集成支付宝提供的SDK，通过SDK跳转到支付宝中完成支付，支付完后跳回商户APP内，展示支付结果。	需签约	已生效	删除
当面付	买家通过支付宝钱包付款码、扫码的方式给商户付款。完成交易，商户快捷收银，无需找零。	需签约	签约后生效	删除

☒ 已阅读并同意《支付宝业务协作协议》，《口碑业务协作协议》，《花呗分期产品推广协作协议》

开发配置

使用支付宝的部分功能前，需要先设置应用环境。查看如何使用

基础环境

支付宝网关	https://openapi.alipay.com/gateway.do
应用网关	设置
授权回调地址	设置

接口加签方式 如何生成密钥?

RSA2(SHA256)密钥(推荐)	设置应用公钥
--------------------	--------

使用其他加签方式 RSA(SHA1)密钥 查看两种加签方式的区别

提交审核

开发者所需配置内容：

应用名称 应用描述

下载对应版本的密钥生成工具

WINDOWS

MAC_OSX

具体的流程我这里就不贴图了参考这个[生成与配置官方文档](#)

上传工具产生的公钥之后会产生支付宝的公钥

以上配置完成我们可以获取到以下配置参数：

- appId 应用的编号
- privateKey 工具的生成应用私钥，公钥已上传并产生了支付宝的公钥
- alipayPulicKey 支付宝的公钥
- serverUrl 应用的网关

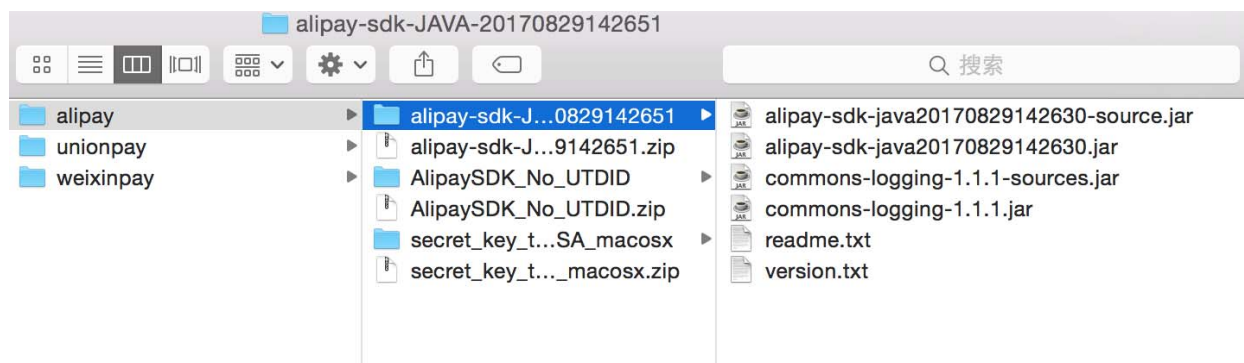
万事俱备只欠东风

开始对接

我这里已Java为例，为大家讲解如何对接支付宝提供的支付方式以及常用的接口。

- [当面付](#) 条码支付、扫码支付、声波支付
- [App支付](#)
- [手机网站支付](#)
- [电脑网站支付](#)
- [单笔转账到支付宝账户](#)
- [交易查询接口](#)
- [交易撤销接口](#)
- [交易退款接口](#)
- [退款查询接口](#)
- [查询转账订单接口](#)
- [查询对账单下载地址接口](#)

开发语言	资源下载	环境要求
JAVA 版资源	SDK1.5	适用于Java语言、jdk版本1.5及以上的开发环境
.NET版资源	SDK	适用于Visual studio 2010及以上版本、Framework3.5 SP1及以上版本的开发环境
PHP版资源	SDK	适用于php5.5以上的开发环境



服务端SDK使用的步骤：

- 1、初始化客户端 AlipayClient
- 2、实例化具体API对应的request类
- 3、设置请求参数
- 4、客户端执行请求 alipayClient.execute(request);
- 5、获取请求相应的对象 response
- 6、从相应的对象 response 中判断执行的结果

不同的接口具体使用总结可以参考我之前写的一篇博客 [支付宝支付-常用支付API详解\(查询、退款、提现等\)](#)

这里贴一下我开源项目 [IJPAY让支付触手可及](#) 中封装的代码


```

/**
 * APP支付
 * @param model
 * @param notifyUrl
 * @return {String}
 * @throws {AlipayApiException}
 */
    public static String startAppPayStr(AlipayTradeAppPayModel
model, String notifyUrl) throws AlipayApiException{
        AlipayTradeAppPayResponse response =
appPay(model,notifyUrl);
        return response.getBody();
    }

/**
 * APP支付
 * https://doc.open.alipay.com/docs/doc.htm?
treeId=54&articleId=106370&docType=1
 * @param model
 * @param notifyUrl
 * @return {AlipayTradeAppPayResponse}
 * @throws {AlipayApiException}
 */
    public static AlipayTradeAppPayResponse
appPay(AlipayTradeAppPayModel model, String notifyUrl) throws
AlipayApiException{
        //实例化具体API对应的request类,类名称和接口名称对应,当前调用接口
名称: alipay.trade.app.pay
        AlipayTradeAppPayRequest request = new
AlipayTradeAppPayRequest();
        //SDK已经封装掉了公共参数,这里只需要传入业务参数。以下方法为sdk的
model入参方式(model和biz_content同时存在的情况下取biz_content)。
        request.setBizModel(model);
        request.setNotifyUrl(notifyUrl);
        //这里和普通的接口调用不同,使用的是sdkExecute
        AlipayTradeAppPayResponse response =
AlipayApiConfigKit.getAlipayApiConfig().getAlipayClient().sdkExecu

```

```

    * @throws IOException
    */
    public static void wapPay(HttpServletResponse
response,AlipayTradeWapPayModel model,String returnUrl,String
notifyUrl) throws AlipayApiException, IOException {
        String form = wapPayToString(response, model, returnUrl,
notifyUrl);
        HttpServletResponse httpResponse = response;
        httpResponse.setContentType("text/html;charset=" +
AliPayApiConfigKit.getAliPayApiConfig().getCharset());
        httpResponse.getWriter().write(form);// 直接将完整的表单
html输出到页面
        httpResponse.getWriter().flush();
    }
    /**
    * WAP支付
    * @param response
    * @param model
    * @param returnUrl
    * @param notifyUrl
    * @return {String}
    * @throws {AlipayApiException}
    * @throws IOException
    */
    public static String wapPayToString(HttpServletResponse
response,AlipayTradeWapPayModel model,String returnUrl,String
notifyUrl) throws AlipayApiException, IOException {
        AlipayTradeWapPayRequest alipayRequest = new
AlipayTradeWapPayRequest();// 创建API对应的request
        alipayRequest.setReturnUrl(returnUrl);
        alipayRequest.setNotifyUrl(notifyUrl);// 在公共参数中设置回
跳和通知地址
        alipayRequest.setBizModel(model);// 填充业务参数
        return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().pageExe
cute(alipayRequest).getBody(); // 调用SDK生成表单
    }

```

```

    }
    /**
     * 条形码支付、声波支付
     * @param model
     * @param notifyUrl
     * @return {AlipayTradePayResponse}
     * @throws {AlipayApiException}
     */
    public static AlipayTradePayResponse
tradePayToResponse(AlipayTradePayModel model, String notifyUrl)
throws AlipayApiException{
        AlipayTradePayRequest request = new
AlipayTradePayRequest();
        request.setBizModel(model);// 填充业务参数
        request.setNotifyUrl(notifyUrl);
        return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request); // 通过
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient()调用API,
获得对应的response类
    }

```

```

    /**
     * 扫码支付
     * https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.i0UVZn&treeId=193&articleId=105170&docType=1#s4
     * @param model
     * @param notifyUrl
     * @return {String}
     * @throws {AlipayApiException}
     */
    public static String
tradePrecreatePay(AlipayTradePrecreateModel model, String
notifyUrl) throws AlipayApiException{
        AlipayTradePrecreateResponse response =

```

```

AlipayTradePrecreateRequest();
    request.setBizModel(model);
    request.setNotifyUrl(notifyUrl);
    return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);
}

/**
 * 单笔转账到支付宝账户
 * https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.54Ty29&treeId=193&articleId=106236&docType=1
 * @param model
 * @return {boolean}
 * @throws {AlipayApiException}
 */
public static boolean
transfer(AlipayFundTransToaccountTransferModel model) throws
AlipayApiException{
    AlipayFundTransToaccountTransferResponse response =
transferToResponse(model);
    String result = response.getBody();
    if (response.isSuccess()) {
        return true;
    } else {
        //调用查询接口查询数据
        JSONObject jsonObject =
JSONObject.parseObject(result);
        String out_biz_no =
jsonObject.getJSONObject("alipay_fund_trans_toaccount_transfer_re
sponse").getString("out_biz_no");
        AlipayFundTransOrderQueryModel queryModel = new
AlipayFundTransOrderQueryModel();
        model.setOutBizNo(out_biz_no);
        boolean isSuccess = transferQuery(queryModel);
        if (isSuccess) {
            return true;
        }
    }
}

```

```

AlipayFundTransToaccountTransferRequest();
    request.setBizModel(model);
    return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);
}

/**
 * 转账查询接口
 * @param model
 * @return {boolean}
 * @throws {AlipayApiException}
 */
public static boolean
transferQuery(AlipayFundTransOrderQueryModel model) throws
AlipayApiException{
    AlipayFundTransOrderQueryResponse response =
transferQueryToResponse(model);
    if(response.isSuccess()){
        return true;
    }
    return false;
}

/**
 * 转账查询接口
 * @param model
 * @return {AlipayFundTransOrderQueryResponse}
 * @throws {AlipayApiException}
 */
public static AlipayFundTransOrderQueryResponse
transferQueryToResponse(AlipayFundTransOrderQueryModel model)
throws AlipayApiException{
    AlipayFundTransOrderQueryRequest request = new
AlipayFundTransOrderQueryRequest();
    request.setBizModel(model);
    return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);
}

```

```

        return true;
    }
    return false;
}
/**
 * 交易查询接口
 * @param model
 * @return {AlipayTradeQueryResponse}
 * @throws {AlipayApiException}
 */
public static AlipayTradeQueryResponse
tradeQuery(AlipayTradeQueryModel model) throws
AlipayApiException{
    AlipayTradeQueryRequest request = new
AlipayTradeQueryRequest();
    request.setBizModel(model);
    return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);
}

/**
 * 交易撤销接口
 * https://doc.open.alipay.com/docs/api.htm?
spm=a219a.7395905.0.0.XIhh6e&docType=4&apiId=866
 * @param model
 * @return {boolean}
 * @throws {AlipayApiException}
 */
public static boolean isTradeCancel(AlipayTradeCancelModel
model) throws AlipayApiException{
    AlipayTradeCancelResponse response = tradeCancel(model);
    if(response.isSuccess()){
        return true;
    }
    return false;
}

```

```

(request);
    return response;
}
/**
 * 关闭订单
 * https://doc.open.alipay.com/docs/api.htm?spm=a219a.7629065.0.0.21yRUe&apiId=1058&docType=4
 * @param model
 * @return {boolean}
 * @throws {AlipayApiException}
 */
public static boolean isTradeClose(AlipayTradeCloseModel
model) throws AlipayApiException{
    AlipayTradeCloseResponse response = tradeClose(model);
    if(response.isSuccess()){
        return true;
    }
    return false;
}
/**
 * 关闭订单
 * @param model
 * @return {AlipayTradeCloseResponse}
 * @throws {AlipayApiException}
 */
public static AlipayTradeCloseResponse
tradeClose(AlipayTradeCloseModel model) throws
AlipayApiException{
    AlipayTradeCloseRequest request = new
AlipayTradeCloseRequest();
    request.setBizModel(model);
    return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);

}
/**
 * 关闭订单并创建接口

```

```
        return
        AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
        (request);
    }
}
```

```
    /**
     * 退款
     * https://doc.open.alipay.com/docs/api.htm?spm=a219a.7395905.0.0.SAyEeI&docType=4&apiId=759
     * @param model
     * @return {String}
     * @throws {AlipayApiException}
     */
    public static String tradeRefund(AlipayTradeRefundModel
model) throws AlipayApiException{
        AlipayTradeRefundResponse response =
        tradeRefundToResponse(model);
        return response.getBody();
    }
    /**
     * 退款
     * @param model
     * @return {AlipayTradeRefundResponse}
     * @throws {AlipayApiException}
     */
    public static AlipayTradeRefundResponse
tradeRefundToResponse(AlipayTradeRefundModel model) throws
AlipayApiException{
        AlipayTradeRefundRequest request = new
AlipayTradeRefundRequest();
        request.setBizModel(model);
        return
        AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
        (request);
    }
}
```

```
...
```



```

/**
 * 退款查询
 * @param model
 * @return {AlipayTradeFastpayRefundQueryResponse}
 * @throws {AlipayApiException}
 */
public static AlipayTradeFastpayRefundQueryResponse
tradeRefundQueryToResponse(AlipayTradeFastpayRefundQueryModel
model) throws AlipayApiException{
    AlipayTradeFastpayRefundQueryRequest request = new
AlipayTradeFastpayRefundQueryRequest();
    request.setBizModel(model);
    return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);
}

```

```

/**
 * 查询对账单下载地址
 * @param model
 * @return {String}
 * @throws {AlipayApiException}
 */
public static String
billDownloadurlQuery(AlipayDataDatSERVICEBillDownloadurlQueryMod
el model) throws AlipayApiException{
    AlipayDataDatSERVICEBillDownloadurlQueryResponse
response = billDownloadurlQueryToResponse(model);
    return response.getBillDownloadUrl();
}

```

```

/**
 * 查询对账单下载地址
 * @param model
 * @return
{AlipayDataDatSERVICEBillDownloadurlQueryResponse}
 * @throws {AlipayApiException}
 */

```

```

/**
 * 交易结算接口
 * https://doc.open.alipay.com/docs/api.htm?
spm=a219a.7395905.0.0.nl0RS3&docType=4&apiId=1147
 * @param model
 * @return {boolean}
 * @throws {AlipayApiException}
 */
    public static boolean
isTradeOrderSettle(AlipayTradeOrderSettleModel model) throws
AlipayApiException{
        AlipayTradeOrderSettleResponse response =
tradeOrderSettle(model);
        if(response.isSuccess()){
            return true;
        }
        return false;
    }
/**
 * 交易结算接口
 * @param model
 * @return {AlipayTradeOrderSettleResponse}
 * @throws {AlipayApiException}
 */
    public static AlipayTradeOrderSettleResponse
tradeOrderSettle(AlipayTradeOrderSettleModel model) throws
AlipayApiException{
        AlipayTradeOrderSettleRequest request = new
AlipayTradeOrderSettleRequest();
        request.setBizModel(model);
        return
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().execute
(request);
    }

/**
 * 电脑网站支付(PC支付)
 * @param httpResponse

```

```

        String form =
AliPayApiConfigKit.getAliPayApiConfig().getAlipayClient().pageExecute(request).getBody();//调用SDK生成表单
        httpResponse.setContentType("text/html;charset=" +
AliPayApiConfigKit.getAliPayApiConfig().getCharset());
        httpResponse.getWriter().write(form);//直接将完整的表单html输出到页面
        httpResponse.getWriter().flush();
        httpResponse.getWriter().close();
    }

```

```

/**
 * 将异步通知的参数转化为Map
 * @param request
 * @return {Map<String, String>}
 */
public static Map<String, String> toMap(HttpServletRequest request) {
    Map<String, String> params = new HashMap<String, String>();
    Map<String, String[]> requestParams = request.getParameterMap();
    for (Iterator<String> iter = requestParams.keySet().iterator(); iter.hasNext();) {
        String name = (String) iter.next();
        String[] values = (String[]) requestParams.get(name);
        String valueStr = "";
        for (int i = 0; i < values.length; i++) {
            valueStr = (i == values.length - 1) ? valueStr + values[i] : valueStr + values[i] + ",";
        }
        // 乱码解决，这段代码在出现乱码时使用。
        // valueStr = new String(valueStr.getBytes("ISO-8859-1"), "utf-8");
        params.put(name, valueStr);
    }
    return params;
}

```

```

import com.jpay.ext.kit.StrKit;
/**
 * @author Javen
 * 2017年5月20日
 */
public class AliPayApiConfig implements Serializable {
    private static final long serialVersionUID =
-4736760736935998953L;

    private String privateKey;
    private String alipayPublicKey;
    private String appId;
    private String serviceUrl;
    private String charset;
    private String signType;
    private String format;
    private AlipayClient alipayClient;

    private AliPayApiConfig() {
    }

    public static AliPayApiConfig New() {
        return new AliPayApiConfig();
    }

    public AliPayApiConfig build(){
        this.alipayClient = new
DefaultAlipayClient(getServiceUrl(), getAppId(), getPrivateKey(),
getFormat(), getCharset(), getAlipayPublicKey(),getSignType());
        return this;
    }

    public String getPrivateKey() {
        if (StrKit.isBlank(privateKey))
            throw new IllegalStateException("privateKey 未被赋
值");
        return privateKey;
    }
}

```

```
    public AliPayApiConfig setAlipayPublicKey(String  
alipayPublicKey) {  
        if (StrKit.isBlank(alipayPublicKey))  
            throw new IllegalArgumentException("alipayPublicKey  
值不能为 null");  
        this.alipayPublicKey = alipayPublicKey;  
        return this;  
    }  
  
    public String getAppId() {  
        if (StrKit.isBlank(appId))  
            throw new IllegalStateException("appId 未被赋值");  
        return appId;  
    }  
  
    public AliPayApiConfig setAppId(String appId) {  
        if (StrKit.isBlank(appId))  
            throw new IllegalArgumentException("appId 值不能为  
null");  
        this.appId = appId;  
        return this;  
    }  
  
    public String getServiceUrl() {  
        if (StrKit.isBlank(serviceUrl))  
            throw new IllegalStateException("serviceUrl 未被赋  
值");  
        return serviceUrl;  
    }  
  
    public AliPayApiConfig setServiceUrl(String serviceUrl) {  
        if (StrKit.isBlank(serviceUrl))  
            serviceUrl = "https://openapi.alipay.com/gateway.do";  
        this.serviceUrl = serviceUrl;  
        return this;  
    }  
  
    public String setCheck() {
```

```

        signType = "RSA2";
    }
    return signType;
}

public AliPayApiConfig setSignType(String signType) {
    if (StrKit.isBlank(signType))
        signType = "RSA2";
    this.signType = signType;
    return this;
}

public String getFormat() {
    if (StrKit.isBlank(format))
        format = "json";
    return format;
}

public AlipayClient getAlipayClient() {
    if (alipayClient == null)
        throw new IllegalStateException("alipayClient 未被初始
化");
    return alipayClient;
}
}

```

具体实现的Demo

GitChat

```

package com.ijpay.controller.alipay;

import java.util.Map;

import com.alibaba.fastjson.JSONObject;
import com.alipay.api.AlipayApiException;
import
com.alipay.api.domain.AlipayDataDatasherviceBillDownloadurlQueryMo

```

```

import com.alipay.api.response.AlipayTradeCreateResponse;
import com.jfinal.kit.Prop;
import com.jfinal.kit.PropKit;
import com.jfinal.log.Log;
import com.jpay.alipay.AliPayApi;
import com.jpay.alipay.AliPayApiConfig;
import com.jpay.util.StringUtils;
import com.jpay.vo.AjaxResult;
/**
 * @Email javen205@126.com
 * @author Javen
 */
public class AliPayController extends AliPayApiController {
    private Log log = Log.getLog(AliPayController.class);

    private final Prop prop = PropKit.use("alipay.properties");
    private String charset = "UTF-8";
    private String private_key = prop.get("privateKey");
    private String alipay_public_key = prop.get("publicKey");
    private String service_url = prop.get("serverUrl");
    private String app_id = prop.get("appId");
    private String sign_type = "RSA2";
    private String notify_domain = prop.get("domain");

    private AjaxResult result = new AjaxResult();

    @Override
    public AliPayApiConfig getApiConfig() {
        AliPayApiConfig aliPayApiConfig = AliPayApiConfig.New()
            .setAppId(app_id)
            .setAlipayPublicKey(alipay_public_key)
            .setCharset(charset)
            .setPrivateKey(private_key)
            .setServiceUrl(service_url)
            .setSignType(sign_type)
            .build();
        return aliPayApiConfig;
    }
}

```

```

        model.setOutTradeNo(StringUtils.getOutTradeNo());
        model.setTimeoutExpress("30m");
        model.setTotalAmount("0.01");
        model.setPassbackParams("callback params");
        model.setProductCode("QUICK_MSECURITY_PAY");
        String orderInfo = AliPayApi.startAppPayStr(model,
notify_domain + "/alipay/app_pay_notify");
        result.success(orderInfo);
        renderJson(result);

    } catch (AlipayApiException e) {
        e.printStackTrace();
        result.addError("system error");
    }
}

/**
 * Wap支付
 */
public void wapPay() {
    String body = "我是测试数据-By Javen";
    String subject = "Javen Wap支付测试";
    String totalAmount = "1";
    String passbackParams = "1";
    String returnUrl = notify_domain + "/alipay/return_url";
    String notifyUrl = notify_domain + "/alipay/notify_url";

    AlipayTradeWapPayModel model = new
AlipayTradeWapPayModel();
    model.setBody(body);
    model.setSubject(subject);
    model.setTotalAmount(totalAmount);
    model.setPassbackParams(passbackParams);
    String outTradeNo = StringUtils.getOutTradeNo();
    System.out.println("wap outTradeNo">"+outTradeNo);
    model.setOutTradeNo(outTradeNo);
    model.setProductCode("QUICK_WAP_PAY");

```



```

        String totalAmount = "88.88";
        String outTradeNo =StringUtils.getOutTradeNo();
        log.info("pc outTradeNo>" +outTradeNo);

        String returnUrl = notify_domain +
"/alipay/return_url";
        String notifyUrl = notify_domain +
"/alipay/notify_url";
        AlipayTradePayModel model = new
AlipayTradePayModel();

        model.setOutTradeNo(outTradeNo);
        model.setProductCode("FAST_INSTANT_TRADE_PAY");
        model.setTotalAmount(totalAmount);
        model.setSubject("Javen PC支付测试");
        model.setBody("Javen IJPay PC支付测试");

        AliPayApi.tradePage(getResponse(),model , notifyUrl,
returnUrl);
        renderNull();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

GitChat

```

/**
 * 条形码支付
 * https://doc.open.alipay.com/docs/doc.htm?
spm=a219a.7629140.0.0.Yhpibd&
 * treeId=194&articleId=105170&docType=1#s4
 */
public void tradePay() {
    String authCode = getPara("auth_code");
    String subject = "Javen 支付宝条形码支付测试";
    String totalAmount = "100";
}

```

```
}  
}
```

```
/**  
 * 声波支付  
 * https://doc.open.alipay.com/docs/doc.htm?  
treeId=194&articleId=105072&docType=1#s2  
 */  
public void tradeWavePay() {  
    String authCode = getPara("auth_code");  
    String subject = "Javen 支付宝声波支付测试";  
    String totalAmount = "100";  
    String notifyUrl = notify_domain + "/alipay/notify_url";  
  
    AlipayTradePayModel model = new AlipayTradePayModel();  
    model.setAuthCode(authCode);  
    model.setSubject(subject);  
    model.setTotalAmount(totalAmount);  
    model.setOutTradeNo(StringUtils.getOutTradeNo());  
    model.setScene("wave_code");  
    try {  
        String resultStr =  
AliPayApi.tradePay(model,notifyUrl);  
        renderText(resultStr);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
/**  
 * 扫码支付  
 */  
public void tradePrecreatePay() {  
    String subject = "Javen 支付宝扫码支付测试";  
    String totalAmount = "86";  
    String outTradeNo = "100";  
}
```

```

JSONObject.parseObject(resultStr);
        String qr_code =
jsonObject.getJSONObject("alipay_trade_precreate_response").getSt
ring("qr_code");
        renderText(qr_code);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * 单笔转账到支付宝账户
 * https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.54Ty29&
 * treeId=193&articleId=106236&docType=1
 */
public void transfer() {
    boolean isSuccess = false;
    String total_amount = "66";
    AlipayFundTransToaccountTransferModel model = new
AlipayFundTransToaccountTransferModel();
    model.setOutBizNo(StringUtils.getOutTradeNo());
    model.setPayeeType("ALIPAY_LOGONID");
    model.setPayeeAccount("abpkvd0206@sandbox.com");
    model.setAmount(total_amount);
    model.setPayerShowName("测试退款");
    model.setPayerRealName("沙箱环境");
    model.setRemark("javen测试单笔转账到支付宝");

    try {
        isSuccess = AliPayApi.transfer(model);
    } catch (Exception e) {
        e.printStackTrace();
    }
    renderJson(isSuccess);
}

...

```

```

    }
}

/**
 * 退款
 */
public void tradeRefund() {

    try {
        AlipayTradeRefundModel model = new
AlipayTradeRefundModel();
        //      model.setOutTradeNo("042517111114931");
        //      model.setTradeNo("2017042521001004200200236813");
        model.setOutTradeNo("081014283315023");
        model.setTradeNo("2017081021001004200200273870");
        model.setRefundAmount("86.00");
        model.setRefundReason("正常退款");
        String resultStr = AliPayApi.tradeRefund(model);
        renderText(resultStr);
    } catch (AlipayApiException e) {
        e.printStackTrace();
    }
}

/**
 * 交易查询
 */
public void tradeQuery() {
    try {
        AlipayTradeQueryModel model = new
AlipayTradeQueryModel();
        model.setOutTradeNo("081014283315023");
        model.setTradeNo("2017081021001004200200273870");

        boolean isSuccess = AliPayApi.isTradeQuery(model);
        renderText(isSuccess);
    }
}

```

GitChat

```

        String resultStr =
AliPayApi.tradeQuery(model).getBody();
        renderText(resultStr);
    } catch (AlipayApiException e) {
        e.printStackTrace();
    }
}
/**
 * 创建订单
 * {"alipay_trade_create_response":
{"code":"10000","msg":"Success","out_trade_no":"081014283315033",
"trade_no":"2017081021001004200200274066"},"sign":"ZagfFZntf0looj
ZzdrBNnHhenhyRrsXwHLBNt1Z/dBbx7cF1o7SZQrzNjRHHmVypHKuCmYifikZIQbN
NrFJauSuhT4MQkBJE+YGPdThqDf4Ajdsv3JEyAM3TR/Xm5gUOpzCY7w+RZzkHevsT
d4cjKeGM54GBh0hQH/gSyhs4pEN3lRWopqcKkrkOGZPcmunkbrUAF7+AhKGUpK+Aq
Dw4xmKFuVChDKaRdnhM6/yVsezJFXzlQeVgFjbfiWqULxBXq1gqicntyUxvRygKA+
5zDTqE5Jj3XRDjVFIDBe0BAnM+u03fUP489wV5V5apyI449RWeybLg08Wo+jUmeOu
XOA=="}
 */
public void tradeCreate(){
    String outTradeNo = getPara("out_trade_no");

    String notifyUrl = notify_domain+ "/alipay/notify_url";

    AlipayTradeCreateModel model = new
AlipayTradeCreateModel();
    model.setOutTradeNo(outTradeNo);
    model.setTotalAmount("88.88");
    model.setBody("Body");
    model.setSubject("Javen 测试统一收单交易创建接口");
    model.setBuyerLogonId("abpkvd0206@sandbox.com");//买家支付
    宝账号, 和buyer_id不能同时为空
    try {

        AlipayTradeCreateResponse response =
AliPayApi.tradeCreate(model,notifyUrl);
        renderJson(response.getBody());
    } catch (AlipayApiException e) {

```

```

        boolean isSuccess = AliPayApi.isTradeCancel(model);
        renderJson(isSuccess);
    } catch (AlipayApiException e) {
        e.printStackTrace();
    }
}

/**
 * 关闭订单
 */
public void tradeClose() {
    String outTradeNo = getPara("out_trade_no");
    String tradeNo = getPara("trade_no");
    try {
        AlipayTradeCloseModel model = new
AlipayTradeCloseModel();
        model.setOutTradeNo(outTradeNo);

        model.setTradeNo(tradeNo);

        String resultStr =
AliPayApi.tradeClose(model).getBody();
        renderText(resultStr);
    } catch (AlipayApiException e) {
        e.printStackTrace();
    }
}

/**
 * 结算
 */
public void tradeOrderSettle() {
    String tradeNo = getPara("trade_no");//支付宝订单号
    try {
        AlipayTradeOrderSettleModel model = new
AlipayTradeOrderSettleModel();
        model.setOutRequestNo(StringUtil.getRandomString(16));
        model.setOutTradeNo(tradeNo);
    } catch (AlipayApiException e) {
        e.printStackTrace();
    }
}

```

```

AliPayApi.toMap(getRequest());
        for (Map.Entry<String, String> entry :
map.entrySet()) {
            System.out.println(entry.getKey() + " = " +
entry.getValue());
        }

        boolean verify_result =
AlipaySignature.rsaCheckV1(map, alipay_public_key, charset,
sign_type);

        if (verify_result) {// 验证成功
            // TODO 请在这里加上商户的业务逻辑程序代码
            System.out.println("return_url 验证成功");
            renderText("success");
            return;
        } else {
            System.out.println("return_url 验证失败");
            // TODO
            renderText("failure");
            return;
        }
    } catch (AlipayApiException e) {
        e.printStackTrace();
        renderText("failure");
    }
}

public void notify_url() {
    try {
        // 获取支付宝POST过来反馈信息
        Map<String, String> params =
AliPayApi.toMap(getRequest());

        for (Map.Entry<String, String> entry :
params.entrySet()) {
            System.out.println(entry.getKey() + " = " +
entry.getValue());
        }
    }
}

```

```

        // TODO
        renderText("failure");
        return;
    }
} catch (AlipayApiException e) {
    e.printStackTrace();
    renderText("failure");
}
}

//=====其实异步通知实现的方法都一样 但是通知中无法区分支付的方式(没有提供支付方式的参数)=====
=====
/**
 * App支付支付回调通知
 * https://doc.open.alipay.com/docs/doc.htm?treeId=54&articleId=106370&docType=1#s3
 */
public void app_pay_notify() {
    try {
        // 获取支付宝POST过来反馈信息
        Map<String, String> params =
AliPayApi.toMap(getRequest());
        for (Map.Entry<String, String> entry :
params.entrySet()) {
            System.out.println(entry.getKey() + " = " +
entry.getValue());
        }
        // 切记alipaypublickey是支付宝的公钥，请去open.alipay.com
对应应用下查看。
        // boolean AlipaySignature.rsaCheckV1(Map<String,
String> params,
        // String publicKey, String charset, String
sign_type)
        boolean flag = AlipaySignature.rsaCheckV1(params,
alipay_public_key, charset,
sign_type);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```



```

    }
    /**
     * 扫码支付通知
     */
    public void precreate_notify_url(){
        try {
            Map<String, String> map =
AliPayApi.toMap(getRequest());
            for (Map.Entry<String, String> entry :
map.entrySet()) {
                System.out.println(entry.getKey()+" =
"+entry.getValue());
            }
            boolean flag = AlipaySignature.rsaCheckV1(map,
alipay_public_key, charset,
                sign_type);
            if (flag) {
                // TODO
                System.out.println("precreate_notify_url
success");
                renderText("success");
                return;
            } else {
                // TODO
                System.out.println("precreate_notify_url
failure");
                renderText("failure");
            }
        } catch (AlipayApiException e) {
            e.printStackTrace();
            renderText("failure");
        }
    }
}
}

```

```
import android.os.Message;
import android.text.TextUtils;
import android.util.Log;

import com.alipay.sdk.app.PayTask;
import com.jpay.JPay.JPayListener;
import com.jpay.weixin.WeixinPay;

import java.util.Map;

/**
 * Created by Javen on 2016/11/21.
 */

public class Alipay {
    private static Alipay mAlipay;
    private Activity mContext;
    private JPayListener mJPayListener;

    //支付失败
    public static final int PAY_ERROR = 0x001;
    //支付网络连接出错
    public static final int PAY_NETWORK_ERROR = 0x002;
    //支付结果解析错误
    public static final int RESULT_ERROR = 0x003;
    //正在处理中
    public static final int PAY_DEALING = 0x004;
    //其它支付错误
    public static final int PAY_OTHER_ERROR = 0x006;
    //支付参数异常
    public static final int PAY_PARAMETERS_ERROE = 0x007;

    private Alipay(Activity context) {
        mContext = context;
    }

    public static Alipay getInstance(Activity context) {
```

```

        Runnable payRunnable = new Runnable() {
            @Override
            public void run() {
                PayTask alipay = new PayTask(mContext);
                Map<String, String> result =
alipay.payV2(orderInfo, true);
                Message msg = new Message();
                msg.obj = result;
                mHandler.sendMessage(msg);
            }
        };
        Thread payThread = new Thread(payRunnable);
        payThread.start();
    }

    private Handler mHandler = new
Handler(Looper.getMainLooper()) {
        @SuppressWarnings("unchecked")
        public void handleMessage(Message msg) {
            PayResult payResult = new PayResult((Map<String,
String>) msg.obj);

            Log.e("aliPay call ", payResult.toString());

            String resultStatus = payResult.getResultStatus();
            if (mJPayListener == null) {
                return;
            }
            // https://doc.open.alipay.com/doc2/detail.htm?
spm=a219a.7629140.0.0.xN1NnL&treeId=204&articleId=105302&docType=
1

            if (payResult == null) {
                mJPayListener.onPayError(RESULT_ERROR, "结果解析错
误");

                return;
            }
            if (TextUtils.equals(resultStatus, "0000")) {

```

```

    } else if(TextUtils.equals(resultStatus, "4000")) {
        //支付错误
        mJPayListener.onPayError(PAY_ERROR,"订单支付失败");
    }else {

mJPayListener.onPayError(PAY_OTHER_ERROR,resultStatus);
    }
    };
}

```

使用JPay 唤起客户端支付

先使用从服务端获取订单信息再使用JPay 封装的方法唤起客户端支付
服务端获取 orderInfo

```

/**
 * app支付
 */
public void appPay() {
    try {
        AlipayTradeAppPayModel model = new
        AlipayTradeAppPayModel();
        model.setBody("我是测试数据-By Javen");
        model.setSubject("App支付测试-By Javen");
        model.setOutTradeNo(StringUtils.getOutTradeNo());
        model.setTimeoutExpress("30m");
        model.setTotalAmount("0.01");
        model.setPassbackParams("callback params");
        model.setProductCode("QUICK_MSECURITY_PAY");
        String orderInfo = AliPayApi.startAppPayStr(model,
        notify_domain + "/alipay/app_pay_notify");
        result.success(orderInfo);
        renderJson(result);
    }
}

```

```
    }

    @Override
    public void onPayError(int error_code, String
message) {

    }

    @Override
    public void onPayCancel() {

    }

});
```

APP支付报错ALI40247处理方案

ALI40247的错误

有2种情况 1、没有权限 2、签名失败

可以查看论坛提供的这个[解决方案](#)

AL38173的错误一般是参数问题造成的,建议检查请求参数,对照文档查看,比如:参数少了、多了、乱码、名称不对,还有必传参数是否都请求提交给支付宝了等等.

如果在网上查阅资料还未解决,可以寻求阿里的在线支持【[支持中心](#)】

支付视频

[视频资源更新中...](#)