

1.支付

现代生活，什么付款方式最常见?不是现金，也不是银行卡，而是手机支付。支付宝和微信支付是当今社会必备的付款方式，所以在中国，出门带手机，就什么都可以做到，而这个生活上的变化离不开互联网的发展以及辛勤的开发人员的努力。而对于众多的app或web网站，一个二维码扫码支付功能是必备的，那么如何实现支付功能呢？让我们一起来探索一下吧~

2.支付宝支付

支付宝官方网站：<https://www.alipay.com/>。

支付宝（中国）网络技术有限公司是国内的第三方支付平台，致力于提供“简单、安全、快速”的支付解决方案。支付宝公司从2004年建立开始，始终以“信任”作为产品和服务的核心。旗下有“支付宝”与“支付宝钱包”两个独立品牌。自2014年第二季度开始成为当前全球最大的移动支付厂商。

支付分类

快捷支付

快捷支付是指支付机构与银行合作直连，形成一个高效、安全、专用（消费）的支付方式

手机支付

2008年开始支付宝开始介入手机支付业务，2009年推出首个独立移动支付客户端，2013年初更名为“支付宝钱包”，并于2013年10月成为与“支付宝”并行的独立品牌；

二维码支付

2010年10月，支付宝推出国内首个二维码支付技术，帮助电商从线上向线下延伸发展空间。

使用方式：用户在“支付宝钱包”内，点击“扫一扫”，对准二维码按照提示就能完成。

声波支付

2013年4月12日，支付宝与合作方青岛易触联合推出全球首个声波售货机。市面尚无同类支付技术商用。

使用方式：用户在支持声波支付的售货机等场景下，选择商品，然后在“支付宝钱包”内点击“当面付”。按照提示完成支付。

NFC支付

2012年7月31日，支付宝推出利用NFC、LBS等技术的新客户端。随后这一技术方案得到进一步改进。

2014年4月28日，支付宝钱包8.1版支持NFC功能，用户可以用于向北京公交一卡通进行充值。

使用方式：将公交卡等放置在具有NFC的安卓手机后，即可查询公交卡余额以及充值。

值得注意的是，支付宝移动支付均为远程在线支付方案，NFC在当中的作用为“近场握手、远程支付”。与统称的NFC略有差异。

iptv支付

2012年3月29日，华数传媒与支付宝推出互联网电视支付，实现3秒支付。

使用方式：注册为华数会员，并关注服务窗号。使用“支付宝钱包”扫描电视上的二维码，完成支付。

指纹支付

2014年7月16日，移动支付平台支付宝钱包宣布试水指纹支付服务。

刷脸支付

2018年12月13日，支付宝宣布推出一款全新的刷脸支付产品——“蜻蜓”，直接将刷脸支付的接入成本降低80%。

3.沙箱(沙盒)

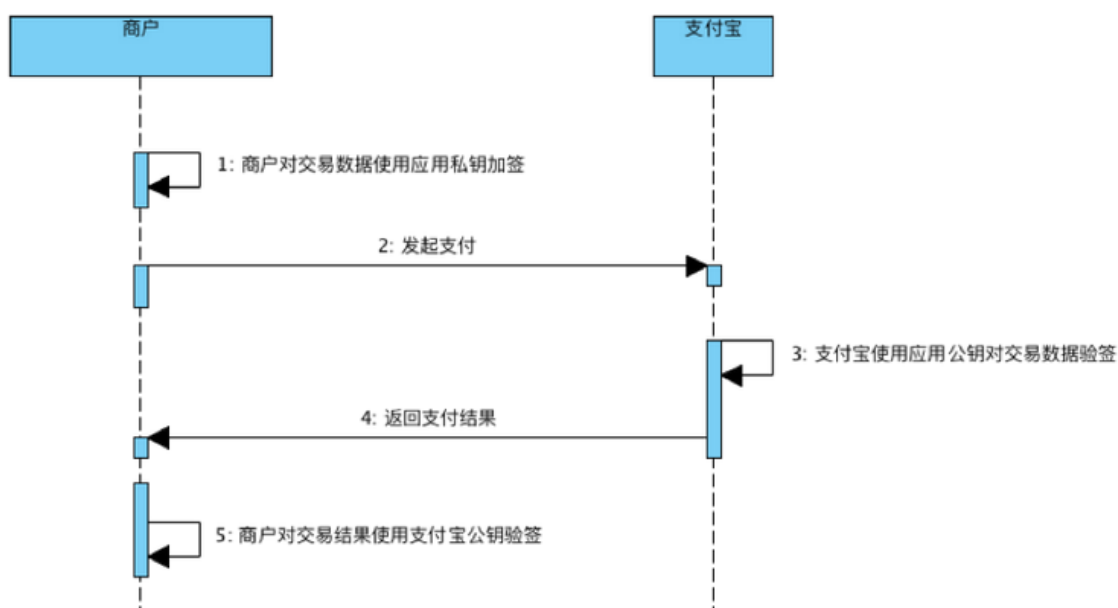
Sandboxie(又叫沙箱、沙盘)即是一个虚拟系统程序，允许你在沙盘环境中运行浏览器或其他程序，因此运行所产生的变化可以随后删除。它创造了一个类似沙盒的独立作业环境，在其内部运行的程序并不能对硬盘产生永久性的影响。在网络安全中，沙箱指在隔离环境中，用以测试不受信任的文件或应用程序等行为的工具。

沙箱是一种按照安全策略限制程序行为的执行环境。早期主要用于测试可疑软件等，比如黑客们为了试用某种病毒或者不安全产品，往往可以将它们在沙箱环境中运行。

经典的沙箱系统的实现途径一般是通过拦截系统调用，监视程序行为，然后依据用户定义的策略来控制 and 限制程序对计算机资源的使用，比如改写注册表，读写磁盘等。

4.支付流程

为了保证交易双方（商户和支付宝）的身份和数据安全，开发者在调用接口前，需要配置双方密钥，对交易数据进行双方校验。密钥包含应用私钥（APP_PRIVATE_KEY）和应用公钥（APP_PUBLIC_KEY）。生成密钥后，开发者需要在开放平台开发者中心进行密钥配置，配置完成后可以获取支付宝公钥（ALIPAY_PUBLIC_KEY）。密钥的配置旨在对交易数据进行双方校验。具体流程如下图所示：



说明：

支付宝开放平台 SDK 封装了签名和验签过程，只需配置账号及密钥参数。

- 应用公钥（商户自身的 RSA/RSA2 公钥）：支付宝使用该公钥验证该交易是商户发起。
- 支付宝公钥（支付宝的 RSA/RSA2 公钥）：商户使用该公钥验证该结果是支付宝返回的。

注意：

- 由于同步返回的不可靠性，支付结果必须以异步通知或查询接口返回为准，不能依赖同步跳转。
- 商户系统接收到异步通知以后，必须通过验签（验证通知中的 sign 参数）来确保支付通知是由支付宝发送的。
- 接收到异步通知并验签通过后，一定要检查通知内容，包括通知中的 app_id、out_trade_no、total_amount 是否与请求中的一致，并根据 trade_status 进行后续业务处理。
- 在支付宝端，partnerId 与 out_trade_no 唯一对应一笔单据，商户端保证不同次支付 out_trade_no 不可重复；若重复，支付宝会关联到原单据，基本信息一致的情况下会以原单据为准进行支付。

补充:SDK和API的区别

SDK全称software development kit，软件开发工具包。

API就是可以轻松实现和其他软件的交互。

API是数据接口，SDK相当于开发集成工具环境。要在SDK的环境下调用API。

API是接口对接接口的过程，SDK不仅提供开发环境，还提供很多的API

扫码支付接入流程

<https://opendocs.alipay.com/open/194/106078>

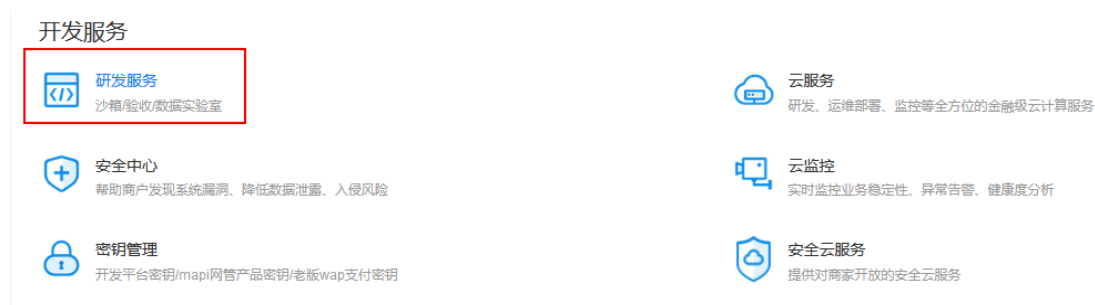
5.支付宝支付准备工作

正式的支付功能上线，需要使用企业的信息，所以我们这里使用支付宝提供的沙箱功能实现测试，如何访问沙箱呢？下面看一下实现步骤：

访问支付宝首页，选择角色：“我是开发者”



使用手机上的支付宝，扫码登录到系统。登录后选择左上角“控制台”，然后在页面下方有“开发服务”



沙箱的使用步骤：

研发服务

沙箱环境

沙箱应用

沙箱账号

沙箱工具

云验收

产品验收记录

活动验收记录

验收人设置


沙箱应用

信息配置

必看部分

APPID	2021000116668413	支付宝分配的程序id
支付宝网关	https://openapi.alipaydev.com/gateway.do	支付宝分配的网关地址
RSA2(SHA256)密钥(推荐)	设置查看	这里的密钥需要人为配置

选看部分（部分接口使用，详见文档）

应用名称	沙箱测试应用
应用图标	
商户UID	2088621954859603

加签管理

1 加签内容配置

2 加签内容完成

×

选择加签模式：

☐ 公钥证书 ☒ 公钥

注：

资金支出类接口不支持公钥的方式进行加签。

填写公钥字符：

支付宝会用SHA256withRsa算法进行接口调用时的验签（不限制密钥长度）。使用 [支付宝密钥生成器](#)或[OpenSSL\(第三方工具\)](#)生成密钥

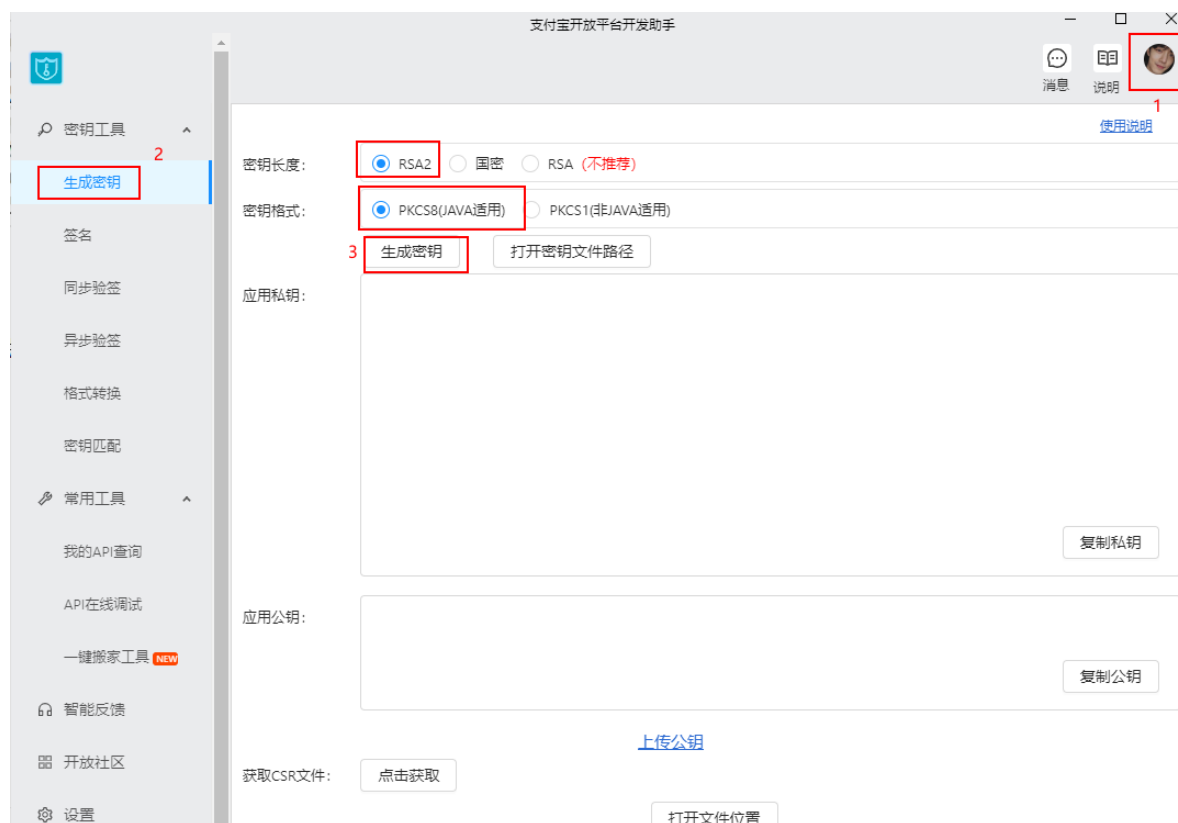
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA...Z
AUU5R...
O0uvILB...
+9vR...
/NS)...
B4...erTid
/rmb...
-----END PUBLIC KEY-----

保存设置

支付宝开放平台开发助手



运行程序，生成密钥：



把这里的“应用公钥”复制到“加签管理”中的公钥位置。设置完毕后，密钥位置会变成“设置/查看”

RSA2(SHA256)密钥(推荐) [设置/查看](#)

步骤3:下载沙箱钱包进行测试，注意这里必须使用沙箱版钱包进行测试，账户信息已给定。**此软件只支持安卓系统**

① 沙箱钱包只支持扫一扫、付款码、门店详情页功能，其余功能不提供



目前沙箱钱包仅提供Android版本，可点击[下载](#)

步骤4：测试账户的信息

商家信息

商家账号 ukc[REDACTED]m

商户UID 2[REDACTED]503

登录密码 111111

账户余额 99.55 [充值](#) [取现](#)

买家信息

买家账号 [REDACTED].com

登录密码 [REDACTED]

支付密码 111111

用户名称 沙箱环境

证件类型 身份证(IDENTITY_CARD)

证件号码 770213195301172046

账户余额 99898.85 [充值](#) [取现](#)

6.代码开发

6.1 支付工具类:AlipayConfig.java

```
public class AlipayConfig {  
  
    //↓↓↓↓↓↓↓↓↓↓请在這裡配置您的基本信息↓↓↓↓↓↓↓↓↓↓↓↓↓↓  
  
    // 应用ID,您的APPID,收款账号既是您的APPID对应支付宝账号  
    public static String app_id = "";  
  
    // 商户私钥,您的PKCS8格式RSA2私钥  
    public static String merchant_private_key = "";  
  
    // 支付宝公钥  
    public static String alipay_public_key = "";  
  
    // 服务器异步通知页面路径  
    //需http://格式的完整路径,不能加?id=123这类自定义参数,必须外网可以正常访问
```

```

public static String notify_url = "";

// 页面跳转同步通知页面路径
//需http://格式的完整路径，不能加?id=123这类自定义参数，必须外网可以正常访问
public static String return_url = "";

// 签名方式
public static String sign_type = "RSA2";

// 字符编码格式
public static String charset = "utf-8";

// 支付宝网关,注意这些使用的是沙箱的支付宝网关，与正常网关的区别是多了dev
public static String gatewayUrl =
"https://openapi.alipaydev.com/gateway.do";

// 支付宝网关
public static String log_path = "C:\\\\";

//↑↑↑↑↑↑↑↑↑↑请在这里配置您的基本信息↑↑↑↑↑↑↑↑↑↑↑↑↑↑

/**
 * 写日志，方便测试（看网站需求，也可以改成把记录存入数据库）
 * @param sword 要写入日志里的文本内容
 */
public static void logResult(String sword) {
    FileWriter writer = null;
    try {
        writer = new FileWriter(log_path + "alipay_log_" +
System.currentTimeMillis()+".txt");
        writer.write(sword);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (writer != null) {
            try {
                writer.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
}

```

6.2.添加maven依赖

```

<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.3.3.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>

<dependencies>
<!--springBoot依赖包-->

```

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- 支付功能SDK -->
<dependency>
  <groupId>com.alipay.sdk</groupId>
  <artifactId>alipay-sdk-java</artifactId>
  <version>4.10.124.ALL</version>
</dependency>
<!--springBoot支持jsp-->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
</dependencies>

```

6.3 修改application.properties属性文件

```

#页面默认前缀目录
spring.mvc.view.prefix=/
#响应页面默认后缀
spring.mvc.view.suffix=.jsp
#修改访问的端口号
server.port=8081
#设置访问的项目路径
server.servlet.context-path=/

```

6.4 创建页面

```

<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>支付宝网站支付</title>
<style>
* {
    margin: 0;
    padding: 0;
}

ul, ol {
    list-style: none;
}

body {
    font-family: "Helvetica Neue", Helvetica, Arial, "Lucida Grande",
    sans-serif;
}

.tab-head {

```



```
        margin-left: 120px;
        margin-bottom: 10px;
    }

    .tab-content {
        clear: left;
        display: none;
    }

    h2 {
        border-bottom: solid #02aaf1 2px;
        width: 200px;
        height: 25px;
        margin: 0;
        float: left;
        text-align: center;
        font-size: 16px;
    }

    .selected {
        color: #FFFFFF;
        background-color: #02aaf1;
    }

    .show {
        clear: left;
        display: block;
    }

    .hidden {
        display: none;
    }

    .new-btn-login-sp {
        padding: 1px;
        display: inline-block;
        width: 75%;
    }

    .new-btn-login {
        background-color: #02aaf1;
        color: #FFFFFF;
        font-weight: bold;
        border: none;
        width: 100%;
        height: 30px;
        border-radius: 5px;
        font-size: 16px;
    }

    #main {
        width: 100%;
        margin: 0 auto;
        font-size: 14px;
    }

    .red-star {
        color: #f00;
```

```
        width: 10px;
        display: inline-block;
    }

    .null-star {
        color: #fff;
    }

    .content {
        margin-top: 5px;
    }

    .content dt {
        width: 100px;
        display: inline-block;
        float: left;
        margin-left: 20px;
        color: #666;
        font-size: 13px;
        margin-top: 8px;
    }

    .content dd {
        margin-left: 120px;
        margin-bottom: 5px;
    }

    .content dd input {
        width: 85%;
        height: 28px;
        border: 0;
        -webkit-border-radius: 0;
        -webkit-appearance: none;
    }

    #foot {
        margin-top: 10px;
        position: absolute;
        bottom: 15px;
        width: 100%;
    }

    .foot-ul {
        width: 100%;
    }

    .foot-ul li {
        width: 100%;
        text-align: center;
        color: #666;
    }

    .note-help {
        color: #999999;
        font-size: 12px;
        line-height: 130%;
        margin-top: 5px;
        width: 100%;
    }
```

```

        display: block;
    }

    #btn-dd {
        margin: 20px;
        text-align: center;
    }

    .foot-ul {
        width: 100%;
    }

    .one_line {
        display: block;
        height: 1px;
        border: 0;
        border-top: 1px solid #eeeeee;
        width: 100%;
        margin-left: 20px;
    }

    .am-header {
        display: -webkit-box;
        display: -ms-flexbox;
        display: box;
        width: 100%;
        position: relative;
        padding: 7px 0;
        -webkit-box-sizing: border-box;
        -ms-box-sizing: border-box;
        box-sizing: border-box;
        background: #1D222D;
        height: 50px;
        text-align: center;
        -webkit-box-pack: center;
        -ms-flex-pack: center;
        box-pack: center;
        -webkit-box-align: center;
        -ms-flex-align: center;
        box-align: center;
    }

    .am-header h1 {
        -webkit-box-flex: 1;
        -ms-flex: 1;
        box-flex: 1;
        line-height: 18px;
        text-align: center;
        font-size: 18px;
        font-weight: 300;
        color: #fff;
    }
}
</style>
</head>
<body text=#000000 bgColor="#ffffff" leftMargin=0 topMargin=4>
    <header class="am-header">
        <h1>支付宝体验入口页</h1>
    </header>

```

```

<div id="main">
  <div id="tabhead" class="tab-head">
    <h2 id="tab1" class="selected" name="tab">付 款</h2>
  </div>
  <form name=alipayment action=pay method=post
    target="_blank">
    <div id="body1" class="show" name="divcontent">
      <dl class="content">
        <dt>商户订单号 : </dt>
        <dd>
          <input id="WIDout_trade_no" name="WIDout_trade_no" />
        </dd>
        <hr class="one_line">
        <dt>订单名称 : </dt>
        <dd>
          <input id="WIDsubject" name="WIDsubject" />
        </dd>
        <hr class="one_line">
        <dt>付款金额 : </dt>
        <dd>
          <input id="WIDtotal_amount" name="WIDtotal_amount" />
        </dd>
        <hr class="one_line">
        <dt>商品描述: </dt>
        <dd>
          <input id="WIDbody" name="WIDbody" />
        </dd>
        <hr class="one_line">
        <dt></dt>
        <dd id="btn-dd">
          <span class="new-btn-login-sp">
            <button class="new-btn-login" type="submit"
              style="text-align: center;">付 款</button>
          </span> <span class="note-help">如果您点击“付款”按钮，即表示
您同意该次的执行操作。</span>
        </dd>
      </dl>
    </div>
  </form>

  <div id="foot">
    <ul class="foot-ul">
      <li>版权所有 2015-2018</li>
    </ul>
  </div>
</div>
</body>
<script language="javascript">
  function GetDateNow() {
    var vNow = new Date();
    var sNow = "";
    sNow += String(vNow.getFullYear());
    sNow += String(vNow.getMonth() + 1);
    sNow += String(vNow.getDate());
    sNow += String(vNow.getHours());
    sNow += String(vNow.getMinutes());
    sNow += String(vNow.getSeconds());
    sNow += String(vNow.getMilliseconds());
  }

```

```

        document.getElementById("WIDout_trade_no").value = sNow;
        document.getElementById("WIDsubject").value = "测试";
        document.getElementById("WIDtotal_amount").value = "0.01";
    }
    GetDateNow();
</script>
</html>

```

6.5 在springBoot环境下创建支付对象

配置参数	示例值解释	获取方式/示例值
URL	支付宝网关（固定），注意沙箱时的地址不同	https://openapi.alipay.com/gateway.do
APP_ID	APPID 即创建应用后生成	详情见 创建应用并获取 APPID
APP_PRIVATE_KEY	开发者应用私钥，由开发者自己生成	详见 配置密钥
FORMAT	参数返回格式，只支持 json 格式	json（固定）
CHARSET	请求和签名使用的字符编码格式，支持 GBK和 UTF-8	开发者根据实际工程编码配置
ALIPAY_PUBLIC_KEY	支付宝公钥，由支付宝生成	详见 配置密钥
SIGN_TYPE	商户生成签名字符串所使用的签名算法类型，目前支持 RSA2 和 RSA，推荐商家使用 RSA2。	RSA2

```

@Configuration
public class BeanUtil {
    @Bean
    public AlipayClient alipayClient(){
        return new DefaultAlipayClient(AlipayConfig.gatewayUrl,
            AlipayConfig.app_id, AlipayConfig.merchant_private_key, "json",
            AlipayConfig.charset, AlipayConfig.alipay_public_key,
            AlipayConfig.sign_type);
    }

    @Bean
    public AlipayTradePagePayRequest alipayTradePagePayRequest(){
        return new AlipayTradePagePayRequest();
    }
}

```

6.6 创建controller处理支付请求

```

@Controller
public class PayController {
    @Resource
    private AlipayClient alipayClient;
    @Resource
    private AlipayTradePagePayRequest alipayTradePagePayRequest;
}

```

```

//商户订单号，商户网站订单系统中唯一订单号，必填 WIDout_trade_no
//付款金额，必填 WIDtotal_amount
//订单名称，必填 WIDsubject
//商品描述，可空 WIDbody
@RequestMapping("/pay")
public void payUtil(String WIDout_trade_no, String WIDtotal_amount, String
WIDsubject, String WIDbody, HttpServletResponse response) throws Exception{
    //1.获得初始化的AlipayClient
    //2.设置请求参数AlipayTradePagePayRequest
    alipayTradePagePayRequest.setReturnUrl(AlipayConfig.return_url);
    alipayTradePagePayRequest.setNotifyUrl(AlipayConfig.notify_url);

    alipayTradePagePayRequest.setBizContent(
        "{\\"out_trade_no\\":\\""+ WIDout_trade_no +\\"","
        + "\"total_amount\\":\\""+ WIDtotal_amount +\\"","
        + "\"subject\\":\\""+ WIDsubject +\\"","
        + "\"body\\":\\""+ WIDbody +\\"","
        + "\"product_code\\":\\"FAST_INSTANT_TRADE_PAY\\""}");
    //请求
    String result =
alipayClient.pageExecute(alipayTradePagePayRequest).getBody();
    //输出
    response.setContentType("text/html;charset=utf-8");
    response.getWriter().println(result);
}
}

```

6.7 配置异步通知的处理类

```

@Controller
public class NotifyUrlController {
    @RequestMapping("/notifyurl")
    public void notifyUrl(HttpServletRequest request, HttpServletResponse
response) throws Exception{
        //获取支付宝POST过来反馈信息
        Map<String,String> params = new HashMap<String,String>();
        Map<String,String[]> requestParams = request.getParameterMap();
        Iterator<String> iter = requestParams.keySet().iterator();
        while(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);
        }
        boolean signVerified = AlipaySignature.rsaCheckV1(params,
AlipayConfig.alipay_public_key, AlipayConfig.charset, AlipayConfig.sign_type);
        //调用SDK验证签名

        //---请在这里编写您的程序（以下代码仅作参考）---

        /* 实际验证过程建议商户务必添加以下校验：

```

```

1、需要验证该通知数据中的out_trade_no是否为商户系统中创建的订单号，
2、判断total_amount是否确实为该订单的实际金额（即商户订单创建时的金额），
3、校验通知中的seller_id（或者seller_email） 是否为out_trade_no这笔单据的对应的操作方
（有的时候，一个商户可能有多个seller_id/seller_email）
4、验证app_id是否为该商户本身。
*/
response.setContentType("text/html;charset=utf-8");
PrintWriter out = response.getWriter();
if(signVerified) { //验证成功
    //商户订单号
    String out_trade_no = request.getParameter("out_trade_no");
    //支付宝交易号
    String trade_no = request.getParameter("trade_no");
    //交易状态
    String trade_status = request.getParameter("trade_status");
    if(trade_status.equals("TRADE_FINISHED")){
        //判断该笔订单是否在商户网站中已经做过处理
        //如果没有做过处理，根据订单号（out_trade_no）在商户网站的订单系统中查到该
        笔订单的详细，并执行商户的业务程序
        //如果有做过处理，不执行商户的业务程序

        //注意：
        //退款日期超过可退款期限后（如三个月可退款），支付宝系统发送该交易状态通知
    }else if (trade_status.equals("TRADE_SUCCESS")){
        //判断该笔订单是否在商户网站中已经做过处理
        //如果没有做过处理，根据订单号（out_trade_no）在商户网站的订单系统中查到该
        笔订单的详细，并执行商户的业务程序
        //如果有做过处理，不执行商户的业务程序

        //注意：
        //付款完成后，支付宝系统发送该交易状态通知
    }

    out.println("success");

}else { //验证失败
    out.println("fail");
    //调试用，写文本函数记录程序运行情况是否正常
    //String sword = AlipaySignature.getSignCheckContentV1(params);
    //AlipayConfig.logResult(sword);
}

}

}

```

异步通知: 其实是双保险机制, 如果同步通知后没有跳转到你的网址, 可能用户关了, 可能网速慢, 即无法触发你更新订单状态为已支付的controller, 这时候异步通知就有作用了, 不过你要判断一下, 如果订单已经变为已支付, 则不必再更新一次了, 只返回给支付宝success即可, 否则他会一直异步通知你

异步通知参数说明文档:<https://opendocs.alipay.com/open/203/105286>

6.8 添加同步通知处理类

```

@Controller //回调请求
public class ReturnUrl {
    @RequestMapping("/returnUrl")

```

```

    public void returnUrl(HttpServletRequest request, HttpServletResponse
response) throws Exception{
        //获取支付宝GET过来反馈信息
        Map<String,String> params = new HashMap<String,String>();
        Map<String,String[]> requestParams = request.getParameterMap();
        Iterator<String> iter = requestParams.keySet().iterator();
        while(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);
        }

        //RSA2验证
        boolean signVerified = AlipaySignature.rsaCheckV2(params,
AlipayConfig.alipay_public_key, AlipayConfig.charset, AlipayConfig.sign_type);
        //调用SDK验证签名
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        //--请在这里编写您的程序（以下代码仅作参考）--
        if(signVerified) {
            //商户订单号
            String out_trade_no = request.getParameter("out_trade_no");
            //支付宝交易号
            String trade_no = request.getParameter("trade_no");
            //付款金额
            String total_amount = request.getParameter("total_amount");
            out.println("trade_no:"+trade_no+"<br/>out_trade_no:"+out_trade_no+"
<br/>total_amount:"+total_amount);
        }else {
            out.println("验签失败");
        }
    }
}

```

同步通知: 用于用户在支付宝页面付款完毕后自动跳转回你自己的网址, 你根据他的参数告诉用户已经支付成功, 然后你再更新你自己订单表的状态为已支付.

区别:1.同步通知是给用户看的 2.异步通知是给服务器看的

6.9 测试

访问地址:<http://localhost:8081/>

支付宝体验入口页

付款

商户订单号： 2020911133841372

订单名称： 测试

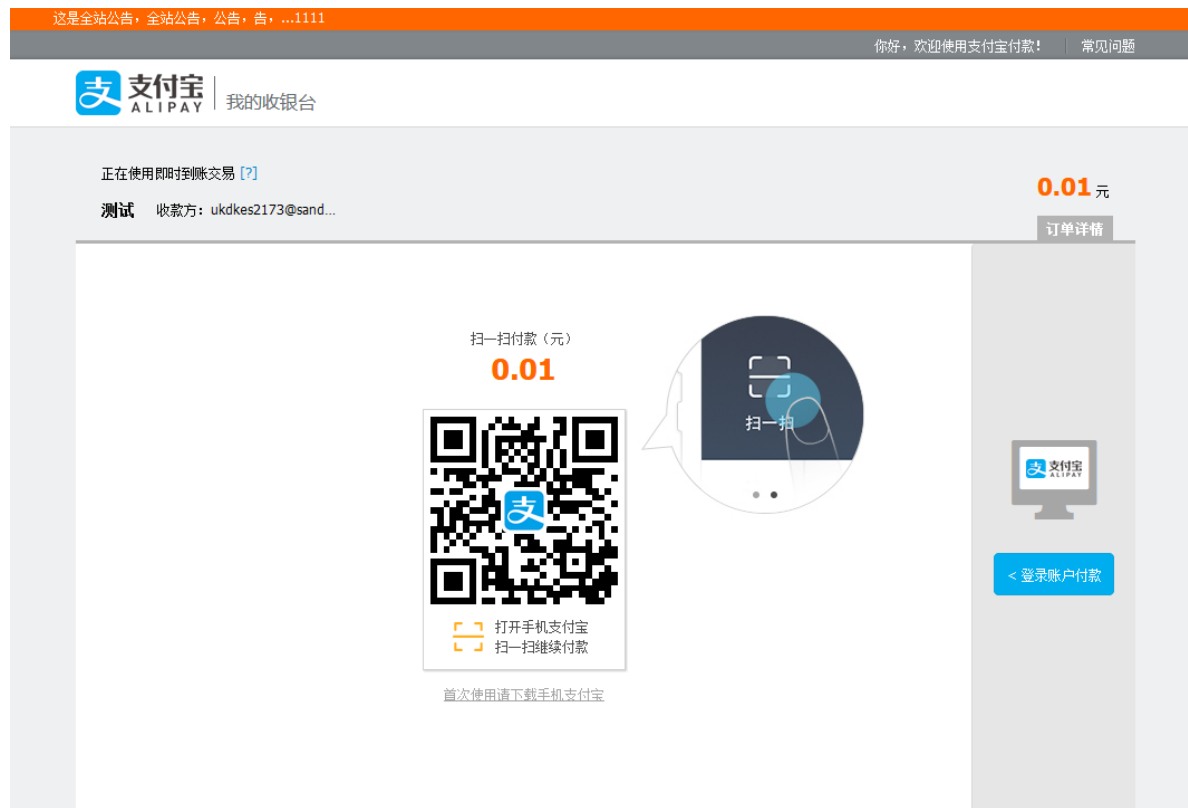
付款金额： 0.01

商品描述：

付款

如果您点击“付款”按钮，即表示您同意该次的执行操作。

跳转到支付页面:



扫码支付,支付成功后，返回订单相关信息:

```
trade_no:202091122001413260500938224
out_trade_no:2020911134517208
total_amount:0.01
```

7.手机网站支付沙箱接入注意点

- 1、手机网站支付支持沙箱接入；在沙箱调通接口后，必须在线上进行测试与验收，所有返回码及业务逻辑以线上为准；
- 2、手机网站支付只支持余额支付，不支持银行卡、余额宝等其他支付方式；
- 3、支付时，请使用沙箱买家账号支付，在登录支付宝，输入手机号的页面，点击右下角支付宝账户登录