# 1.背景

由于历史原因,旧支付系统和其他业务系统出现严重耦合,特别是跟支付相关的pay\_order表加入了很多业务数据.同时系统代码编写也比较混乱,不利系统扩展与升级.因此,需要对现有支付系统进行重写.重写后,新支付系统将做为独立的应用系统存在,只负责与外部支付系统进行对接,不依赖其他内部其他系统,也不会处理与内部系统业务相关的逻辑.

# 2.系统规划

由于当前有支付相关的需求需要开发,故重写的第一个版本只完成本次需求所需功能,即扫码支付,重点还是在系统可扩展性方面考虑,后面会根据业务需求逐步对接app支付,小程序支付,h5支付等.同时第一个版本与内部系统对接时,暂不支持签名验签功能,后面考虑会引入RSA2做安全方面升级.

# 3.集成

本文档仅针对SpringCloud微服务集成支付服务,非SpringCloud也可以集成,但是暂时不建议.

|  |
| --- |
| <dependency>  <groupId>**com.sandu**</groupId>  <artifactId>**sandu-cloud-pay-api**</artifactId>  <version>**0.0.1-SNAPSHOT**</version>  </dependency> |

# 4.相关接口

## 4.1 微信/支付宝支付

### 4.1.1付款代码

|  |
| --- |
| **public** **void** testPay**()** **{**  PayParamDto payParamDto **=** **new** PayParamDto**();**  //本系统自己生成的交易号,可以通过此交易号去支付服务查询相关交易  payParamDto**.**setIntenalTradeNo**(**"123456"**);**  //交易业务描述  payParamDto**.**setTradeDesc**(**"度币充值"**);**  //支付金额,单位为分  payParamDto**.**setTotalFee**(**1L**);**  //支付方式: 微信扫码支付,h5支付,app支付,小程序支付,支付宝扫码支付,h5支付,app支付  payParamDto**.**setPayMethod**(**PayMethodContants.WX\_SCANCODE\_PAY**);**  //客户端ip  payParamDto**.**setIp**(**"127.0.0.1"**);**  //支付结果回调通知地址  payParamDto**.**setNotifyUrl**(**"http://127.0.0.1:8089/v1/pay/callback/test/notify"**);**  //客户端id:服务id或者应用id,需要在支付服务注册  payParamDto**.**setClientId**(**"001"**);**  //第三方收款应用id  payParamDto**.**setAppId**(**"2016092301955254"**);**  //需要保证上述参数设置完成后再生成签名  payParamDto**.**setSign**(**payParamDto**.**createSign**(**"123456"**));**  //返回结果为json格式  String result **=** payServiceClient**.**doPay**(**payParamDto**);**  //本应用系统做业务处理  System**.**out**.**println**(**result**);**  **}** |

### 4.1.2返回结果

|  |  |
| --- | --- |
| **支付类型** | **返回数据** |
| 微信扫码支付 | **{**  //二维码链接  "qrCodeUrl"**:**"weixin://wxpay/bizpayurl?pr=tyYZgog"**,**  //支付系统交易号  "payTradeNo"**:**"20190520155614225995603277531902"  **}** |
| 微信h5支付  (参照微信h5支付接口, 根据返回结果,调起h5付款界面) | **{**  "payTradeNo"**:** "20190528154023239656095147723977"**,**  "mwebUrl"**:** "https://wx.tenpay.com/cgi-bin  /mmpayweb-bin/checkmweb?prepay\_id=wx28154022657818  793c55444a1182097400&package=278616718"  **}** |
| 微信app支付  (参照微信app支付接口, 根据返回结果,调起app付款界面) | **{**  "package"**:** "Sign=WXPay"**,**  "appid"**:** "wxd4934d0dab14d276"**,**  "sign"**:** "73EEDC733F95AE6BBE0BF3702C7D58E8"**,**  "partnerid"**:** "1509195691"**,**  "prepayid"**:** "wx281536402386406d4cb7b2941357724800"**,**  "noncestr"**:** "1559029001243"**,**  "timestamp"**:** "1559029001"  **}** |
| 微信小程序支付  (参照微信小程序支付接口, 根据返回结果,调起小程序付款界面)  支付参数需要提供openId | **{**  "timeStamp"**:** "1559029630"**,**  "package"**:** "prepay\_id=wx281547094221749d540390a39747283500"**,**  "paySign"**:** "9146AFCFF672C1634015EAD0432CAFED"**,**  "appId"**:** "wx42e6b214e6cdaed3"**,**  "signType"**:** "MD5"**,**  "nonceStr"**:** "1559029630405"  **}** |
| 支付宝扫码支付 | **{**  "qrCodeUrl"**:**  "https://qr.alipay.com/bax04754v9m8ryawords404f"**,**  "payTradeNo"**:** "20190528155750156219628508326023"  **}** |
| 支付宝h5支付  (参照支付宝h5支付接口, 根据返回结果,调起h5付款界面) | **{**  "payTradeNo"**:** "20190528160001088875799146277193"**,**  "mwebUrl"**:** "<form name=\"punchout\_form\"..."  **}** |
| 支付宝app支付  (参照支付宝app支付接口, 根据返回结果,调起app付款界面) | **{**  "form"**:** "alipay\_sdk\u003dalipay-sdk-..."**,**  "payTradeNo"**:** "20190528160250548597112802067623"  **}** |

## 4.2 微信/支付宝支付结果回调

前面调用支付系统时,请求参数中的notifyUrl表示当支付系统收到外部支付通知(支付宝,微信)后,会将相应的支付结果通过此url转发给应用系统.应用系统接收回调参数后,通过resultCode是否等于"SUCCESS"判断支付是否成功

如果支付成功,则调用

response.getWriter().print("SUCCESS");

通知支付系统已经回调处理成功,否则

response.getWriter().print("FAIL");

通知支付系统回调处理失败.

### 4.2.1回调参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **变量名** | **示例值** | **描述** |
| 支付结果 | resultCode | SUCCESS/FAIL | SUCCESS:支付成功  FAIL:支付失败 |
| 支付结果描述 | resultMsg | OK/ FAIL |  |
| 应用系统交易号 | intenalTradeNo | 201502014124 | 应用系统自己生成的交易号 |
| 支付系统交易号 | payTradeNo | 112015020141249963 |  |
| 交易描述 | tradeDesc | 积分充值 | 业务场景描述 |
| 支付金额 | totalFee | 1 | 以分为单位 |
| 收款应用id | appId | 2016092301955254 |  |
| 交易来源 | passbackParams | 1 | 回传参数. |

### 4.2.2示例代码

|  |
| --- |
| @RequestMapping**(**value **=** "/test/notify"**)**  **public** **void** test**(**HttpServletRequest request**,** HttpServletResponse response**)** **{**  **try** **{**  log**.**error**(**"返回参数:"**);**  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");  log**.**error**(**name**+**" : "**+**valueStr**);**  **}**  response**.**getWriter**().**print**(**"SUCCESS"**);**  **}** **catch** **(**IOException e**)** **{**  // TODO Auto-generated catch block  e**.**printStackTrace**();**  **}**    **}** |

## 4.3 微信/支付宝退款:

### 4.3.1示例代码

|  |
| --- |
| **public** **void** testRefund**()** **{**  RefundParamDto refundParamDto **=** **new** RefundParamDto**();**  //付款时,支付系统返回的支付交易号  refundParamDto**.**setOriginPayTradeNo**(**"20190520163536993810774284495696"**);**  //本系统自己生成的退款交易号,可以通过此交易号去支付服务查询相关退款交易  refundParamDto**.**setInternalRefundNo**(**"654321"**);**  //原支付金额,以分为单位  refundParamDto**.**setTotalFee**(**1L**);**  //退款金额,以分为单位  refundParamDto**.**setRefundFee**(**1L**);**  //原支付方式  refundParamDto**.**setPayMethod**(**PayMethodContants.WX\_SCANCODE\_PAY**);**  //退款交易描述  refundParamDto**.**setRefundDesc**(**"取消订单退款"**);**  //ip  refundParamDto**.**setIp**(**"127.0.0.1"**);**  //退款结果回调通知地址  refundParamDto**.**setNotifyUrl**(**"http://127.0.0.1:8089/v1/refund/callback/test/notify"**);**  //客户端id:服务id或者应用id,需要在支付服务注册  refundParamDto**.**setClientId**(**"001"**);**  //第三方收款应用id  refundParamDto**.**setAppId**(**"wxd4934d0dab14d276"**);**  //需要保证上述参数设置完成后再生成签名  refundParamDto**.**setSign**(**refundParamDto**.**createSign**(**"123456"**));**  //返回退款结果为json格式  String result **=** payServiceClient**.**doRefund**(**refundParamDto**);**  //本应用系统做业务处理  System**.**out**.**println**(**result**);**  **}** |

### 4.3.2微信退款返回结果

|  |
| --- |
| **{**  //支付时的应用系统内部交易号  "originInternalTradeNo"**:** "123456"**,**  //支付时的支付系统交易号  "originPayTradeNo"**:** "20190520163536993810774284495696"**,**  //退款时的退款交易号  "internalRefundNo"**:** "654321"  //支付系统退款交易号  "payRefundNo"**:** "RF20190520163652336200089139821650"**,**  //退款金额  "refundFee"**:** "1"  **}** |

### 4.3.3支付宝退款返回结果

比微信退款结果多返回一个successTime,并且实时返回退款结果,没有回调.

|  |
| --- |
| **{**  //支付时的应用系统内部交易号  "originInternalTradeNo"**:** "123456"**,**  //支付时的支付系统交易号  "originPayTradeNo"**:** "20190520163536993810774284495696"**,**  //退款时的退款交易号  "internalRefundNo"**:** "654321"  //支付系统退款交易号  "payRefundNo"**:** "RF20190520163652336200089139821650"**,**  //退款金额  "refundFee"**:** "1"  //退款时间  " successTime"**:** "2019-01-01 11:12:00"  **}** |

## 4.4 微信退款回调

前面调用退款方法时,请求参数中的notifyUrl表示当支付系统收到外部退款回调通知(微信)后,会将相应的退款结果通过此url转发给应用系统.应用系统接收回调参数后,通过resultCode是否等于"SUCCESS"且refundStatus=" SUCCESS "判断是否退款成功

如果退款成功,则调用

response.getWriter().print("SUCCESS");

通知支付系统已经处理成功,否则

response.getWriter().print("FAIL");

通知支付系统处理失败.

注意:支付宝实时返回退款结果,不需要回调

### 4.4.1回调参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **变量名** | **示例值** | **描述** |
| 退款调用结果 | resultCode | SUCCESS/FAIL | SUCCESS:调用成功  FAIL:调用失败 |
| 退款调用描述 | resultMsg | OK/ FAIL |  |
| 应用系统交易号 | intenalTradeNo | 201502014124 |  |
| 应用系统退款号 | internalRefundNo | 112015020141249963 |  |
| 支付系统交易号 | payTradeNo | 1235 |  |
| 支付系统退款号 | payRefundNo | 112122 |  |
| 退款金额 | refundFee | 1 | 以分为单位 |
| 退款描述 | refundDesc | 取消订单退款 | 业务场景描述 |
| 退款状态 | refundStatus |  | SUCCESS-退款成功 CHANGE-退款异常 REFUNDCLOSE—退款关闭 |
| 退款成功时间 | successTime |  |  |
| 收款应用id | appId |  |  |

### 4.4.2示例代码

|  |
| --- |
| @RequestMapping**(**value **=** "/test/notify"**)**  **public** **void** test**(**HttpServletRequest request**,** HttpServletResponse response**)** **{**  **try** **{**  log**.**error**(**"返回参数:"**);**  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");  log**.**error**(**name**+**" : "**+**valueStr**);**  **}**  // return ResponseEnvelope.ok();  response**.**getWriter**().**print**(**"SUCCESS"**);**  **}** **catch** **(**IOException e**)** **{**  // TODO Auto-generated catch block  e**.**printStackTrace**();**  **}**    **}** |

## 4.5 微信小程序企业付款

### 4.5.1示例代码

|  |
| --- |
| @PostMapping**(**"/testTransfers"**)**  **public** **void** testTransfers**()** **{**  TransfersParamDto transfersParamDto **=** **new** TransfersParamDto**();**  //应用系统自己生成的付款交易号,可以通过此交易号去支付服务查询相关付款交易  transfersParamDto**.**setIntenalTradeNo**(**"112233"**);**  //小程序用户openId  transfersParamDto**.**setOpenId**(**"o-D9N5SS\_eyau3b8Q1qZMo\_FWxB"**);**  //付款金额,最少0.3元  transfersParamDto**.**setAmount**(**30L**);**  //交易描述  transfersParamDto**.**setTradeDesc**(**"tradeDesc"**);**  //微信小程序支付  transfersParamDto**.**setPayMethod**(**PayMethodContants.WX\_MINI\_PAY**);**  //客户端ip  transfersParamDto**.**setIp**(**"127.0.0.1"**);**  //客户端id:服务id或者应用id,需要在支付服务注册  transfersParamDto**.**setClientId**(**"001"**);**  //第三方收付款应用id  transfersParamDto**.**setAppId**(**"wx42e6b214e6cdaed3"**);**  //需要保证上述参数设置完成后再生成签名  transfersParamDto**.**setSign**(**transfersParamDto**.**createSign**(**"123456"**));**  //返回退款结果为json格式  String result **=** payServiceClient**.**doTransfer**(**transfersParamDto**);**  //本应用系统做业务处理  System**.**out**.**println**(**result**);**  **}** |

### 4.5.2返回结果

|  |
| --- |
| **{**  //应用系统交易号  "internalTradeNo"**:** "112233"**,**  //支付系统交易号  "payTradeNo"**:** "20190521105054288797453935267333"**,**  //付款时间  "paymentTime"**:** "2019-05-21 10:51:05"  **}** |