# wxss-sdk

| Date | Version | Owner | Comments |
| --- | --- | --- | --- |
| 2019-05-21 | 0.1 | jingyang | 新建初始化 |
| 2019-05-24 | 0.2 | jingyang | 增加项目信息 |
| 2019-05-28 | 0.3 | jingyang | 修正标题 |



# 提交给业务方的内容

1. utils目录下的七个js文件


# 开发环境

## Git仓库

1. http://git.eju-inc.com/ejupay/cashier-wxss.git，小程序SDK代码，分支develop

## 环境搭建

1. 将git仓库demo克隆到本地后，在微信开发者工具中打开。
2. 更新项目的appid
3. 首次导入，会出现一个color.wxss文件的编译错误。两种解决方式：
    - 方式一：
        1. 重命名style/base/variable/color.wxss文件为/style/base/variable/color.wxss.bak
        2. 在style/style.wxss文件中，注释掉import color.wxss的行：*/@import "./base/variable/color.wxss";/*
    - 方式二：
        1. 编译打包小程序，在项目根目录执行命令：gulp build:ejupay
        2. 为了避免编译报错问题，采用编译后的目录/dist为项目目录导入到小程序开发工具
4. 编译加载js过程中，会出现初始化报错问题。在ejupay/index/index.js文件中，注释掉变量初始化的代码块、后续测试直接填充测试数据。

```
// const {partner, openId, token, DESKEY, body, sign} = servers.getStorageSync('postBody');
// this.setData({partner, openId, token, DESKEY, body, sign})


// servers.errorTip('123')
// servers.errorTip('我们')
```

## 代码结构

- 以/dist目录为例：
    - ejupay：主要存放支付小程序测试相关的页面文件，通常每个子文件夹对应一个页面
    - modules：存放第三方或自定义js模块文件。本项目中用到了加密相关的文件
    - style：存放第三方公共样式wxss文件。
    - utils：存放工具类文件，这个文件夹下的内容就是提供给业务方的SDK。
    - app.js：用来定义小程序的全局数据和函数，控制、监听小程序的全生命周期。目前仅获取了用户信息。
    - app.json：小程序的配置文件。目前配置了页面路径、窗口表现、网络超时。
    - app.wxss：小程序公共样式。
    - project.config.json：小程序开发时工具的配置信息。

# 业务交互逻辑

1. 查看app.json，

```
{
  "pages":[
    "ejupay/index/index",
    "ejupay/logs/logs",
    "ejupay/cashier/cashier",
    "ejupay/balancePay/balancePay"
  ],
```

2. 可见ejupay/index/index.wxml为首页 找到ejupay/index/index.js，找到onLoad: function ()。 可以看到，初始化index页面时，小程序发起了一次对Server端的调用，获取相关的测试数据。

```
acquirerWithBrandPay({
      requestTime: new Date().getTime(),
      ticketNo: '',
      partner: '', //合作伙伴Id（一级商户接入号）
      sign: '',
      token: '',
      version: 'v1.1',
      body: body,
      payurl: 'https://ejupay.17shihui.com/gateway-outrpc/acquirer/interact'
})
```

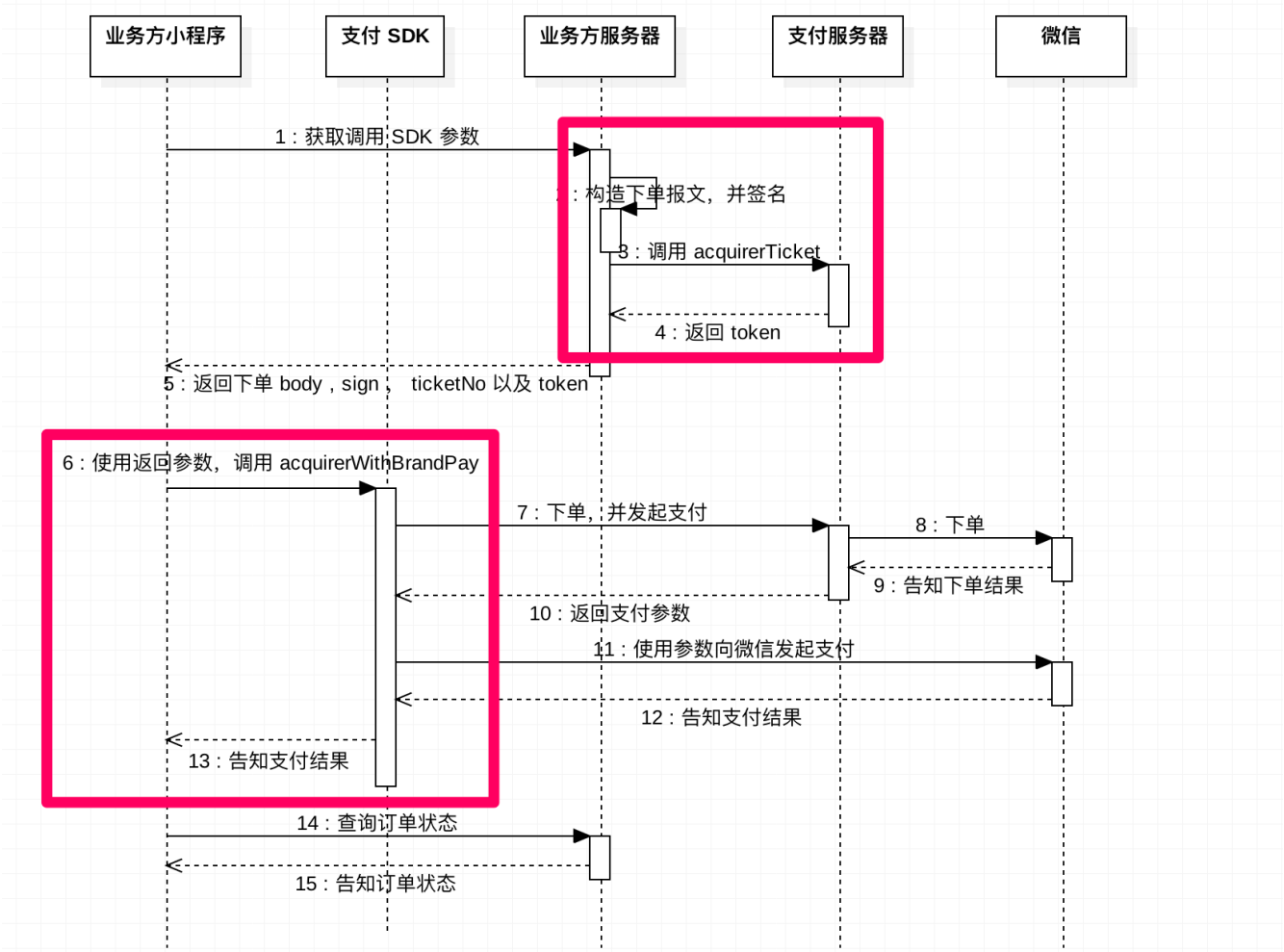
注意：参数的设置，默认的payurl是生产环境的，容易让对接开发人员犯错！

3. 该操作完成了调用支付后端服务接口 **【时序图中第7步】**/gateway-outrpc/acquirer/interact，创建支付订单再调用wx.requestPayment()函数，完成支付 **【时序图中第11步】**。

```
wx.requestPayment({
    'appId': appid, //公众号名称，由商户传入
    'timeStamp': timestamp, //时间戳，自1970年以来的秒数
    'nonceStr': noncestr, //随机串
    'package': pkg,
    'signType': signType, //微信签名方式:
    'paySign': paySign, //微信签名
```

# 时序图

注意：第6步才开始调用SDK的接口！前面的步骤是业务方小程序通过业务方服务器，从支付服务端获取下单需要的token（支付服务端用于区分不同的业务方），sign等信息。也就是利用支付服务端帮助构建和拼装调用微信wx.requestPayment(Object)接口需要的参数。




# 测试指南

## 获取测试参数

以拼住为例：

### 1、查询业务方partnerId （已创建partner，~一级商户）

```
select MERCHANT_NAME, MERCHANT_ID from cash.tb_contract_tool t1, cash.tb_contract t2 where t1.tool_id = 36 and t1.contract_id = t2.id;
```

返回：

```
1    库拍    100001
2    开放平台        100029
3    新实惠  100044
4    萌宠    100024
5    宝库汇  100035
6    一铺    100022
7    拼猪    100030
8    时钟教室NEW  100032
9    酒店民宿        100049
10   实惠    100014
```

## 2、创建商户merchant (TBD, 二级商户：为用户提供服务/产品，收取资金的商家)**

先采用已有数据测试:

1. 从mb.tb_member_type表中获取member_type，一般是非default eg：

```
select id, name from mb.tb_member_type where area_id= 100030;
```

返回：

```
1   50  拼猪房客普通会员
2   51  拼猪房东普通会员
3   52  拼猪房东机构商户
4   53  拼猪经纪人
```

2. 查找现有的二级商户merchant eg：

```
select id, name, status from mb.tb_member where area_id = 100030 and membertype_id=51 and
rownum < 100;   //51  拼猪房东普通会员
```

返回：

```
1   1017854 11  1
2   1017972 Landlord28515000590380  1
3   1018086 Landlord48615000590409  1
4   1018401 Landlord69315000590900  1
5   1018402 Landlord69518516232951  1
6   1018403 Landlord69715000590329  1
7   1018591 Landlord73915072443787  1
8   1018638 Landlord76715071443721  1
9   1017919 钱迷  1
10  1017942 秦海贤 1
11  1017943 仇才人 1
....
```

作为示例，取merchant.ID = 1017854 拼猪的商户——钱迷

## 3、创建用户member （也可采用已有用户）

进入《申请网关交互》页面：URL

1. 请选择一种算法：均采用默认值，测试环境的key不要动，默认的。【用于业务方服务端与平台的交互】
2. 请填下header部分信息： 合伙人：partnerId --》100030 请求体信息
    - requestTime= 打开一个新的标签页，输入《申请网关交互》页面URL，刷新后响应结果里会返回当前系统时间。
    - loginModeId=47 （参见mb.tb_login_mode，包含UID的指的是业务方的用户。） // select * from mb.tb_login_mode where rownum < 100 and area_id=100030; area_id即partnerId
    - memberTypeId=50 （参见mb.tb_member_type） //select * from mb.tb_member_type where area_id=100030 area_id即partnerId，这里选择了 50 拼猪房客普通会员
    - loginFlag 登录用的账号，一般用手机号，邮箱

- memberName
- mp 手机号

3. 点击《签名》按钮，生成签名信息
4. 点击《提交》按钮，提交请求到服务网关gateway，如正常，则返回生成的memberId eg：

```
{
    "requestTime": 1547547271198,
    "service": "createMember",
    "loginModeId": "47",
    "memberTypeId": 50,
    "loginFlag": "15001709702",
    "memberName": "yangjing",
    "mp": "15001709702"
}
```

响应结果示例：

```
{"memberId":1018864,"operatorId":18876,"responseCode":"000"}
```

## 4、获取workingKey

在《申请网关交互》页面，【模拟业务方后台服务与支付平台的交互，详见gateway Api的getWorkingKey接口 】

1. 请选择一种算法：均采用默认值，测试环境的key不要动，默认的。【用于业务方服务端与平台的交互】
2. 请填下header部分信息： 合伙人：partnerId --》100030 请求体信息
    - requestTime，通过打开并刷新新的《申请网关交互》页面，在响应结果中获取
    - expiryDate，requestTime基础上加一些毫秒数。 //多加一点，测试用
    - service=getWorkingKey
    - memberId=用户ID，例如：1018864
    - terminalType=member的terminalType
3. 点击《签名》按钮，生成签名信息
4. 点击《提交》按钮，提交请求到服务网关gateway，如正常，则返回生成的memberId 如正常，则返回生成的memberId
    - 输出：partnerId，cipherKey工作密钥（DES加密、解密），signatureKey工作密钥（MD5签名、验签），token工作密钥（当终端类型不属于SDK时，有值） eg：

```
{
    "requestTime": 1547547271198,
    "expiryDate": 1547647271198,
    "service": "getWorkingKey",
    "memberId": 1018864,
    "terminalType": "wechat"
}
```

请求的terminalType如果是SDK，则响应结果中无token，此处测试小程序，采用wechat。 响应结果示例：

{"cipherKey":"VGns79eHrsrFZ2gBCfIWUPiKE5qwStd9/OjC8AwitOMqh3php+KLBSo5BdvILNV8zGgGVtVgZ55zJLwM1PsjOCzTChBjVuLByggEvTbQszVG4f1uus7WyS6eFqKMQzPpzsrvj12BVOSFZcolpjdsCyUIJibkwUTKkD+ZPyifJLA=","partnerId":1018864,"responseCode":"000","responseMsg":"操作成功","signatureKey":"yGZX5yfu/1o8V46G7CX0tl7zXFwm6gZQVCAi15jCF6SpnAw4A87t4uVA2s+Icnyq47IGw8R5CD68i4KwGr+3r1TYT1K+fqRLrxYMXpvtgTtHDc0froaWWLMDdozUEpq/v1vawZIu2ox6A0ccnziO5nIRw0/B7bZyoy/mrwiOUUY=","token":"fd0408d51f6945c5103b27c5d83ad3d5"}

## 5、创建acquirerWithBrandPay请求的签名sign

在《申请网关交互》页面，【模拟业务方后台服务的操作】

1. 选择算法：默认的。
2. 合伙人=partnerId 100030
3. 请求体信息， 【以下单品牌支付二合一为例，详细请参考 doc/adapter/cash/acquirerWithBrandPay.html】
   eg:

{"amount":0.01,"describe":"拼住押金支付","merchantId":100030,"notifyUrl":"http://10.122.130.7:50020/lessee/pay/notifyCallBack","operationId":140101,"orderNO":"20181024210020664810","payRequestParams":{"openId":"ofPoM5PtH--NIZRbS6RrJOakku2I"},"payToolCode":"weixinMicroProPay","payee":1018390,"requestTime":1547535610599,"returnUrl":"../payment-pindan/app","service":"acquirerWithBrandPay","terminalType":"microPro","traceNO":"1540386020664543518OO"}

4. 点击《签名》按钮，生成签名信息

hJmoU7ygqDZPHXVzOYHI0Ln+FmSUvDtJ/GKbHv6vYIZFSbStDmsPAyf7Dzsjbl7zcvKcV/r2cc2jr2ag+T6OD0EdLO4z8AN+JG1aWhtp5VKg0DvFnpSb6cZYuznNg7YpLrEWFkBEpZB4PDMsQGjuQp0orEybJqtXRp97TdtG1u8=

注：此处的openId必须与appId匹配，参考微信获取用户openId 此处分两步实现： eg:

```
1、获取code （直接调用微信接口未通过，暂采用微信自带接口）
  wx.login({
      success(res) {
        if (res.code) {
          console.log(res.code)
        } else {
          console.log('登录失败! ' + res.errMsg)
        }
      }
    })
2、用code获取openId
https://api.weixin.qq.com/sns/jscode2session?appid=wx6161eb9b3b977496&secret=4ce84246a2ea7b6d011e6e98bd22e9fb&js_code=033m2hjZ0wcW4W198EgZ0oDhjZ0m2hjP&grant_type=authorization_code
```

## 6、小程序调用函数acquirerWithBrandPay()

在index.js中 1.输入 + ticketNo=订单编号 参见cash的acquirerWithBrandPay服务接口 此处可为空 + partner=partnerId + sign=请求体的签名sign + token=获取workingKey时，返回的token工作密钥 + body=请求体（步骤：创建acquirerWithBrandPay请求的签名sign，在网关交互系统后台，用于创建签名的请求体） + payurl=https://inte.yjyyun.com/gateway-outrpc/acquirer/interact （根据生产、预发、集成、测试、开发环境调整域名）
eg:

```
  acquirerWithBrandPay({
      requestTime: new Date().getTime(),
      ticketNo: '',
      partner: '100030', //合作伙伴Id（一级商户接入号）
      sign:
'hJmoU7ygqDZPHXVzOYHIOLn+FmSUvDtJ/GKbHv6vYIZFSbStDmsPAyf7Dzsjbl7zcvKcV/r2cc2jr2ag+T6OD0EdLO4
z8AN+JG1aWhtp5VKgODvFnpSb6cZYuznNg7YpLrEWFkBEpZB4PDMsQGjuQpOorEybJqtXRp97TdtG1u8=',
      token: '0680f57a749ee9f139120f9cd2c7c65e',
      version: 'v1.1',
      body: '{"amount":0.01,"describe":"拼住押金支
付","merchantId":100030,"notifyUrl":"http://10.122.130.7:50020/lessee/pay/notifyCallBack","op
erationId":140101,"orderNO":"20181024210020664810","payRequestParams":{"openId":"ofPoM5PtH--
NIZRbS6RrJOakku2I"},"payToolCode":"weixinMicroProPay","payee":1018390,"requestTime":15475356
10599,"returnUrl":"../payment-
pindan/app","service":"acquirerWithBrandPay","terminalType":"microPro","traceNO":"1540386020
66454351800"}',
      payurl: 'https://inte.yjyyun.com/gateway-outrpc/acquirer/interact'
    })
```

2. 输出
    - 参看fail() service-operation.js
    - responseCode=111 （InProgress）
    - 调用wechatPay()函数

### 7、小程序调用函数wechatPay()函数

```
 + wechatParams: res，转换为wx.requestPayment()的参数要求。【注：目前汇付返回的结果应该包括除了
success, fail, complete三个回调函数外，所有的参数appId, timeStamp, nonceStr, package, signType,
paySign
 + See 函数receiveSyncResult()    【service-operation.js】
 + 调用wx.requestPayment()函数，拉起微信支付界面。
```

## 本地测试

1. 找到测试入口文件ejupay/index/index.js，作如下修改： 加入登录方法、注释掉变量初始化的代码块、直接填充测试数据（参考上面的获取方式）

```
    wx.login({
      success(res) {
        if (res.code) {
          console.log("jscode is ---> " + res.code)
        } else {
          console.log('登录失败! ' + res.errMsg)
        }
      }
    })

  //  const {partner, openId, token, DESKEY, body, sign} = servers.getStorageSync('postBody');
  //  this.setData({partner, openId, token, DESKEY, body, sign})

    servers.errorTip('123')
    servers.errorTip('我们')

    acquirerWithBrandPay({
      requestTime: new Date().getTime(),
      ticketNo: 'EJUPAYTEST_20190515210020664820',
      partner: '999999', //合作伙伴Id（一级商户接入号）
      sign: 'r4dueH/E8iJeKtg4Nv+FqQn8v+bPozdtU7Iw98ZvTmRZhr+vW5OQqxYwrAJxm2UlsU2XQ766DxFh4V8ewlQS8IpD2M2BThWFqRvzPzMWot5V4OiIv4BWTcR3emtYe2SM/IDCW4LWHT0fihXC+J+1uO/ooK9xz2I83+YoX43CPRc=',
      token: 'f112bc2962db5a66cc1039efe6454cf5',
      version: 'v1.1',
      body: '{"amount":0.01,"describe":"支付测试","merchantId":999999,"notifyUrl":"https://lark.yjyyun.com/eju/pay/notifyCallBack","operationId":1990,"orderNO":"EJUPAYTEST_20190515210020664820","payRequestParams":{"openId":"onEmQ4hwVg5nZOG2SIXNFRxcoirw"},"payToolCode":"weixinMicroProPay","requestTime":1557906172613,"service":"acquirerWithBrandPay","terminalType":"microPro","traceNO":"37d867f3-3a0b-472e-a9cc-563b4c616333"}',
      payurl: 'https://inte.yjyyun.com/gateway-outrpc/acquirer/interact'
    })
  }
```

1、初始化时，调用微信登录方法，获取jsCode，之后通过jsCode获取openId

2、不通过页面输入测试情况下（目前不可用），可直接在初始化时，填充测试数据，调用支付接口

2. 准备好测试数据，重新编译小程序（一般保存代码后默认自动编译），接口调用成功会出现扫码弹出框，如下：


3. 扫码支付成功，表示测试完成。