API接入技术文档

科技有限公司

2017.7

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V1.0 | jinpeng |  |  | 初版 |

目录

[1 总体描述 5](#_Toc480533299)

[1.1 引言 5](#_Toc480533300)

[1.2 功能简述 5](#_Toc480533301)

[2 综述 5](#_Toc480533302)

[2.1 技术对接流程 5](#_Toc480533303)

[2.2 平台通信协议 6](#_Toc480533304)

[2.3 平台通信加密方法 6](#_Toc480533305)

[3 接口功能 7](#_Toc480533306)

# 总体描述

## 引言

* 本文档介绍旖美分期平台中对外接口的概要设计。
* 本文档的预期读者是：
* 双方技术管理人员；
* 详细设计人员、代码编写人员以及测试人员；
* 参与本项目开发进程各阶段验证、确认以及负责为最后项目验收、鉴定提供相应报告的有关人员。

## 功能简述

* 1. 旖美预授信
  2. 技术对接流程
* 接入方向平台方联调测试。
* 待双方测试并验收后，提交平台上线申请。

## 平台通信协议

* 平台使用HTTP/HTTPS作为传输协议；
* 平台使用UTF-8对报文进行编码；

## 平台通信加密方法

平台通过对接入方发送的数据进行加密（可选），保证其请求不被伪造。

旖美将为接入方发放加密密钥，它们是：

1、用户密钥（securedKey）：用于在通讯时加密。

接入方必须对密钥严格保密。平台将定期更换密钥以确保系统安全。平台使用RSA算法对参数中的关键信息加密，并对加密结果进行BASE64编码。

# 接口功能

## 跳转旖美进件页面

#### 调用入口地址（GET请求）

测试地址：http://example.juxiangfen.com/#/kelan/login

生产地址：https://beauty.juxiangfen.com/#/kelan/login

#### 请求字段定义

请求参数如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| outUserId | string | \*第三方用户id | 客户系统的唯一用户ID |
| outOrderId |  | \*第三方订单id | 客户系统的唯一订单ID  (type=0002时不用传值) |
| platform | String | \*来源平台 | 固定值：heyi |
| randomNum | String | \*随机数（一次有效） | 随机数=一个当前的毫秒数+4位随机整数 |
| type | String | \*类型 | 根据type类型跳转不同页面:  0001=进件申请(跳转标准进件流程页面)  0002=还款列表(跳转还款列表页面,页面展示该用户下所有订单信息) |
| backUrl | String | 返回的url | 从旖美跳转回商户系统的页面 |
| notifyUrl | String | 消息通知地址 | type=0002时不用传值 |
| hosCode | String | 医院编码 | type=0002时不用传值 |
| latitude | String | 纬度 | type=0002时不用传值 |
| longitude | String | 经度 | type=0002时不用传值 |
| sign | string | \*业务参数签名 | 业务参数拼接后过SHA1WithRSA算法（旖美私钥）的加密后的字符串 |

#### 加密步骤如下：（旖美私钥）

1. 业务参数名称经过**字母升序**排列，拼接成待签名的字符串（参数值为空不参与字符串拼接），示例：
2. 拼接后的json字符串经过rsa加密后

加密后的字符串再进行urlEncoder编码

#### 示例：

**待加密的字符串：(只需要下面四个参数参与签名)**

**outUserId=value&**

**platform=value&**

**randomNum=value&**

**type=value**

#### 页面请求完整url示例：

http://example.juxiangfen.com/#/kelan/login

?amount=URLEncoder.encode(value, "UTF-8")

&backUrl=URLEncoder.encode(value, "UTF-8")

&hosCode=URLEncoder.encode(value, "UTF-8")

&hosName=URLEncoder.encode(value, "UTF-8")

&latitude=URLEncoder.encode(value, "UTF-8")

&longitude=URLEncoder.encode(value, "UTF-8")

&mobile=URLEncoder.encode(value, "UTF-8")

&notifyUrl=URLEncoder.encode(value, "UTF-8")

&outOrderId=URLEncoder.encode(value, "UTF-8")

&outUserId=URLEncoder.encode(value, "UTF-8")

&period=URLEncoder.encode(value, "UTF-8")

&platform=URLEncoder.encode(value, "UTF-8")

&projName=URLEncoder.encode(value, "UTF-8")

&platform=URLEncoder.encode(value, "UTF-8")

&randomNum=URLEncoder.encode(value, "UTF-8")

&type=URLEncoder.encode(value, "UTF-8")

&sign=URLEncoder.encode(value, "UTF-8")

#### 返回报文说明

无

## 订单审核通知接口

#### 调用入口地址(POST-JSON)

服务地址：**跳转旖美进件时的notifyUrl**

注意：**接口我们只调用3次，3次都失败，我们就会放弃通知**

#### 请求字段定义

业务参数如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| outUserId | string | \*第三方用户id | 客户系统的唯一用户ID |
| outOrderId | String | \*第三方订单ID | 客户系统的唯一订单ID |
| notifyType | String | \*通知类型 | **A0001=审核通知**  A0003=支付通知  A0004=逾期通知  A0005=提醒还款通知 |
| stauts | String | 处理状态 | 0000=审核通过  1000=审核拒绝 |
| message | String | 描述 |  |
| sign | string | \*业务参数签名 | 业务参数拼接后过SHA1WithRSA算法（贝贝私钥）的加密后的字符串 |

#### 加密步骤如下：(私钥)

1. 业务参数名称经过**字母升序**排列，拼接成待签名的字符串
2. outUserId、outOrderId、notifyType三个字段参与签名
3. 拼接后的字符串经过rsa加密

#### 返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| code | string | \*接口调用返回标识，JSON格式 | 200=接口调用成功 |

## 2.3 提醒还款通知

#### 调用入口地址(POST-JSON)

服务地址：**跳转旖美进件时的notifyUrl**

注意：**接口我们只调用3次，3次都失败，我们就会放弃通知**

#### 请求字段定义

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| outUserId | string | \*第三方用户id | 客户系统的唯一用户ID |
| outOrderId | String | \*第三方订单ID | 客户系统的唯一订单ID |
| notifyType | String | \*通知类型 | A0001=审核通知  **A0003=支付通知**  A0004=逾期通知  A0005=提醒还款通知 |
| satus | String | \*还款提醒 | 0000=还款前一天的提醒 |
| mssage | String | 描述 | 描述 |
| data | JSON | \*信息 |  |
| period | String | \*还款期数 |  |
| amount | String | \*还款金额 |  |
| sign | string | \*业务参数签名 | 业务参数拼接后过SHA1WithRSA算法（贝贝私钥）的加密后的字符串 |

#### 加密步骤如下：（私钥）

1. 业务参数名称经过**字母升序**排列，拼接成待签名的字符串
2. outUserId、outOrderId、notifyType两个字段参与签名
3. 拼接后的字符串经过rsa加密

#### 传参方式：JSON格式

#### 返回报文说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| code | string | \*接口调用返回标识，JSON格式 | 200=通知成功 |

## 2.4 用户支付通知

#### 调用入口地址(POST-JSON)

服务地址：**跳转旖美进件时的notifyUrl**

注意：**接口我们只调用3次，3次都失败，我们就会放弃通知**

#### 请求字段定义

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| outUserId | string | \*第三方用户id | 客户系统的唯一用户ID |
| outOrderId | String | \*第三方订单ID | 客户系统的唯一订单ID |
| notifyType | String | \*通知类型 | A0001=审核通知  **A0003=支付通知**  A0004=逾期通知  A0005=提醒还款通知 |
| satus | String | \*还款状态 | 0000=支付成功  1000=支付失败 |
| mssage | String | 描述 | 代扣描述 |
| data | JSON | \*信息 |  |
| period | String | \*还款期数 |  |
| amount | String | \*还款金额 |  |
| sign | string | \*业务参数签名 | 业务参数拼接后过SHA1WithRSA算法（贝贝私钥）的加密后的字符串 |

#### 加密步骤如下：（私钥）

1. 业务参数名称经过**字母升序**排列，拼接成待签名的字符串
2. outUserId、outOrderId、notifyType两个字段参与签名
3. 拼接后的字符串经过rsa加密

#### 传参方式：JSON格式

#### 返回报文说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| code | string | \*接口调用返回标识，JSON格式 | 200=通知成功 |

## 2.5 逾期推送通知接口

#### 调用入口地址(POST-JSON)

服务地址：**跳转旖美进件时的notifyUrl**

注意：**接口我们只调用3次，3次都失败，我们就会放弃通知**

**请求字段定义**

业务参数如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| outUserId | string | \*第三方用户id | 客户系统的唯一用户ID |
| outOrderId | String | \*第三方订单ID | 客户系统的唯一订单ID |
| notifyType | String | \*通知类型 | A0001=审核通知  A0003=支付通知  **A0004=逾期通知**  A0005=提醒还款通知 |
| data | JSON对象 | \*逾期的借款信息 |  |
| status | String | 还款状态  0:未还款  1;已支付  2:扣款失败  3:正在扣款  -1:取消 |  |
| seq | String | 第几期 |  |
| totalAmount | String | 本期应还金额 |  |
| principal | String | 本金 |  |
| overdue | String | 逾期金额 |  |
| interest | String | 利息 |  |
| planDate | String | 计划还款时间 |  |
| actualDate | String | 实际还款时间 |  |
| sign | string | \*业务参数签名 | 业务参数拼接后过SHA1WithRSA算法（贝贝私钥）的加密后的字符串 |

#### 加密步骤如下：(私钥)

1. 业务参数名称经过**字母升序**排列，拼接成待签名的字符串
2. outUserId、outOrderId、notifyType三个字段参与签名
3. 拼接后的字符串经过rsa加密

#### 返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| code | string | \*接口调用返回标识，JSON格式 | 200=接口调用成功 |

## 2.6 补件通知接口

#### 调用入口地址(POST-JSON)

服务地址：**跳转旖美进件时的notifyUrl**

注意：**接口我们只调用3次，3次都失败，我们就会放弃通知**

#### 请求字段定义

业务参数如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| outUserId | string | \*第三方用户id | 客户系统的唯一用户ID |
| outOrderId | String | \*第三方订单ID | 客户系统的唯一订单ID |
| notifyType | String | \*通知类型 | A0001=审核通知  A0003=支付通知  A0004=逾期通知  A0005=提醒还款通知  A0006=补件通知 |
| data | JSON对象 | \*逾期的借款信息 |  |
| supplementList | List<JSON> | 需要重新上传的附件集合(notifyType=A0006,需要补件时返回) |  |
| type | String | 图片类型 | 标准化结构字段,参见附录:附件类型说明 |
| info | String | 附件不合格的原因 | 如:不清晰 |
| sign | string | \*业务参数签名 | 业务参数拼接后过SHA1WithRSA算法（贝贝私钥）的加密后的字符串 |

报文格式:

{

"outuserId": "1000001",

"outOrderId": "1000",

"notifyType": "A0006",

"data": {

"supplementList":   [{

"type": "1",

"type": "图片不清晰"

}, {

"type": "2",

"type": "图片不清晰"

}, {

"type": "3",

"type": "图片不清晰"

} ]

}

}

加密步骤如下：(私钥)

1. 业务参数名称经过**字母升序**排列，拼接成待签名的字符串
2. outUserId、outOrderId、notifyType三个字段参与签名
3. 拼接后的字符串经过rsa加密

**返回参数说明：**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型（字段长度） | 说明（\*为必填项） | 示例 |
| code | string | \*接口调用返回标识，JSON格式 | 200=接口调用成功 |

## 附件类型说明

|  |  |
| --- | --- |
| type | 照片类型 |
| 1 | 身份证正面照 |
| 2 | 身份证背面照 |
| 3 | 手持身份证照 |
| 4 | 医疗确认照 |
| 5 | 房产证原件照片 |
| 6 | 行驶证原件照片 |
| 7 | 工作证原件照 |
| 8 | 名片原件照 |
| 9 | 代办人照片/申请人与实际消费者在商户合影 |
| 10 | 首付凭证照 |
| 11 | 课程购买合同 |

## 测试环境加密密钥

|  |  |
| --- | --- |
| 名称 | 私钥值 |
| 旖美RSA私钥 |  |

## 生产环境加密密钥

|  |  |
| --- | --- |
| 名称 | 私钥值 |
| 旖美RSA私钥 |  |