**信用借还订单创建/完结异步通知**

**版本号：1.0**

目录

[1. 文档说明 3](#_Toc469581937)

[1.1通知功能描述 3](#_Toc469581938)

[2. 请求参数说明 3](#_Toc469581939)

[2.1异步通知参数 3](#_Toc469581940)

[2.2样例 4](#_Toc469581941)

[3. 异步通知技术接入案例 4](#_Toc469581942)

[3.1商户异步通知接收地址配置 4](#_Toc469581943)

[3.2异步通知商户回调请求编写注意事项及案例 6](#_Toc469581944)

## 文档说明

## 1.1通知功能描述

订单创建事件：

商户调用信用借还订单创建接口（*见信用借还订单创建文档*）后，借用物品的C端用户在芝麻信用借还平台借用成功后，发送的异步通知。

注意：

1. *订单创建成功异步通知保证一定会发，但不保重只发一次，极端情况下可能会发2次及以上，所以商户端做好幂等控制。*
2. *订单创建成功异步通知发送时机是用户在芝麻信用借还平台完成借用后，极端情况下用户可能过了1-2个小时或者1-2天后完成确认借用。所以商户端在这种情况下，一方面是确认用户是否已借用该物品，如果前期未借用成功，请及时调用订单撤销接口（见信用借还订单撤销文档），核销用户在芝麻的借用订单。*

订单完结事件：

1. 借用用户在商户端发起正常归还流程，商户端调用信用借还平台的完结接口（*见信用借还订单完结文档*），信用借还平台完成借用用户的扣款（存在租金或者赔偿金的情况下）后，发送的异步通知。
2. 借用用户如丢失了借用物品，可在信用借还平台订单详情页面发起主动赔偿功能，用户付款完成，也会发送给商户订单完结的异步通知。

注意：

*订单完结可分为正常租金支付归还完结及物品丢失支付赔偿完结，如果商户需要区分这两种不同的完结，需要调用订单查询接口（见信用借还订单查询文档），取得pay\_amount\_type字段，值”RENT”表示租金支付归还完结，值”DAMAGE”表示物品丢失支付赔偿完结。*

## 请求参数说明

## 2.1异步通知参数

| **参数** | **参数名称** | **类型** | **必填** | **描述** | **范例** |
| --- | --- | --- | --- | --- | --- |
| notify\_type | 通知类型 | String（30） | 是 | 取值范围：  ORDER\_CREATE\_NOTIFY  (订单创建异步事件)  ORDER\_COMPLETE\_NOTIFY  (订单完结异步事件) | ORDER\_COMPLETE\_NOTIFY |
| order\_no | 信用借还平台订单号 | String(20) | 是 | 芝麻信用借还平台生成的订单号 | 1231521 |
| out\_order\_no | 外部商户订单号 | String(36) | 是 | 外部商户生成的订单号，与芝麻信用借还平台生成的订单号存在关联关系 | 2016101821001004580200552288 |
| sign | 签名 | String(256) | 是 | 请参考[异步返回结果的验签](https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.QvzKUR&treeId=203&articleId=105286&docType=1#s6) | 601510b7970e52cc63db0f44997cf70e |
| sign\_type | 签名类型 | String(10) | 是 | 签名算法类型，目前支持RSA | RSA |

## 2.2样例

{

"notify\_type":["ORDER\_COMPLETE\_NOTIFY"],

"order\_no":["501"],

"out\_order\_no":["201611230000400200"],

"sign":["ZXdX9nvSG8O9ADgOlTCiKdFtAjJKW5oapxDpHbkuMvlIffznTDn/wo+/PATt3MzpTswQ1BMzNUJQ4zG+BAnVU7Vs9Hi7fL+ofhCVQkrKNBGJnEhujmnhI/qb6kcZbDws0vjGsYXLKrUF02+WeEBHZ5AYOxzjzI/Ke6nVzHFJaQM="],

"sign\_type":["RSA"]

}

## 异步通知技术接入案例

## 3.1商户异步通知接收地址配置

1）首先登录蚂蚁开放平台（https://open.alipay.com/platform/home.htm），利用商户有效的账号登录进入平台。

2）进入后选择研发管理。



3）在“我的应用”中找到签约的芝麻信用借还应用，并点击操作的“查看”按钮。



4）进入到芝麻信用借还应用后，选择“应用环境”菜单，在基础环境中找到“应用网关”并点击修改。



5）在应用网关中填写商户有效的回调地址，并单击“确定”。



## 3.2异步通知商户回调请求编写注意事项及案例

3.2.1回调请求编写

商户提供的回调请求地址需要支撑标准的http或（https）POST请求。

3.2.2异步通知重试策略

1. 程序执行完后必须打印输出“success”（不包含引号）。如果商户反馈给芝麻信用的字符不是success这7个字符，芝麻信用服务器会不断重发通知，直到超过24小时22分钟。一般情况下，25小时以内完成8次通知（通知的间隔频率一般是：4m,10m,10m,1h,2h,6h,15h）；
2. 程序执行完成后，该页面不能执行页面跳转。如果执行页面跳转，芝麻信用会收不到success字符，会被芝麻信用服务器判定为该页面程序运行出现异常，而重发处理结果通知；

3.2.3商户回调请求编写案例

*以java语言代码为例，使用spring mvc框架*

|  |
| --- |
| @Controller  **public** **class** BorrowGoodsNotifyTestController {  /\*\* 日志. \*/  **private** **static** **final** Logger *logger* = LoggerFactory  .*getLogger*(BorrowGoodsNotifyTestController.**class**);  /\*\*  \* 异步通知请求入口.  \* **@param** modelMap  \* **@param** request  \* **@return**  \* **@throws** IOException  \*/  @RequestMapping(value = "borrows/borrowNotifyTest.json", method = {RequestMethod.*POST* })  **public** **void** index(**final** ModelMap modelMap, **final** HttpServletRequest request, HttpServletResponse response) **throws** IOException {  //1、签名验证  ……  //2、通知参数解析  Map<String, String[]> notifyParams = request.getParameterMap();  String notifyParamStr=JSONObject.*toJSONString*(notifyParams);  LoggerUtil.*warn*(*logger*, "异步通知:" + notifyParamStr);    //3、执行业务逻辑  ……  //4、向芝麻反馈处理是否成功  printResponse(response, "success");  }  **protected** **void** printResponse(HttpServletResponse response, String content) **throws** IOException {  PrintWriter writer = **null**;  **try** {  writer = response.getWriter();  writer.write(content);  writer.flush();  } **finally** {  **if** (writer != **null**) {  writer.close();  }  }  }  } |

*POST请求参数打印的内容案例：*

{

"notify\_type":["ORDER\_COMPLETE\_NOTIFY"],

"order\_no":["501"],

"out\_order\_no":["201611230000400200"],

"sign":["ZXdX9nvSG8O9ADgOlTCiKdFtAjJKW5oapxDpHbkuMvlIffznTDn/wo+/PATt3MzpTswQ1BMzNUJQ4zG+BAnVU7Vs9Hi7fL+ofhCVQkrKNBGJnEhujmnhI/qb6kcZbDws0vjGsYXLKrUF02+WeEBHZ5AYOxzjzI/Ke6nVzHFJaQM="],

"sign\_type":["RSA"]

}