

事件上报接口文档

调用步骤

接口简述

事件上报接口

请求

返回

结果码

示例

访问事件

定时到账事件

退货成功事件

手动到账事件 + 确认到账事件

调用步骤

1. 引入依赖

```
1 <dependency>
2   <groupId>cn.com.antcloud.api</groupId>
3   <artifactId>antcloud-api-ent</artifactId>
4   <version>1.3.15</version>
5 </dependency>
6
7 <dependency>
8   <groupId>cn.com.antcloud.api</groupId>
9   <artifactId>antcloud-api-sdk</artifactId>
10  <version>3.4.0</version>
11 </dependency>
```

2. 初始化客户端

使用预先准备的AK/SK初始化客户端。

测试环境联调需要添加IP白名单，请向接口提供方提供调用方 IP

```

1 // 测试环境，请使用以下代码进行线下联调
2 AntFinTechProfile profile = AntFinTechProfile.getProfile(
3     "http://acprodapigw.dl.alipaydev.com",
4     "ACSaQbAqX5E6vV6g",
5     "VUNYGsz8s9UWNQrCHM2tJfez3btSzz6e"
6 );
7
8 // 生产环境，填入AK/SK
9 AntFinTechProfile profile = AntFinTechProfile.getProfile(
10     "https://openapi.antchain.antgroup.com",
11     "<AccessKey>",
12     "<SecretKey>"
13 );
14
15 AntFinTechApiClient client = new AntFinTechApiClient(profile);

```

3. 接口调用：构造请求->调用客户端发送请求并接受返回->读取数据

所有请求务必带上 ProductInstanceId 参数，测试环境请填写：ent-sit，生产环境请填写：ent-prod
 结果为的 resultCode 为 OK 并且 isSuccess 为 true 时，可认为接口调用成功。其他错误请查看
 resultMsg，必要时提供 reqMsgId 给技术支持的同学。

接口简述

名称	Method	Content-Type	描述
EventReport	HTTPS	applications/json	事件上报接口

事件上报接口

请求

字段	类型	必填	描述
eventType	String	是	事件类型，支持类型 VISIT/TRADE
eventInstanceCode	String	是	事件实例编码，访问事件： VISIT:

			<ul style="list-style-type: none"> • CONTRACT_DETAIL_PAGE • SHARE_LANDING_PAGE TRADE: <ul style="list-style-type: none"> • TRADE_SUCCESS • REFUND_SUCCESSES • TRADE_CONFIRM
eventChild	User: <ul style="list-style-type: none"> • userIdNo: 手机号/支付宝登录账号 • userIdType: PHONE/ALIPAY_LOGON_ID 	是	访问者或购买者账号，支付宝登录账号或者手机号
eventParent	User: <ul style="list-style-type: none"> • userIdNo: 手机号/支付宝登录账号 • userIdType: PHONE/ALIPAY_LOGON_ID 	否	分享者账号，支付宝登录账号或者手机号
requestId	String	是	业务单号，同一调用方全局唯一
projectId	String	是	项目ID
confirmType	String	是	<ul style="list-style-type: none"> • 访问事件: DIRECT 直接到账 • 购票事件: TWO_STEP_MANUAL 手动确认到账，TWO_STEP 定时到账
confirmTime	Date	否	确认到账时间，精确到毫秒。 定时到账事件该项必填
tradeOrder	TradeOrder: <ul style="list-style-type: none"> • orderNo: 订单编号，必填。 	否	交易订单。 当事件类型为TRADE时，该项必填

	<ul style="list-style-type: none"> originalTotalAmount: 原始金额，必填。 actualTotalAmount: 实际金额，必填。 itemCount: 数量，必填。 currency: 币种，暂时只支持CNY。 	
--	---	--

返回

字段	类型	描述
eventId	String	事件唯一标识
eventStatus	String	事件状态 <ul style="list-style-type: none"> PROCESSED: 已处理 NEED_RETRY: 需重试 EVENT_NOT_SUPPORTED: 该事件暂不支持。
tokenStatus	String	token状态, <ul style="list-style-type: none"> TOKEN_SUCCESS: 发放成功 TOKEN_FAILED: 发放失败 NO_NEED_ALLOCATION: 无需发放
tokenAmount	Long	token数量，单位分个
message	String	提示信息

结果码

错误码	错误信息	描述
INVALID_PARAMETER	1001	参数错误
LOGON_ID_NOT_FOUND	1005	登录账号不存在
FAILED_NO_RETRY	1010	失败，请勿重试

FAILED_NEED_RETRY	1009	失败，请重试
UNKNOWN	1011	未知错误，请重试

示例

访问事件

```

1 ExecEventReportRequest request = new ExecEventReportRequest();
2 request.setEventType("VISIT");
3 request.setEventInstanceCode("CONTRACT_DETAIL_PAGE"); // 或者 SHARE_LANDING_PAGE
4 User eventChild = new User();
5 eventChild.setUserIdNo("+86-13000000000"); // 大陆手机号格式必须为 +86-13位手机号
6 eventChild.setUserIdType("PHONE");
7 User eventParent = new User();
8 eventParent.setUserIdNo("+86-13000000001");
9 eventParent.setUserIdType("PHONE");
10 request.setEventChild(eventChild);
11 request.setEventParent(eventParent); // 可以为空
12 request.setRequestId("056289b5d0504b35af4418d0520499c0"); // 唯一业务单号，每次事件上报都唯一
13 request.setProjectId("123");
14 request.setConfirmType("DIRECT"); // 访问事件只能使用 DIRECT
15 request.setProductInstanceId("ent-dev"); // dev环境
16
17 ExecEventReportResponse response = client.execute(request);
18 System.out.println(JSON.toJSONString(response));
19 System.out.println(request.getBizId());

```

定时到账事件

```

1 ExecEventReportRequest request = new ExecEventReportRequest();
2 request.setEventType("TRADE");
3 request.setEventInstanceCode("TRADE_SUCCESS"); // 购票成功事件
4 User eventChild = new User();

```

```

5 eventChild.setUserIdNo("+86-13000000000");
6 eventChild.setUserIdType("PHONE");
7 User eventParent = new User();
8 eventParent.setUserIdNo("+86-13000000001");
9 eventParent.setUserIdType("PHONE");
10 request.setEventChild(eventChild);
11 request.setEventParent(eventParent); // 分享人可为空
12 request.setRequestId("056289b5d0504b35af4418d0520499c0"); // 唯一
    业务单号，每次事件上报都唯一
13 request.setProjectId("123");
14 request.setConfirmType("TWO_STEP"); // 定时到账类型
15 request.setConfirmTime(new Date(1609065254000L)); // 定时到账类型事
    件必须传时间
16 TradeOrder tradeOrder = new TradeOrder();
17 tradeOrder.setOrderNo("d1284971f4b64f438124fa5cf8147ef6"); // 购买
    订单号
18 tradeOrder.setOriginalTotalAmount(100L);
19 tradeOrder.setActualTotalAmount(100L);
20 tradeOrder.setItemCount(2L);
21 tradeOrder.setCurrency("CNY");
22 request.setTradeOrder(tradeOrder);
23 request.setProductInstanceId("ent-dev");
24
25 ExecEventReportResponse response = client.execute(request);
26 System.out.println(JSON.toJSONString(response));
27 System.out.println(request.getBizId());
28 System.out.println(tradeOrder.getOrderNo());

```

退货成功事件

```

1 ExecEventReportRequest request = new ExecEventReportRequest();
2 request.setEventType("TRADE");
3 request.setEventInstanceCode("REFUND_SUCCESS"); // 退票成功事件
4 User eventChild = new User();
5 eventChild.setUserIdNo("+86-13000000000");
6 eventChild.setUserIdType("PHONE");
7 User eventParent = new User();
8 eventParent.setUserIdNo("+86-13000000001");

```

```

9 eventParent.setUserIdType("PHONE");
10 request.setEventChild(eventChild);
11 request.setEventParent(eventParent); // 分享人可为空
12 request.setRequestId("056289b5d0504b35af4418d0520499c0"); // 唯一
    业务单号，每次事件上报都唯一
13 request.setProjectId("123");
14 TradeOrder tradeOrder = new TradeOrder();
15 tradeOrder.setOrderNo("d1284971f4b64f438124fa5cf8147ef6"); // 这传
    之前购买成功时传的orderNo
16 tradeOrder.setOriginalTotalAmount(100L);
17 tradeOrder.setActualTotalAmount(100L);
18 tradeOrder.setItemCount(2L);
19 tradeOrder.setCurrency("CNY");
20 request.setTradeOrder(tradeOrder);
21 request.setProductInstanceId("ent-dev");
22
23 ExecEventReportResponse response = client.execute(request);
24 System.out.println(JSON.toJSONString(response));
25 System.out.println(request.getBizId());
26 System.out.println(tradeOrder.getOrderNo());

```

手动到账事件 + 确认到账事件

```

1 // 先上报购买成功事件
2 ExecEventReportRequest request = new ExecEventReportRequest();
3 request.setEventType("TRADE");
4 request.setEventInstanceCode("TRADE_SUCCESS"); // 购票成功事件
5 User eventChild = new User();
6 eventChild.setUserIdNo("+86-13000000000");
7 eventChild.setUserIdType("PHONE");
8 User eventParent = new User();
9 eventParent.setUserIdNo("+86-13000000001");
10 eventParent.setUserIdType("PHONE");
11 request.setEventChild(eventChild);
12 request.setEventParent(eventParent); // 分享人可为空
13 request.setRequestId("056289b5d0504b35af4418d0520499c0"); // 唯一
    业务单号，每次事件上报都唯一
14 request.setProjectId("123");

```

```

15 request.setConfirmType("TWO_STEP_MANUAL"); // 手动到账类型
16 TradeOrder tradeOrder = new TradeOrder();
17 tradeOrder.setOrderNo("d1284971f4b64f438124fa5cf8147ef6"); // 订单
   号
18 tradeOrder.setOriginalTotalAmount(100L);
19 tradeOrder.setActualTotalAmount(100L);
20 tradeOrder.setItemCount(2L);
21 tradeOrder.setCurrency("CNY");
22 request.setTradeOrder(tradeOrder);
23 request.setProductInstanceId("ent-dev");
24
25 ExecEventReportResponse response = client.execute(request);
26 System.out.println(JSON.toJSONString(response));
27 System.out.println(request.getBizId());
28 System.out.println(tradeOrder.getOrderNo());
29
30
31 // 上报确认到账事件
32 ExecEventReportRequest request = new ExecEventReportRequest();
33 request.setEventType("TRADE");
34 request.setEventInstanceCode("TRADE_CONFIRM"); // 购票成功事件
35 User eventChild = new User();
36 eventChild.setUserIdNo("+86-13000000000");
37 eventChild.setUserIdType("PHONE");
38 User eventParent = new User();
39 eventParent.setUserIdNo("+86-13000000001");
40 eventParent.setUserIdType("PHONE");
41 request.setEventChild(eventChild);
42 request.setEventParent(eventParent); // 分享人可为空
43 request.setRequestId("056289b5d0504b35af4418d0520499c0"); // 唯一
   业务单号，每次事件上报都唯一
44 request.setProjectId("123");
45 TradeOrder tradeOrder = new TradeOrder();
46 tradeOrder.setOrderNo("d1284971f4b64f438124fa5cf8147ef6"); // 要确
   认到账的订单号
47 tradeOrder.setOriginalTotalAmount(100L);
48 tradeOrder.setActualTotalAmount(100L);
49 tradeOrder.setItemCount(2L);
50 tradeOrder.setCurrency("CNY");
51 request.setTradeOrder(tradeOrder);

```



```
52 request.setProductInstanceId("ent-dev");  
53  
54 ExecEventReportResponse response = client.execute(request);  
55 System.out.println(JSON.toJSONString(response));  
56 System.out.println(request.getBizId());  
57 System.out.println(tradeOrder.getOrderNo());
```