WebHook接口参数接入文档-V1.0

版本说明

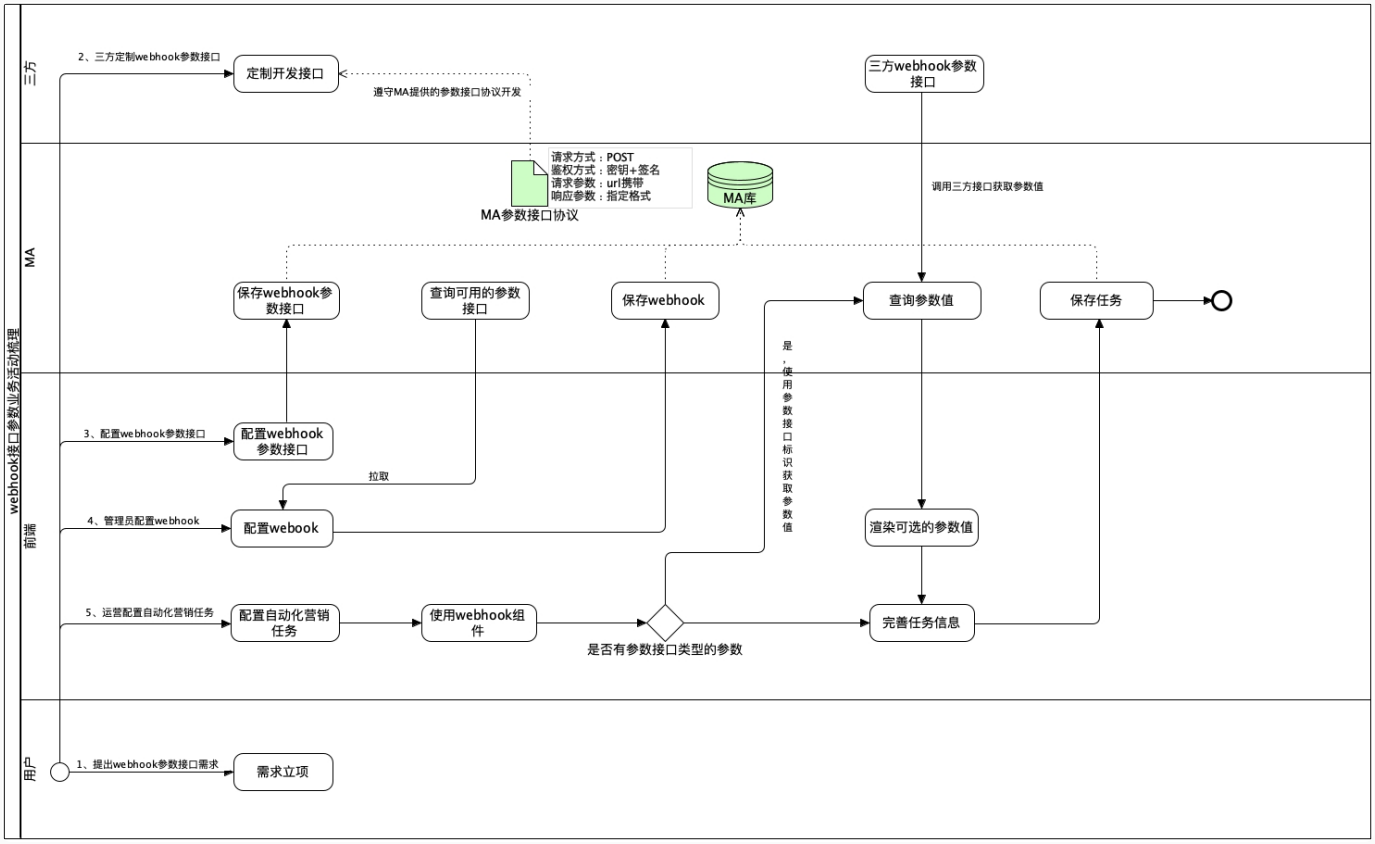
版本号	变更时间	变更说明	变更人
V1.0	2022–04–14	初始化	智岩

一、介绍

背景：运营在使用webhook组件时，时常会需要填写一些参数，用于发送到三方系统，例如发放优惠券场景，需要填写券code，之前运营需要手动输入三方系统的券code，为了简化和减少误操作，产品上提供了支持从三方接口获取优惠券提供给运营下拉选择的功能。

本文档用于描述三方接口如何接入QA WebHook接口参数。

业务活动说明如下图：



二、接入指南

2.1 接口规范

2.1.1 接口请求方式

POST

2.1.2 接口请求入参

接口参数分为三部分：url参数、payload参数、header请求头。

1) url参数

参数名称	类型	是否必传	说明
timestamp	Long	是	时间戳
nonce	String	是	随机数
pn	Long	否	分页参数，第几页，从第1页开始计算。 当不传时，代表不需要分页，返回所有
ps	Integer	否	分页参数，每页查询数量,默认：10
searchText	String	否	搜索参数，一般用于支持模糊搜索的场景，例如：通过searchText的值匹配券名称或者券ID。
paramXXX	String	否	自定义的参数，支持配置多个

> 说明：
> 1、paramXXX说明，当需要使用自定义参数时，可以在QA控制台的webhook接口参数填写接口地址时，通过url参数的形式传递自定义参数

2) payload参数

参数名称	类型	是否必传	说明
version	String	是	版本号
user_profile	Object	是	请求资料信息， 举例： {"id":"5a66f21318c34ea1b9eea0c5b6405708", "type":"API"}

3) header请求头

参数名称	类型	是否必传	说明
X-QA-Hmac-Signature	String	是	鉴权签名

4) 注意事项

- 发送前对url参数进行了unicode编码，接收方需要进行解码

请求示例

HTTP | 复制代码

```
curl  https://api.xxx.com/xxx/getCouponList?timestamp=1649988468&nonce=5a05fc7b594d4f8291f22ce40d17375f&pn=1&ps=10&searchText=\u5e97\u94fa\u5238  -X POST -H "X-QA-Hmac-Signature:dca61467415416fd8d89c3fba0bd5a9e06ab47875e2ee1cf0ec6b0913b6732ee"
```

2.1.3 接口请求出参

responseBody

参数名称	类型	是否必传	说明
content	List<WebhookParamValue>	是	参数值列表
totalCount	long	是	满足的参数值数量

WebhookParamValue

参数名称	类型	是否必传	说明
id	String	是	参数值ID，用于区分
name	String	是	参数值说明，用于展示
ext	Map<String, String>	否	扩展信息，由接口提供方控制

出参示例

出参示例 JSON 复制代码

```
{
  "totalCount":1,
  "content":[
    {
      "id":"code1",
      "name":"100-10店铺券(ID:code1)",
      "ext":{
          "ext1":"扩展信息"
       },
    }
  ]
}
```

2.2 接口鉴权

为了保证数据的安全性，QA和三方提供的接口的交互需要支持鉴权。

2.2.1 密钥+签名方式

密钥+签名的方式，采用HMAC-SHA256算法实现签名鉴权。

1）前提

引入签名算法依赖

maven依赖 XML 复制代码

```xml
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>1.11</version>
</dependency>
```

HMAC-SHA256算法使用示例

示例代码 XML 复制代码

```
/**
* secret为业务双方约定的密钥，不可对外泄露
* plaintext为参数的明文
* sign签名值
* 调用方需要传递sign和plaintext给被调用方
* 被调用方重复对明文签名后，比对sign值，若不一样，说明数据源不一致
* 如果对接业务系统较多，建议新建一个表格来存储对secret的配置。
*/
String sign = HmacUtils.hmacSha256Hex(secret, plaintext);
```

2）鉴权说明

1、从请求头中取出鉴权签名

QA发起的请求中的鉴权签名通过请求头中的X-QA-Hmac-Signature传递

2、生成签名

具体操作：

（1）将密钥、时间戳、随机数三个字符串进行排序，然后连接成一个字符串。

（2）通过HmacUtils.hmacSha256Hex函数使用密钥对该字符串签名

示例代码 Java 复制代码

```java
public static String generateSignature(String key, String timestamp,
String nonce){
    String[] array = new String[] { key, timestamp, nonce};
    StringBuffer sb = new StringBuffer();
    // 字符串排序
    Arrays.sort(array);
    for (int i = 0; i < 3; i++) {
        sb.append(array[i]);
    }
    String plantText=sb.toString();
    String sign = HmacUtils.hmacSha256Hex(plantText,
str.replaceAll("\\s+", ""));
    return sign;
}
```

> 说明：
> 1、密钥由三方定义，在QA的webhook接口参数中录入。
> 2、时间戳在请求的url参数中携带，通过timestamp获取
> 3、随机数在请求的url参数中携带，通过nonce获取

3、鉴权

比较请求携带的签名和生成的签名是否一致，一致代表鉴权通过，不一致代表鉴权失败。

示例代码 Java 复制代码

```java
public static String auth(String requestSign, String targetSign) {
    return targetSign.equals(requestSign);
}
```