

卡券服务接入文档

v 0.0.7

by 欧莱雅IT

修订历史

版本	修订说明	提交人	生效日期
0.0.1	初始化创建文档	Larry Yu	
0.0.2	针对第一次多方会议后提出的建议做了修改。 卡券结构增加了ConsumerRefID, ChannelID。 核销时增加了extraInfo的参数，可以通过长轮询获取到。 核销接口根据卡券类型核销时，支持批量核销。 新增查询卡券类型接口	Larry Yu	
0.0.3	增加测试环境信息。 【TBD: 增加核销时extraInfo的加密存储】	Larry Yu	
0.0.4	增加环境测试用户testuser	Larry Yu	
0.0.5	增加环境测试测试client	Larry Yu	
0.0.6	更新host, authurl域名	Larry Yu	
0.0.7	完善批量操作时的错误描述	Larry Yu	

卡券服务接入文档

[修订历史](#)

[引言](#)

[目的](#)

[准备工作](#)

[测试环境环境变量：](#)

[接入认证服务器](#)

[关于oAuth2的一些术语：](#)

[用户角色表：](#)

[客户端表：](#)

[认证服务器的Endpoints：](#)

[接入卡券服务](#)

[卡券类型数据结构](#)

[json字典样本](#)

[字典解释](#)

[卡券数据结构](#)

[json字典样本](#)

[字典解释](#)

卡券规则

- 卡券领用次数限制
- 卡券在自然时间单位内兑换
- 卡券核销次数限制
- 卡券核销时间
- 同品牌核销限制

API 列表

获取卡券类型列表

- Headers
- Response Status Code
- Response Body

创建卡券

- Headers
- Request Body
- Response Status Code
- Response Body

查询某张卡券

- Headers
- Response Status Code
- Response Body

查询用户的卡券（列表）

- Headers
- Response Status Code
- Response Body

核销卡券

- Headers
- Request Body
- extraInfo 的详细解释
- Response Status Code
- Response Body

轮询卡券消息

- Headers
- Response Status Code
- Response Body

错误表

- 错误格式
- 错误列表 - 持续增加中

引言

目的

为了第三方系统接入卡券服务，本文档介绍卡券服务以及如何和卡券服务进行集成。

【注意：卡券服务仍在第一次开发中，API可能会有变动。】

准备工作

为了更好地理解本文档，可以使用Postman来测试卡券服务的API。

请使用Postman的 Import 功能导入卡券服务API的Collection，Link：<https://www.getpostman.com/collections/77aaee6262e1f91b7c0a>。

注意：使用Postman，请确保您已经理解Postman的基本功能：

导入Collection，配置环境变量，使用内置的oAuth2功能刷新token。

测试环境环境变量：

KEY	VALUE	备注
host	https://dl-api-uat.lorealchina.com/ceh/cvc	卡券服务的host，注意：正式环境会是https协议。
authurl	https://dl-api-uat.lorealchina.com/auth/realm/Lorealcn/protocol/openid-connect	认证服务的url，注意：正式环境会是https协议。
clientsecret	869e5092-3861-4745-b0ef-aa90195a8847	测试用的app的clientsecret。注意：正式环境需要独立申请app。
clientid	someapp	测试用的app的clientid。注意：正式环境需要独立申请app。

测试环境测试用户名：**testuser**

测试环境测试用户密码：**testuser**

注意：该用户是全功能的用户，拥有所有的访问权限。

接入认证服务器

认证服务器管理访问卡券服务的用户，认证服务器提供oAuth2 服务。客户端可以使用认证服务器的 endpoints 获得相关服务。

关于oAuth2的一些术语：

术语	解释	备注
用户	在认证服务注册的用户，用户会有角色。	
客户端	访问认证服务和卡券中心的应用，应用需要在认证服务注册。	比如campaign tools或者云积分都是一个应用。

用户角色表：

角色	解释	备注
coupon_listener	可以监听卡券服务的事件，比如核销事件。	目前客户端不用关心角色
coupon_issuer	可以在卡券服务签发卡券	
coupon_redeemer	可以核销卡券	

客户端表：

客户端	解释	备注
campaign_tool	这里举个例子，可能是其他的名字。	对应客户端名字(client id)，会有client secret. 应用开发者会拿到该信息用来获取用户token
yun-ji-fen	云积分	

认证服务器的Endpoints：

Endpoint	用途	备注
{host}/Loreal/protocol/openid-connect/token	获取jwt加密的token	请参考Postman内任一API的获取token的例子。

要获取到用户的token，应用需要用户名，用户密码，**client id**，**client secret**。

注意：Grant Type为 **Password Credentials**。

获取到token后，可以简单地在 <https://jwt.io/> 查看该token自包含的一些信息。

接入卡券服务

卡券服务以Http API的方式提供服务。卡券服务的API尽量使用HTTP协议来描述。

卡券类型数据结构

卡券服务返回给前端的卡券类型结构。

注意：目前对第三方开发团队比较有用的是卡券类型id，**name**，**description**等，其他的数据卡券中心尚未内部支持，可能会有变更，先别使用。

json字典样本

```
{
  "name": "欧莱雅入会礼",
```

```
"id": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
"template_id": "dba25cb3-4ad9-44c2-8815-6d8b12c10f5a",
"description": "欧莱雅的入会礼券，凭此券可以联系欧莱雅获取礼品。注意：需要自付邮费。",
"internal_description": "这是发布模板的内部描述，给模板发布者以及开发者开发时的参考。不会显示给终端用户",
"state": 0,
"publisher": "ff27204e-6ef2-48e2-a437-7e48cc49d659",
"visible_start_time": "2019-01-01T00:00:00+08:00",
"visible_end_time": "2030-01-31T23:59:59+08:00",
"created_time": "2019-12-12T15:12:12+08:00",
"deleted_time": null,
"rules": {
  "REDEEM_PERIOD_WITH_OFFSET": "{ \"offSetFromAppliedDay\": 0, \"timeSpan\": 365 }",
  "APPLY_TIMES": "{ \"inDays\": 0, \"times\": 300 }",
  "REDEEM_TIMES": "{ \"times\": 3 }",
  "REDEEM_IN_CURRENT_NATURE_MONTH_SEASON_YEAR": "{ \"unit\": \"MONTH\", \"endInAdvance\": 0 }",
  "REDEEM_BY_SAME_BRAND": "{ \"brand\": \"\" }"
}
```

字典解释

Key	描述	备注
name	卡券类型名称	
id	卡券类型ID	卡券类型唯一id
template_id	卡券类型模板id	预留。暂时未启用。
description	卡券的描述	如果未来有类似微信卡包的app，那么打开卡包，查看卡片，卡片上显示的描述内容。
internal_description	卡券类型的内部描述	这是发布模板的内部描述，给模板发布者以及开发者开发时的参考。不会显示给终端用户。
state	描述卡券模板的当前状态。	整型数值。 0：处于激活状态，表示可以被用来为消费者申请卡券。 1：被作废。
publisher	卡券类型的发布者	预留。
visible_start_time	建议可以开始展示的时间	假设这样的场景，卡券类型制作好了，但促销活动尚未开始。这里的时间可以设置为促销活动的时间，这样程序可以自动到期显示和隐藏申请卡券的入口。
visible_end_time	建议可以结束展示的时间	
rules	绑定在卡券类型上的规则	这是一个字典结构，Key表示规则的ID, Value是规则体。规则体是一个使用字符串描述的json字典结构。 详见卡券规则章节。

卡券数据结构

卡券服务返回给前端的卡券结构是如下的

json字典样本

```
{
```

```
"ID": "8ed92b78-1cb9-418f-827c-4fa74d1dfe45",
"CouponTypeID": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
"ConsumerID": "larry.yu",
"ConsumerRefID": "wechat123456",
"ChannelID": "LANCOME_JD",
"State": 0,
"Properties": {
  "binding_rule_properties": {
    "REDEEM_BY_SAME_BRAND": {
      "brand": "LANCOME"
    },
    "REDEEM_TIMES": {
      "times": 3
    },
    "REDEEM_PERIOD_WITH_OFFSET": {
      "endTime": "2021-01-16 11:58:47 +08",
      "startTime": "2020-01-17 11:58:47 +08"
    }
  }
},
"CreatedTime": "2020-01-17T11:58:47.4364758+08:00"
}
```

字典解释

Key	描述	备注
ID	卡券ID	每个卡券都有唯一的ID。
CouponTypeID	卡券类型ID	每个卡券属于某个卡券类型，比如入会礼卡券。
ConsumerID	消费者ID	卡券中心唯一识别一位消费者的id。比如是CRM内的用户ID.
ConsumerRefID	消费者的参考ID	消费者的参考ID，比如腾讯系的openid。当用户在欧莱雅解绑又重新绑定后，ConsumerID可能会变化了，但该ID不会变化。
ChannelID	渠道ID	比如某张卡券时由兰蔻线上京东渠道创建的。
State	描述卡券的当前状态。	整型数值。0：处于激活状态，表示可以调用API进行核销，在有效核销次数内，仍然是激活状态。1：被撤回。2：被兑换。当处于该状态时，表示核销次数已达上限。
Properties	卡券的属性集合	binding_rule_properties 是和该卡券有关的各种规则限制等。这是一个字典结构，Key表示规则的ID, Value是规则体。规则体同样也是一个字典结构。 详见卡券规则章节。

卡券规则

应用可以根据卡券的规则结合自身的业务来支持用户的各种操作。

卡券规则在持续增加中。

卡券领用次数限制

```
"APPLY_TIMES": {
  "inDays": 0,
  "times": 300
}
```

表示卡券发布后每人可以领用300次。

卡券在自然时间单位内兑换


```
"REDEEM_IN_CURRENT_NATURE_MONTH_SEASON_YEAR": {
  "unit": "MONTH",
  "endInAdvance": 0
}
```

unit 是时间单位，目前仅支持MONTH, 也就是月，表示在当前月份兑换。
endInAdvance 是否提前结束兑换，单位是天。比如当endInAdvance = 4 时，卡券时6月份申领的，那么截止日期是6月26日（26 = 30-4）。

卡券核销次数限制

```
"REDEEM_TIMES": {
  "times": 3
}
```

表示总共可以兑换3次。。

卡券核销时间

```
"REDEEM_PERIOD_WITH_OFFSET": {
  "startTime": "2020-01-17 21:58:47 +08",
  "endTime": "2021-01-16 21:58:47 +08"
}
```

上面的时间结构为：年-月-日 时:分:秒 时区。（一般来说，中国属于+8区）

同品牌核销限制

```
"REDEEM_BY_SAME_BRAND": {
  "brand": "LANCOME"
}
```

卡券只可以被具有 LANCOME 品牌属性的操作者进行核销。

API 列表

获取卡券类型列表

具有权限的用户可以获取卡券类型列表。
根据获取到的卡券类型可以为消费者创建卡券。

名称	值	备注
资源路径	{{host}}/api/coupons	
Method	GET	

Headers

Headers	值	备注
Authorization	Bearer ey.....	<u>必填项</u> 。 这是一个jwt风格的token
consumerID	Jekey.chen	预留。查询某个用户可以申请的卡券类型。
consumerRefID	wechat_user123456	预留。
channelID	lancome_jd	预留。查询某个渠道可以申请的卡券类型。

Response Status Code

Response Status Code	错误描述	备注
200	成功获取卡券类型列表信息	
400	输入参数格式有错误 或者 参数有错误。	请参阅后面的错误表。
401	用户token无效。	
403	如果用户没有查看卡券列表权限。	
405	Method不被允许	
500	服务器内部出错。	

Response Body

成功调用后，目前将会得到如下的信息。

```
[
  {
    "name": "欧莱雅入会礼",
    "id": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
    "template_id": "dba25cb3-4ad9-44c2-8815-6d8b12c10f5a",
    "description": "欧莱雅的入会礼券，凭此券可以联系欧莱雅获取礼品。注意：需要自付邮费。",
    "internal_description": "这是发布模板的内部描述，给模板发布者以及开发者开发时的参考。不会显示给终端用户",
    "state": 0,
    "publisher": "ff27204e-6ef2-48e2-a437-7e48cc49d659",
    "visible_start_time": "2019-01-01T00:00:00+08:00",
    "visible_end_time": "2030-01-31T23:59:59+08:00",
    "created_time": "2019-12-12T15:12:12+08:00",
    "deleted_time": null,
    "rules": {
      "REDEEM_PERIOD_WITH_OFFSET": "{ \"offSetFromAppliedDay\": 0, \"timeSpan\": 365 }",
      "APPLY_TIMES": "{ \"inDays\": 0, \"times\": 300 }",
      "REDEEM_TIMES": "{ \"times\": 3 }",
```

```
"REDEEM_IN_CURRENT_NATURE_MONTH_SEASON_YEAR": "{ \"unit\": \"MONTH\",
\"endInAdvance\": 0 }",
"REDEEM_BY_SAME_BRAND": "{ \"brand\": \"\" }"
}
}
]
```

当有错误消息返回时，返回如下结构的结果：

```
{
  "error-code": 2001,
  "error-message": "coupon type not found"
}
```

创建卡券

具有权限的用户可以为消费者创建卡券。

创建分为两种形式：

- 一次创建单张卡券。
- 一次为多个用户创建某一类型卡券。

名称	值	备注
资源路径	{{host}}/api/coupons/	请注意资源后面有一个斜杠。
Method	POST	

Headers

Headers	值	备注
Content-Type	application/x-www-form-urlencoded	<u>必填项</u> 。
Authorization	Bearer ey.....	<u>必填项</u> 。 这是一个jwt风格的token

Request Body

Request Body	值	备注
consumerIDs	Tony,tom,jerry,john,wechat_user_xxyy	必填项。 consumerIDs 传递的是消费者的用户id，注意用户名不能有空格，用 英文的逗号 分隔。 如果只需要为一位消费者创建，那就简单传递一个用户id就行，比如 realtrump 注意：不要有换行符之类的其它字符。 单次创建 consumerID 数量上限待定
consumerRefIDs	类似consumerIDs 的值	consumerRefIDs 传递的是消费者的用户附加id。注意id不能有空格，用 英文的逗号 分隔。 如果所有的consumer都没有RefID值，传空值。 只要有一位consumer有RefID，那么该参数就需要和consumerIDs一样，一一对应。 举例：假设consumerIDs = foo,bar,hello 共三个用户，那么 consumerRefIDs的值可能为空值，或者 consumerRefIDs = oof,,olleh 表示foo和hello两个人有RefID, 但bar没有。 consumerRefIDs=oof,olleh 是错误的参数，因为只有两个数据，而 consumerIDs有三个数据。 注意：不要有换行符之类的其它字符。 单次创建 consumerID 数量上限待定
channelID	渠道ID	由业务方定义的一个字符串，表示发行卡券的渠道，比如兰蔻的线上京东渠道。
couponTypeID	678719f5-44a8-4ac8-afd0-288d2f14daf8	必填项。 这是一个UUID类型的值，是卡券类型ID. 目前通过其他方式交付给开发者。

Response Status Code

Response Status Code	错误描述	备注
200	成功创建卡券	
400	输入参数格式有错误 或者 参数有错误	请参阅后面的错误表。
401	用户token无效。	
403	如果用户没有创建卡券权限	
500	服务器内部出错。	

Response Body

成功调用后，目前将会得到如下的信息。

```
[
  {
    "ID": "8ed92b78-1cb9-418f-827c-4fa74d1dfe45",
    "CouponTypeID": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
    "ConsumerID": "larry.yu",
    "ConsumerRefID": "wechat123456",
    "ChannelID": "LANCOME_JD",
    "State": 0,
    "Properties": {
      "binding_rule_properties": {
        "REDEEM_BY_SAME_BRAND": {
```

```

        "brand": "LANCOME"
      },
      "REDEEM_TIMES": {
        "times": 3
      },
      "REDEEM_PERIOD_WITH_OFFSET": {
        "endTime": "2021-01-16 11:58:47 +08",
        "startTime": "2020-01-17 11:58:47 +08"
      }
    },
    "CreatedTime": "2020-01-17T11:58:47.4364758+08:00"
  }
]

```

当有错误返回时，返回如下结构的结果。【请参阅后面错误表部分。其他API同。】
 批量消费者创建的错误TBD.

```

{
  "error-code": 2001,
  "error-message": "coupon type not found"
}

```

或者在批量创建卡券时，可能返回如下的结果：

```

{
  "usera": [
    {
      "Code": 1001,
      "Message": "rule not found"
    },
    {
      "Code": 1000,
      "Message": "rule judge not found"
    }
  ],
  "userb": [
    {
      "Code": 1001,
      "Message": "rule not found"
    },
    {
      "Code": 1000,
      "Message": "rule judge not found"
    }
  ]
}

```

查询某张卡券

具有权限的用户可以查询消费者的一张卡券信息。

名称	值	备注
资源路径	{{host}}/api/coupons/{{couponID}}	{{couponID}}是卡券的ID
Method	GET	

Headers

Headers	值	备注
Authorization	Bearer ey.....	<u>必填项</u> 。 这是一个jwt风格的token

Response Status Code

Response Status Code	错误描述	备注
200	成功获取卡券信息	
400	输入参数格式有错误 或者 参数有错误。	请参阅后面的错误表。
401	用户token无效。	
403	如果用户没有查看卡券权限。	
405	Method不被允许	
500	服务器内部出错。	

Response Body

```
{
  "ID": "8ed92b78-1cb9-418f-827c-4fa74d1dfe45",
  "CouponTypeID": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
  "ConsumerID": "larry.yu",
  "ConsumerRefID": "wechat123456",
  "ChannelID": "LANCOME_JD",
  "State": 0,
  "Properties": {
    "binding_rule_properties": {
      "REDEEM_BY_SAME_BRAND": {
        "brand": "LANCOME"
      },
    },
    "REDEEM_TIMES": {
      "times": 3
    }
  }
}
```

```
    },
    "REDEEM_PERIOD_WITH_OFFSET": {
      "endTime": "2021-01-16 11:58:47 +08",
      "startTime": "2020-01-17 11:58:47 +08"
    }
  },
  "CreatedTime": "2020-01-17T11:58:47.4364758+08:00"
}
```

查询用户的卡券（列表）

具有权限的用户可以查询消费者的一张卡券信息。

名称	值	备注
资源路径	{{host}}/api/coupons/	请注意资源后面有一个斜杠。
Method	GET	

Headers

Headers	值	备注
Authorization	Bearer ey.....	必填项。 这是一个jwt风格的token
consumerID	678719f5-44a8-4ac8-afd0-288d2f14daf8	必填项。 消费者的ID。

Response Status Code

Response Status Code	错误描述	备注
200	成功获取卡券的信息	
400	输入参数格式有错误 或者 参数有错误。	请参阅后面的错误表。
401	用户token无效。	
403	如果用户没有查看卡券权限。	
405	Method不被允许	
500	服务器内部出错。	

Response Body

```
[
  {
    "ID": "8ed92b78-1cb9-418f-827c-4fa74d1dfe45",
```

```
{
  "CouponTypeID": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
  "ConsumerID": "larry.yu",
  "ConsumerRefID": "wechat123456",
  "ChannelID": "LANCOME_JD",
  "State": 0,
  "Properties": {
    "binding_rule_properties": {
      "REDEEM_BY_SAME_BRAND": {
        "brand": "LANCOME"
      },
      "REDEEM_TIMES": {
        "times": 3
      },
      "REDEEM_PERIOD_WITH_OFFSET": {
        "endTime": "2021-01-16 11:58:47 +08",
        "startTime": "2020-01-17 11:58:47 +08"
      }
    }
  },
  "CreatedTime": "2020-01-17T11:58:47.4364758+08:00"
}
```

核销卡券

具有权限的用户可以核销消费者的一张卡券信息。

核销有两种模式：

- 明确核销消费者的一张卡券，需要提交 **couponID**。
- 核销某一类卡券，比如入会礼卡券，需要提交消费者的列表以及 **couponTypeID**。

注意：两种核销模式是**互斥**的，当**couponID**不为空时，那么将只会进行第一种模式。反之进入第二种模式。

名称	值	备注
资源路径	{{host}}/api/redemptions	
Method	POST	

Headers

Headers	值	备注
Content-Type	application/x-www-form-urlencoded	<u>必填项</u> 。
Authorization	Bearer ey.....	<u>必填项</u> 。 这是一个jwt风格的token

Request Body

Request Body	值	备注
consumerIDs	jacky.chen	必填项。 消费者的ID。不论是哪种模式，都需要提交。 核销某个卡券的情况下，consumerIDs的值就是某个用户id，比如 jacky.chen 。 如果是核销某一个类型，可以传入多个消费者，以英文半角逗号（,）分隔，比如 jacky.chen,jet.lee ，如果只有一位消费者，那就类似 jacky.chen 。
couponID	8ed92b78-1cb9-418f-827c-4fa74d1dfe45	卡券ID.
couponTypeID	678719f5-44a8-4ac8-afd0-288d2f14daf8	卡券类型ID. 如果要按照卡券类型ID核销，那么couponID必须为空。
extraInfo	调用方在核销时提供的额外信息	请参考下面的详细结构描述。 可以通过长轮询来获取这些额外信息。

extraInfo 的详细解释

考虑到不同系统间需要传递丰富的数据以完成业务，这里提供extraInfo字段供调用放置额外的信息。

extraInfo字段是一个json 字典风格的字符串。

卡券中心根据现有已知业务预定义一些字段供多方使用，其他的各业务方可以自行协商格式。

预定义的字段都有【**cs_**】前缀。

注意：卡券中心不对**extraInfo**进行校验，在联调时，各方需要注意。

一个典型的extraInfo结构如下：

```
{
  "cs_mobile": "18666666666",
  "cs_tmallID": "tmallid_134345",
  "cs_tmallChannelCode": "some channel code",
  "cs_tmallShopCode": "tmall_lancome",
  "cs_redeemSource": "tmall_lancome_shop",
  "cs_counterCode": "lancome_chengdu_counter456",
  "cs_baCode": "lancome_ba_L09878"
}
```

extraInfo的Key	值示例	备注
cs_mobile	18666666666	字符串类型手机号码
cs_tmallID	tmallid_134345	字符串类型的天猫ID
cs_channelCode	Channel_lancome	字符串类型的渠道编号
cs_tmallShopCode	tmall_lancome	字符串类型的天猫店铺码
cs_redeemSource	tmall_lancome_shop	字符串类型的核销源
cs_counterCode	lancome_chengdu_counter456	字符串类型的柜台编号
cs_baCode	lancome_ba_L09878	字符串类型的BA编号

Response Status Code

Response Status Code	错误描述	备注
200	成功核销卡券	
400	输入参数格式有错误 或者 参数有错误。	请参阅后面的错误表。
401	用户token无效。	
403	如果用户没有核销卡券权限。	
405	Method不被允许	
500	服务器内部出错。	

Response Body

操作成功的情况下，response为空的。当发生错误时，返回如下格式：

```
{
  "error-code": 2001,
  "error-message": "coupon type not found"
}
```

或者在批量核销卡券时，可能返回如下的结果：

```
{
  "usera": [
    {
      "Code": 1001,
      "Message": "rule not found"
    },
    {
```

```
        "Code": 1000,
        "Message": "rule judge not found"
    }
],
"userb": [
    {
        "Code": 1001,
        "Message": "rule not found"
    },
    {
        "Code": 1000,
        "Message": "rule judge not found"
    }
]
}
```

轮询卡券消息

有些应用需要知道卡券是否被核销等信息，该接口提供一个基于Http的长轮询的机制给有接收通知权限的应用。

应用必须在独立的线程发起Http请求，一般会hang住，当卡券中心有新的消息，将会返回消息。比如一个卡券核销的消息。

应用必须确保请求不中断，当成功获得消息或者因为任何原因中断连接，应用必须立即恢复轮询。

目前仅提供核销的消息。

名称	值	备注
资源路径	{{host}}/api/events/	
Method	GET	

Headers

Headers	值	备注
Authorization	Bearer ey.....	<u>必填项</u> 。 这是一个jwt风格的token

Response Status Code

Response Status Code	错误描述	备注
200	成功获取一个消息	
400	输入参数格式有错误 或者 参数有错误。	请参阅后面的错误表。
401	用户token无效。	
403	如果用户没有核销卡券权限。	
405	Method不被允许	
500	服务器内部出错。	

Response Body

```
{
  "type": 2,
  "payload": {
    "extrainfo": "678719f5-44a8-4ac8-afd0-288d2f14daf8",
    "coupons": [

    ]
  }
}
```

注意：目前仅提供的type值为2. 不同的值对应的payload结构可能有差别，开发者注意这一点。

Response Body	值	备注
type	消息类型	0：发券，暂不提供。1：作废，暂不提供。2：核销
payload.extrainfo	附加的信息	核销时提供的附加信息。
payload.coupons	卡券列表	被核销的卡券，是一个卡券结构的数组。 需要注意的一点，卡券的state表示最新的卡券状态，如果某个卡券可以多次被核销，那么返回的卡券State可能仍然是0。

错误表

卡券中心在不同的业务中会返回有意义的错误，包含code和描述。应用可以根据code自定义符合自身业务要求的提示。

错误格式

```
{
  "error-code": 2001,
  "error-message": "coupon type not found"
}
```

Response Body	值	备注
error-code	错误码	
error-message	错误消息	

错误列表 - 持续增加中

error-code	error-message	备注
1000	rule judge not found	规则校验没有找到
1001	rule not found	没有找到规则定义
1002	coupon apply times exceeded	
1003	the coupon applied has been expired	卡券模板已经过期
1004	the coupon has a bad formatted rules	卡券的规则格式错误
1005	the coupon has not start the redemption	卡券的核销开始日期还没到
1006	coupon redeem times exceeded	
1007	coupon has no redeem times rule	卡券没有设置核销次数的规则
1008	the coupon is expired	
1009	the coupon redeem time unit unsupport	不支持的自然时间单位，比如不支持旬。
	coupon template not	

1100	found	卡券模板没有找到
1200	coupon id is invalid	
1201	coupon not found	
1202	coupon is not active	
1203	coupon was redeemed	
1204	too much coupons to redeem	按照类型来核销时，一次只能核销一张卡券
1205	the coupon's owner is not the provided consumer	提供的消费者ID和卡券内的消费者ID不一致
1300	consumer id or coupon type is invalid	
1301	consumer id is invalid	
1302	consumer ids and the ref ids mismatch	申领卡券时提供了消费者的ref id，但和消费者id的数量不匹配。比如提供了5个消费者，但是只有4个ref id。
1400	requester was forbidden to do this action	请求者没有权限执行此操作
1401	requester has no brand information	请求者身份中没有品牌信息
1402	redeem coupon with different brand	卡券被不同品牌的请求者核销。比如卡券被设置了同品牌核销的限制，那么可能会发生这样的错误。
1500	validate token failed	token校验失败
1501	the token is expired	token过期了，需要重新获取新的token。

