# 加油券推送接口说明

**（v2.2）**

**版本变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **简述** | **日期** | **修改人** |
| v1.0 | 初稿 | 2016-10-24 | 阮项 |
| V1.1 | 将4.2章节中获取电子全接口发送信息表中的“usercode”改为“userid”。 | 2016-10-24 | 阮项 |
| V1.2 | 1. 发送信息说明中针对时间类型增加了格式说明； 2. result结果码说明中增加了返回类型说明。 | 2016-11-02 | 阮项 |
| V1.3 | 1.获取券类型接口发送信息中的“imageurl”字段由必填项改为可选项。 | 2016-11-05 | 阮项 |
| V1.4 | 修改签名算法 | 2016-11-09 | 乙峰 |
| V1.5 | 增加了获取派发账单和获取已使用账单接口 | 2016-12-05 | 阮项 |
| V1.6 | 在“获取派发账单”和“获取已使用账单接口”上增加了返回值 | 2016-12-06 | 乙峰 |
| V1.7 | 修改了“获取电子券”接口返回信息中的“coucode”说明。由原来的券号改为券号+校验码。 | 2016-12-08 | 阮项 |
| V1.8 | 增加“注销电子券”接口 | 2017-01-10 | 乙峰 |
| V1.9 | “注销电子券”中的“batchno”改为“businessid”，并增加了业务处理逻辑说明。 | 2017-01-16 | 阮项 |
| V2.0 | 增加“电子券状态查询”接口 | 2017-02-15 | 乙峰 |
| V2.1 | 增加电子券使用通知接口规范 | 2017-06-08 | 阮项 |
| V2.2 | 修改了电子券使用通知接口的返回信息，增加错误码。 | 2017-06-11 | 阮项 |

# 概述

安徽石油（有路平台）系统简称A系统，申请券系统简称B系统。

接口调用方式为webapi。

## 接口地址及参数说明

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx &jsondata=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **备注** |
| bizid | 业务参数 | String | getcoutypes: 获取券 |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文 |
| sign | 签名 | String | 使用MD5对传输参数进行签名，签名字串如下：  bizid=#bizid#&jsondata=#jsondata#  &marking=#marking#&secretkey=#secretKey# |

### bizid

业务参数

getcoutypes：获取券类型，B系统获取A系统分配给其的券类型信息。

getcoutypeimg：获取券图片，B系统获取A系统电子券类型图片信息。

getcoupons: 获取电子券，B系统从A系统获取电子券。

statuscallback：状态回调，B系统通过该回调接口设置所获取电子券状态。

queryusedcoupons：已使用券查询，B系统获取某时间段内已经使用的电子券。

### marking

B系统标识

### jsondata

bizid不同，相应的jsondata不同，jsondata需要加密传输调用。

B系统需要使用加密：EcardEncryptUtil. encrypt(s, secretKey);

A系统需要使用解密：EcardEncryptUtil. decrypt(enc, secretKey);

### sign

sign= EcardEncryptUtil. md532(bizid=#bizid#

&jsondata=#jsondata#&marking=#marking#&secretkey=#secretKey#);

#号之间的是参数值

## 返回结果参数说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **备注** |
| result | 返回结果码 | String |  |
| msg | 返回结果码说明 | String |  |
| jsonresult | 业务数据 | String |  |
| sign | 签名 |  | 上述三个值加密后计算出的sign签名 |

result、msg、jsonresult 加密数据

### result结果码说明

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **结果编码** | **结果信息** | **备注** |
| 1 | 0000 | 成功 |  |
| 2 | 0001 | sign签名错误 |  |
| 3 | 0002 | jsondata参数错误 |  |
| 4 | 0003 | 电子券库存不足 |  |
| 5 | 0004 | 电子券状态错误 |  |
| 6 | 0005 | 券类型不存在 |  |
| 7 | 1000 | 参数错误 |  |
| 8 | 1001 | 异常 |  |

### jsonresult

应答的具体业务数据。请查看具体的业务结果jsonresult串。

## 接口调用流程

B系统按照A系统规范传递相应的bizid,marking,jsondata,sign数据。

上线前约定双方需要约定marking值以及加解密算法用到的secretKey，secretKey示例如下:

secretKey=”f8ee541137a2aa381abaac17886653ba”

应用示例如下：

String jsondata=EcardEncryptUtil.encrypt(param, secretKey);

String sign = EcardEncryptUtil.md532(jsondata);

接口调用时，A系统执行以下步骤：

1. 校验marking是否是与A系统存在接口的合法系统；
2. 按sign规则校验sign是否正确，防篡改；
3. 解密jsondata获得业务参数；
4. 根据bizid，处理jsondata反馈结果。

B系统接收到应答结果，首先对应答结果进行sign值校验，保证数据传输过程中没有错误。

解密结果串后进行后续处理。

## 签名算法及加解密算法

|  |
| --- |
| */\*\*  \** 签名算法及加解密算法 *\* Created by qcsea on 2016/4/18.  \*/* **import** java.io.ByteArrayOutputStream; **import** java.io.UnsupportedEncodingException; **import** java.security.MessageDigest;  */\*\*  \* 采用MD5加密解密  \*  \** ***@author*** *Qcsea  \** ***@datetime*** *2016-04-06  \*/* **public class** EcardEncryptUtil {  */\*\*\*  \* MD5加码 生成32位md5码  \*/* **public static** String md532(String inStr) {  MessageDigest md5 = **null**;  **try** {  md5 = MessageDigest.*getInstance*(**"MD5"**);   } **catch** (Exception e) {  System.***out***.println(e.toString());  e.printStackTrace();  **return ""**;  }  **char**[] charArray = inStr.toCharArray();  **byte**[] byteArray = **new byte**[charArray.**length**];   **for** (**int** i = 0; i < charArray.**length**; i++)  byteArray[i] = (**byte**) charArray[i];  **byte**[] md5Bytes = md5.digest(byteArray);  StringBuffer hexValue = **new** StringBuffer();  **for** (**int** i = 0; i < md5Bytes.**length**; i++) {  **int** val = ((**int**) md5Bytes[i]) & 0xff;  **if** (val < 16)  hexValue.append(**"0"**);  hexValue.append(Integer.*toHexString*(val));  }  **return** hexValue.toString();  }   */\*\*  \* 加密解密算法  \** ***@param inStr*** *加密字符串  \** ***@param secretKey*** *秘钥  \* 算法：  \* 1：加密字符串和秘钥转换成字符数组；  \* 2：秘钥去重复  \* 3：循环一（秘钥字符串数组）{ 循环二（加密字符串数组）{  \* 秘钥字符的ASC码 与 加密字符的ASC码 进行二进制异或运算  \* }  \* }  \* 4：把字符串转为16进制  \*/* **private static** String convert(String inStr, String secretKey) {  **char**[] a = inStr.toCharArray();  **char**[] s = *rmRepeated*(secretKey).toCharArray();  **for** (**int** i = 0; i < s.**length**; i++) {  **for** (**int** j = 0; j < a.**length**; j++) {  a[j] = (**char**) (a[j] ^ s[i]);  }  }  String r = **new** String(a);  **return** r;  }  */\*\*  \* 清除字符串中重复字母算法  \** ***@param s*** *\** ***@return*** *\*/* **private static** String rmRepeated(String s) {  **int** len = s.length();  **int** k = 0;  **int** count = 0;  String str = **""**;  **char**[] c = **new char**[len];  **for** (**int** i = 0; i < len; i++) {  c[i] = s.charAt(i);  }  **for** (**int** i = 0; i < len; i++) {  k = i + 1;  **while** (k < len - count) {  **if** (c[i] == c[k]) {  **for** (**int** j = k; j < len - 1; j++) {  c[j] = c[j + 1];*// 出现重复字母，从k位置开始将数组往前挪位* }  count++;*// 重复字母出现的次数* k--;  }  k++;  }  }  **for** (**int** i = 0; i < len - count; i++) {  str += String.*valueOf*(c[i]);  }  **return** str;  }  */\*  \* 将字符串编码成16进制数字,适用于所有字符（包括中文）  \*/* **private static** String *hexString* = **"0123456789ABCDEF"**;  **public static** String encode(String str) {  *// 根据默认编码获取字节数组* String r=**""**;  **try** {  **byte**[] bytes = str.getBytes(**"UTF-8"**);  StringBuilder sb = **new** StringBuilder(bytes.**length** \* 2);  *// 将字节数组中每个字节拆解成2位16进制整数* **for** (**int** i = 0; i < bytes.**length**; i++) {  sb.append(*hexString*.charAt((bytes[i] & 0xf0) >> 4));  sb.append(*hexString*.charAt((bytes[i] & 0x0f) >> 0));  }  r=sb.toString();  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  **return** r;  }  */\*  \* 将16进制数字解码成字符串,适用于所有字符（包括中文）  \*/* **public static** String decode(String bytes) {  String r=**""**;  **try** {  ByteArrayOutputStream baos = **new** ByteArrayOutputStream(bytes.length() / 2);  *// 将每2位16进制整数组装成一个字节* **for** (**int** i = 0; i < bytes.length(); i += 2){  baos.write((*hexString*.indexOf(bytes.charAt(i)) << 4 | *hexString*.indexOf(bytes.charAt(i + 1))));  }  r= **new** String(baos.toByteArray(),**"UTF-8"**);  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  **return** r;  }  */\*\*  \* 加密  \*  \** ***@param inStr*** *\* 原字符串  \** ***@param secretKey*** *\* 秘钥  \** ***@return*** *\*/* **public static** String encrypt(String inStr, String secretKey) {  String hexStr=*convert*(inStr, secretKey);  **return** *encode*(hexStr);  }   */\*\*  \* 解密  \*  \** ***@param inStr*** *\* 原字符串  \** ***@param secretKey*** *\* 秘钥  \** ***@return*** *\*/* **public static** String decrypt(String inStr, String secretKey) {  String hexStr=*decode*(inStr);  **return** *convert*(hexStr, secretKey);  }   *// 测试主函数* **public static void** main(String args[]) {  String s = new String("12bvdde`中户人民共和国，1234@￥#%&\*（）-=|+\_}{[]/.,;:,.>》》。，《dkfjaskfaskdjfkdasj");  String md5 = md532(s);  String enc = encrypt(s, "f8ee541137a2aa381abaac17886653ba");  String dnc = decrypt(enc, "f8ee541137a2aa381abaac17886653ba");  System.out.println("原始：" + s);  System.out.println("MD5后：" + md5);  System.out.println("加密的：" + enc);  System.out.println("解密的：" + dnc);  } } |

# 获取券类型

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx&jsondata=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | getcoutypes |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | (M必填，O选填) | 备注 |
| applytime | DateTime | M | 申请时间,东8区标准时间 格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

Json格式如下：

{

"applytime": "",

"checkcode ": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 备注 |
| typecode | Int | 券类型编码-8位 |
| typetitle | String | 券类型名称 |
| typealias | String | 券类型别名 |
| moneytype | int | 券面额类型：0-固定金额；2-动态金额。 |
| facevalue | Int | 券面值金额 |
| enabletime | DateTime | 生效时间 |
| disabletime | DateTime | 失效时间 |
| imageurl | String | 券图片路径 |
| usedeclare | String | 券使用说明 |
| total | Int | 条数 |

结果示例：

{

"rows":[

{

"typecode":"34000028"

"typetitle":"50元代金券",

"typealias":"wsydjq",

"moneytype":0,

"facevalue":50,

"enabletime":2016-10-16 00:00:00,

"disabletime":2017-10-16 00:00:00,

"imageurl":"/img/cou/daijinquan.png",

"usedeclare":"使用说明文字描述。"

},

{

"typecode":"34000028"

"typetitle":"50元代金券",

"typealias":"wsydjq",

"moneytype":0,

"facevalue":50,

"enabletime":2016-10-16 00:00:00,

"disabletime":2017-10-16 00:00:00,

"imageurl":"/img/cou/daijinquan.png",

"usedeclare":"使用说明文字描述。"

}

],

"total":2

}

# 获取券图片

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx&jsondata=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | getcoutypeimg |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | (M必填，O选填) | 备注 |
| typealias | String | M | 券类型别名 |
| imageurl | String | O | 券图片 |
| applytime | DateTime | M | 申请时间,东8区标准时间 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

Json格式如下：

{

"typealias": "",

"imageurl":"",

"applytime": "",

"checkcode ": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| typealias | String | 券类型别名 |
| imagefile | String | 图片文件 |

结果示例：

{

"typealias":"",

"imagefile":""

}

# 获取电子券

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx&jsondata=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | getcoupons |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| businessid | String | M | 保单ID或保险公司的交易唯一标识，供未来对账使用 |
| userid | String | M | 客户ID |
| orgcode | String | M | 机构编码-具体值见附录1 |
| typealias | String | M | 券类型别名 |
| moneytype | String | M | 券面额类型：0-固定金额；2-动态金额 |
| facevalue | String | O | 电子券金额，moneytype为0时，不填；moneytype为2时，必填。 |
| num | String | M | 张数，最小1张，最大100张。 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

Json格式如下：

{

"businessid": "100000031234198751",

"userid": "123321",

"orgcode": "100098",

"coupons": [

{

"typealias":"paichbwsy",

"moneytype":"0",

"facevalue":"" ,

"num":"1"

},

{

"typealias":"paichbwsy",

"moneytype":"0",

"facevalue":"" ,

"num":"1"

}

],

"applytime": "",

"checkcode ": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| batchno | String | 派发批次号 |
| createtime | DateTime | 生成券时间 |
| coucode | String | 共25位，由券编号和校验码组成，格式18位券编号+“.”+6位校验码。 |
| facevalue | Int | 券面值 |
| coustartdate | DateTime | 可用开始时间 |
| couenddate | DateTime | 可用结束时间 |
| imageurl | String | 券图片路径 |
| usedeclare | String | 券使用说明 |
| coustatus | Int | 券状态1：派发中 |
| total | Int | 条数 |

结果示例：

{

"batchno":"0000000001",

"rows":[

{

"createtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1},

"coucode":"3400000114754965052431224",

"facevalue":50,

"coustartdate":2016-10-16,

"couenddate":2016-11-16,

"imageurl":"/img/cou/daijinquan.png",

"usedeclare":"使用说明文字描述。",

"coustatus":1

},

{

"createtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1},

"coucode":"3400000114754965052431224",

"facevalue":50,

"coustartdate":2016-10-16,

"couenddate":2016-11-16,

"imageurl":"/img/cou/daijinquan.png",

"usedeclare":"使用说明文字描述。",

"coustatus":1

},

{

"createtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1},

"coucode":"3400000114754965052431224",

"facevalue":50,

"coustartdate":2016-10-16,

"couenddate":2016-11-16,

"imageurl":"/img/cou/daijinquan.png",

"usedeclare":"使用说明文字描述。",

"coustatus":1

}

],

"total":3

}

# 状态回调

B系统将所获取的券成功派发给客户后，调用该接口以更新券状态为已派发状态。

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx &jsonData=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | statuscallback |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

A系统需要校验申请券状态是不是全部为1-派发中状态。如果是，将上次所申请的券设置为2-已派发未激活状态。否则，应答错误信息。

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| batchno | String | M | 流水号，申请券时A系统返回的标示值 |
| businessid | String | M | 保单ID或保险公司的交易唯一标识，供未来对账使用 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

{

"batchno": "",

"businessid": "",

"applytime":,

"checkcode": ""

}

## 返回信息（jsonresult）说明

空串。

# 注销电子券

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx&jsondata=xxxxx&sign=xxxx

业务处理逻辑说明：

举例：同一个流水号下有10张电子券，只传流水号过来，如下处理：

1、10张券如果有部分券已经使用，则将未使用的券注销，并返回所有注销的券列表。

2、10张券如果有部分券已经注销过，则剩下的未使用并且没有注销的券注销成功，并返回所有注销的券列表。

传流水号&券号过来，如下处理：

1、要求券号和流水号必须是对应的，不对应则注销失败。

2、只注销该流水号下传过来的电子券，没有传过来的电子券不做处理也不判断状态。

3、如果传过来的券号列表中，有部分已经使用，则将未使用的券注销，并返回所有注销的券列表。

4、如果传过来的券号列表中，有部分已经注销，则剩下的未使用并且没有注销的券注销成功，并返回所有注销的券列表。

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | cancelcoupons |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| businessid | String | M | 业务流水号，保单ID或保险公司的交易唯一标识，供未来对账使用 |
| coupons | String[] | M | 券号列表：格式18位券编号，可为空 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

Json格式如下：

{

"businessid": "",

"coupons": ["coucode1", "coucode2", "coucode3"],

"applytime": "",

"checkcode ": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| businessid | String | 业务流水号，保单ID或保险公司的交易唯一标识，供未来对账使用 |
| coucode | String | 所有注销成功的券编号，格式18位券编号。 |
| coustatus | Int | 券状态9：失效 |
| total | Int | 条数 |

结果示例：

{

"businessid":"584995",

"rows":[

{

"coucode":"3400000114754965052431224",

"coustatus":9

},

{

"coucode":"3400000114754965052431224",

"coustatus":9

},

{

"coucode":"3400000114754965052431224",

"coustatus":9

}

],

"total":3

}

# 电子券状态查询

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx&jsondata=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | couponstatus |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| businessid | String | M | 业务流水号，保单ID或保险公司的交易唯一标识，供未来对账使用 |
| coupons | String[] | M | 券号列表：格式18位券编号 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

Json格式如下：

{

"businessid": "",

"coupons": ["coucode1", "coucode2", "coucode3"],

"applytime": "",

"checkcode ": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| businessid | String | 业务流水号，保单ID或保险公司的交易唯一标识，供未来对账使用 |
| coucode | String | 格式18位券编号。 |
| coustatus | Int | 券状态。1-派发中，2:-已派发未激活，3-已派发已激活，5-已使用，9-已注销 |
| total | Int | 条数 |

结果示例：

{

"businessid":"0000000001",

"rows":[

{

"coucode":"3400000114754965052431224",

"coustatus":9

},

{

"coucode":"3400000114754965052431224",

"coustatus":9

},

{

"coucode":"3400000114754965052431224",

"coustatus":9

}

],

"total":3

}

# 已使用券查询

查询某个时间段客户已经使用的电子券信息，支持分页查询，一次最多返回2000条数据。

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx &jsonData=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | queryusedcoupons |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| pagenum | Int | O | 页码，默认为0 |
| pagesize | Int | O | 条数，默认2000 |
| startusedtime | DateTime | M | 开始时间，格式2016-01-01 13:10:00 |
| endusedtime | DateTime | M | 结束时间，格式2016-01-01 12:59:59 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

{

"pagenum": ,

"pagesize":,

"startusedtime":,

"endusedtime":,

"applytime":

"checkcode": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| coucode | String | 18位券编号 |
| usedtime | Int | 使用时间 |
| total | Int | 总条数 |

结果示例：

{

"rows":[

{

"itemno":"130000103016090990A171",

"usedtime":2016-10-19 12:90:12

},

{

"itemno":"130000103016090990A171",

"usedtime":2016-10-19 12:90:12

},

{

"itemno":"130000103016090990A171",

"usedtime":2016-10-19 12:90:12

},

…..

{

"itemno":"130000103016090990A171",

"usedtime":2016-10-19 12:90:12

}

],

"total":2000

}

# 获取派发账单

每天凌晨获取前一天已派发电子券账单，支持分页查询，一次最多返回2000条数据。

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx &jsonData=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | getdistbill |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| pagenum | Int | O | 页码，每页2000条 |
| billdate | String | M | 账单日期，格式20160603 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

{

"pagenum": 1,

"billdate": "20160603",

"applytime":2016-10-16 18:31:21,

"checkcode": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| coucode | String | 18位券编号 |
| businessid | String | 保单ID或保险公司的交易唯一标识，供未来对账使用 |
| batchno | String | 流水号 |
| disttime | DateTime | 派发时间； |
| totalpage | Int | 总页数 |
| currentpage | Int | 当前页数 |
| billdate | String | 账单日 |
| watermark | String | jsonresult数据json序列化的md5签名 |

结果示例：

{

"rows":[

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"disttime":2016-10-19 12:90:12

},

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"disttime":2016-10-19 12:90:12

},

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"disttime":2016-10-19 12:90:12

},

…..

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"disttime":2016-10-19 12:90:12

}

],

"totalpage":20,

"currentpage":1,

"billdate":"20161110",

"watermark":""

}

# 获取已使用账单

每天凌晨获取前一天已使用电子券账单，支持分页查询，一次最多返回2000条数据。

## 接口参数

A系统对外提供的接口地址如下：

http://服务器地址/api/coupon?bizid=xxx&marking=xx &jsonData=xxxxx&sign=xxxx

|  |  |  |  |
| --- | --- | --- | --- |
| **英文名称** | **中文名称** | **类型** | **值** |
| bizid | 业务参数 | String | getusedbill |
| marking | 系统标识 | String | 接口双方上线时约定： |
| jsondata | 业务数据 | String | 表单业务数据密文，见发送信息（jsondata）说明。 |
| sign | 签名 | String | 使用MD5对传输的jsondata参数进行签名。 |

## 发送信息（jsondata）说明

具体参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| pagenum | Int | O | 页码,每页2000条 |
| billdate | String | M | 账单日期，格式20160603 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

{

"pagenum": 1,

"billdate": "20160603",

"applytime":2016-10-16 18:31:21,

"checkcode": ""

}

## 返回信息（jsonresult）说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| coucode | String | 18位券编号 |
| businessid | String | 保单ID或保险公司的交易唯一标识，供未来对账使用 |
| batchno | String | 流水号 |
| usedtime | DateTime | 使用时间； |
| totalpage | Int | 总页数 |
| currentpage | Int | 当前页数 |
| billdate | String | 账单日 |
| watermark | String | jsonresult数据json序列化的md5签名 |

结果示例：

{

"rows":[

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"usedtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1}

},

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"usedtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1}

},

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"usedtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1}

},

…..

{

"coucode":"130000103016090990A171",

"businessid":"12345678",

"batchno":"0000000001",

"usedtime":{"date":13,"hours":17,"seconds":27,"month":1,"timezoneOffset":-480,"year":117,"minutes":2,"time":1486976547000,"day":1}

}

],

"totalpage":20,

"currentpage":1,

"billdate":"20161110",

"watermark":""

}

# 电子券使用通知

客户在加油站使用电子券成功后，A系统会立即将电子券的使用信息发送给B系统。B系统需要接收处理，并返回应答。

不管A系统收到B系统的应答是成功、失败或者超时，都不影响A系统电子券的使用。针对B系统返回失败或者超时的情况，A系统会在每天凌晨零点零5分对上一个自然日所有发送失败或者超时的通知，统一做一次补充发送，以尽可能的减少发送通知的笔数。

**注意：A系统多次发送通知，B系统应该能正确处理重复的通知。针对电子券使用时间，应该以最后一次通知的时间为准。**

接口统一采用HTTP请求，POST提交方式，发送和返回数据都为JSON格式。统一采用UTF-8字符编码。

## 发送信息（jsondata）说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| coucode | String | M | 电子券编号 |
| businessid | String | M | 业务交易号 |
| stationcode | String | M | 油站编码 |
| usedtime | DateTime | M | 使用时间，格式：2016-10-16 18:31:21 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

## 返回信息（jsonresult）说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **(M必填，O选填)** | **备注** |
| returncode | String | M | 响应码；SUCCESS/FAILED，是否成功通过此编码反应。 |
| failurecode | String | O | 错误码。返回失败是填写 |
| failuremsg | String | O | 错误描述。返回失败是填写 |
| applytime | DateTime | M | 申请时间,东8区标准时间，格式：2016-10-16 18:31:21 |
| checkcode | String | M | 23位，13位时间戳+10位数字随机码 |

# 附录-1 机构

|  |  |
| --- | --- |
| **机构编码** | **机构名称** |
| 200 | 安徽石油分公司 |
| 200001 | 合肥石油分公司 |
| 200002 | 阜阳石油分公司 |
| 200003 | 池州石油分公司 |
| 200004 | 马鞍山石油分公司 |
| 200005 | 六安石油分公司 |
| 200006 | 安庆石油分公司 |
| 200007 | 蚌埠石油分公司 |
| 200008 | 芜湖石油分公司 |
| 200009 | 宣城石油分公司 |
| 200010 | 亳州石油分公司 |
| 200011 | 宿州石油分公司 |
| 200013 | 滁州石油分公司 |
| 200014 | 黄山石油分公司 |
| 200016 | 铜陵石油分公司 |
| 200017 | 淮北石油分公司 |
| 200018 | 淮南石油分公司 |