# 积分API说明

Steven 创立于2018/01/27

版本0.03版，一些没有说明白的，以后的文档会继续说明

修改历史：

…

2018.01.28 验证码发送，注册

2018.01.29 注册登录

2018.01.30 修改验证码发送，注册

2018.01.30 修改测试ip地址

2018.01.30 修改首次流程

2018.01.31 修改modityPassword单词错误

2018.02.01 修改忘记密码请求路径

2018.02.02 添加查看反馈信息接口

2018.02.04 增加3.14测试接口，可以查询全部用户登陆信息

2018.02.04 增加3.10积分钱包分页

2018.02.04 增加3.11捐赠记录分页

2018.02.04 增加3.15获取闪屏页广告

2018.02.04 增加3.16积分页广告

2018.02.04 增加3.17获取个人用户信息

2018.02.04 增加3.18根据位置返回商家列表

2018.02.04 增加3.19根据关键词返回绿色商家列表

2018.02.04 增加3.20获取新闻

2018.02.05 启用正式服务器，响应地址修改

## 1关于请求服务器的说明

正式环境 https://api.greencitycoin.cn/v2 (2017年2月5日后可用)

## 2流程

### 2.1用户首次流程

1，短信验证使用三方

2，用户注册 register

3，用户注册后成功后将token保存手机中。

### 2.2非首次流程

1. 本地有token，验证是否可用 checkToken;
2. 不可用,让用户重新登陆

### 2.3忘记密码

1， 从走2.1 测试请求地址 https://api.greencitycoin.cn/v2/forgetPassword

## 3具体api说明

### 3.1注册

#### 3.1.1测试请求地址

https://api.greencitycoin.cn/v2/register

#### 3.1.2参数

参数 手机号 和 密码 和 签名码 的json串

签名码加密使用 防止被盗用 具体加密算法如下：

1. 使用电话号码+”@”+密码

例如：13013505566@1517277703

1. 使用HMAC-SHA1加密，key是Guall0130

例如 HMAC-SHA1("Guall0130", "13013505566@1517277703") = " ea1df8e473d8fe9631bc96d40cba7ab2eed642d1"

这个值请作为测试使用 测试网址http://tool.oschina.net/encrypt?type=2

1. 将上面的编码Base64

ZWExZGY4ZTQ3M2Q4ZmU5NjMxYmM5NmQ0MGNiYTdhYjJlZWQ2NDJkMQ==

这个值也是作为测试验证使用

#### 3.1.3 返回

返回 字符串token

正确的

{

"msg": "Token更新",

"code": 0,

"data": "jxXsYDmbNmexXhBRLgpd8g"

}

错误的：

{

"msg": "该手机号已注册！",

"code": 500

}

### 3.1.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.1.4

### ~~3.2短信验证（接口取消，改为app直接使用3方）~~

#### ~~3.2.1测试请求地址~~

~~https://api.greencitycoin.cn/v2/sendPhoneCode~~

#### ~~3.2.2参数~~

~~参数tel 手机号 和salt~~

#### ~~3.2.3 返回~~

~~success或者失败信息~~

### ~~3.2.4 例子~~

~~请求~~

~~见 PointsApi 测试服务器.postman\_collection.json 的3.2.4~~

~~返回 字符串 success 或者failture~~

### 3.3登陆(用户名密码登录)

#### 3.3.1测试请求地址

https://api.greencitycoin.cn/v2/login

#### 3.3.2参数

用户名和密码的json串

#### 3.3.3 返回

返回 token字符串

成功：

{

"msg": "Token更新",

"code": 0,

"data": "y2HckFdFpeRWD+sat1xMmw"

}

失败：

{

"msg": "用户名或密码错误",

"code": 500

}

### 3.3.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.3.4

### 3.4检查token是否过期(非首次登录)

#### 3.4.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/checkToken

#### 3.4.2参数

参数token令牌字符串

#### 3.4.3 返回

返回

正确信息

{

"msg": "登陆成功",

"code": 0

}

错误信息

{

"msg": "该用户不存在",

"code": 500

}

### 3.4.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.4.4

### 3.5个人信息修改

#### 3.5.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/modityUserInfo

#### 3.5.2参数

参数token令牌字符串

个人信息的json串

#### 3.5.3 返回

success或者失败信息

成功

{

"msg": "更改成功",

"code": 0

}

失败

{

"msg": "用户不存在",

"code": 500

}

### 3.5.4 例子

请求

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.5.4

返回 字符串 success 或者failture

### 3.6密码修改

#### 3.6.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/modityPassword

#### 3.6.2参数

参数token令牌字符串

用户新旧密码的json串

#### 3.6.3 返回

success或者

{

"msg": "更改成功",

"code": 0

}

失败信息

{

"msg": "原密码错误",

"code": 500

}

### 3.6.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.6.4

返回 字符串

### 3.7反馈信息

#### 3.7.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/putFeedback

#### 3.7.2参数

参数token令牌字符串

#### 3.7.3 返回

success或者失败信息

{

"msg": "更改成功",

"code": 0

}

失败：

{

"msg": "反馈信息不存在或反馈字数少于10个字",

"code": 500

}

### 3.7.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.7.4

### 3.8 语言

保存以本地

### 3.9关于

#### 3.9.1测试请求地址

https://api.greencitycoin.cn/v2/getAbout

#### 3.9.2参数

无

#### 3.9.3 返回

字符串信息

返回 字符串

{

"msg": "this is about",

"code": 0

}

### 3.9.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.9.4

### 3.10积分钱包

#### 3.10.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/getPoints?page=0&size=10

#### 3.10.2参数

参数token令牌字符串

#### 3.10.3 返回

Json列表

{

"msg": "积分钱包",

"code": 0,

"data": [

{

"id": 1,

"tel": "13683391111",

"dealerId": 1,

"dealerName": "星巴克",

"dealerLogo": "Starbuckslogo.png",

"memberDeaName": "星巴克的张三",

"balance": 40

},

{

"id": 2,

"tel": "13683391111",

"dealerId": 2,

"dealerName": "嘀嘀打车",

"dealerLogo": "ddicon02.png",

"memberDeaName": "嘀嘀打车的张三",

"balance": 10

}

]

}

### 3.10.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.10.4

返回：

### 3.11捐赠记录

#### 3.11.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/getContributionHistory

#### 3.11.2参数

参数token令牌字符串

#### 3.11.3 返回

Jsoon列表

{

"msg": "捐赠记录",

"code": 0,

"data": [

{

"id": 1,

"dealerId": 1,

"dealerName": "星巴克",

"tel": "13683391111",

"memberDeaName": "星巴克的张三",

"donateNum": 15.52,

"createdDate": 1517474826000

},

{

"id": 2,

"dealerId": 1,

"dealerName": "星巴克",

"tel": "13683391111",

"memberDeaName": "星巴克的张三",

"donateNum": 14.48,

"createdDate": 1517475594000

},

{

"id": 3,

"dealerId": 2,

"dealerName": "嘀嘀打车",

"tel": "13683391111",

"memberDeaName": "嘀嘀打车的张三",

"donateNum": 10,

"createdDate": 1517475645000

}

]

}

### 3.11.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.11.4

返回：

### 3.12绿色地图

#### 3.12.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/getDealers

#### 3.12.2参数

参数token令牌字符串

#### 3.12.3 返回

Jsoon列表

{

"msg": "绿色地图",

"code": 0,

"data": [

{

"id": 1,

"name": "星巴克",

"description": "星巴克（Starbucks）是美国一家连锁咖啡公司的名称，1971年成立，是全球最大的咖啡连锁店，其总部坐落美国华盛顿州西雅图市。星巴克旗下零售产品包括30多款全球顶级的咖啡豆、手工制作的浓缩咖啡和多款咖啡冷热饮料、新鲜美味的各式糕点食品以及丰富多样的咖啡机、咖啡杯等商品。星巴克在全球范围内已经有近21300间分店遍布北美、南美洲、欧洲、中东及太平洋区。",

"logo": "Starbuckslogo.png",

"coordinate": "116.40486522068845,39.93579440797929",

"address": "北京市朝阳区三里屯路19号三里屯太古里F2",

"level": 0,

"link": 0,

"createdDate": 1517562567000,

"del": 0

},

{

"id": 2,

"name": "嘀嘀打车",

"description": "滴滴出行是涵盖出租车、[1] 专车、[2] 快车、[3] 顺风车、[4] 代驾及[5] 大巴等多项业务在内的一站式出行平台，2015年9月9日由“滴滴打车”更名而来。",

"logo": "ddicon02.png",

"coordinate": "116.29401509092246,39.95596035114303",

"address": "海淀区",

"level": 1,

"link": 0,

"createdDate": 1517562573000,

"del": 0

}

]

}

### 3.12.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的3.12.4

返回：

[{"id":1,"name":"啊啊啊","description":"1","logo":"1","coordinate":"1","address":"1","level":1,"link":1,"createdDate":1516341196000,"del":0}]

### 3.13查看反馈信息

#### 3.13.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/getFeedback

#### 3. 13.2参数

参数token令牌字符串

#### 3. 13.3 返回

json列表

{

"msg": "反馈记录",

"code": 0,

"data": [

{

"id": 1,

"memberTel": "13313499588",

"memberName": null,

"memberFeedback": "??????20180101",

"adminHelp": null,

"feedbackDate": 1517627081000,

"helpDate": null,

"createdDate": 1517627081000

},

{

"id": 2,

"memberTel": "13313499588",

"memberName": null,

"memberFeedback": "??????20180101",

"adminHelp": null,

"feedbackDate": 1517627085000,

"helpDate": null,

"createdDate": 1517627085000

}

]

}

错误信息

### 3. 13.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json 的

返回：

### 3.15获取闪屏页广告

#### 3.15.1测试请求地址

https://api.greencitycoin.cn/v2/getPointsScAd

#### 3. 15.2参数

无

#### 3. 15.3 返回

json列表

### 3. 15.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json

返回：

### 3.16积分页广告

#### 3.16.1测试请求地址

https://api.greencitycoin.cn/v2/getPointsPageAd

#### 3. 16.2参数

无

#### 3. 16.3 返回

json列表

### 3. 16.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json

返回：

### 3.17获取个人用户信息

#### 3.17.1测试请求地址

https://api.greencitycoin.cn/v2/{token}/getUserInfo

#### 3. 17.2参数

无

#### 3. 17.3 返回

json列表

### 3. 17.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json

返回：

### 3.18根据位置返回商家列表

#### 3.18.1测试请求地址

https://api.greencitycoin.cn/v2/KwT4Dw5NPIGPhYAR8Wi4Kw/getDealersByAddress?address=16.5333,17.3456&page=0&size=10

#### 3. 18.2参数

Address 位置坐标

Page 页号

Size 每页行数

#### 3. 18.3 返回

json列表

### 3. 18.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json

返回：

### 3.19根据关键词返回绿色商家列表

#### 3.19.1测试请求地址

https://api.greencitycoin.cn/v2/KwT4Dw5NPIGPhYAR8Wi4Kw/getDealersByName? name =星巴克address=16.5333,17.3456&page=0&size=10

#### 3. 19.2参数

Name 商家相关的关键词

Address 位置坐标

Page 页号

Size 每页行数

#### 3. 19.3 返回

json列表

### 3. 19.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json

返回：

### 3.20获取新闻

#### 3.20.1测试请求地址

https://api.greencitycoin.cn/v2/KwT4Dw5NPIGPhYAR8Wi4Kw/getNews?page=0&size=10

#### 3. 20.2参数

Token登陆令牌

Page 页号

Size 每页行数

#### 3. 20.3 返回

json列表

### 3. 20.4 例子

请求

见 PointsApi 测试服务器.postman\_collection.json

返回：