

# RealFriend API 文档

---

- acmore
- 2018.5.31

## 0. API

---

### 0.1 API列表

直接访问[后端地址](#)就能看到后端的API列表，这份列表是基于 `django-rest-framework` 自动生成的，基本包含了所有的API，部分API未被检测到，也会在下边单独列出。

**对于访问的任何API接口，都要在其后加上 `?format=json` 这样才能得到Json风格的返回值。**

下边的文档中任何接口的请求都有返回值，其中在[后端地址](#)能看到的接口地址都是固定的，在浏览器访问试一下就能清楚。对于[后端地址](#)上没有的，我会单独写明。

### 0.2 如何访问

关于Rest风格的接口，基本操作有四种：

- get：获取操作
- update/patch：更新操作
- post/put：新增操作
- delete：删除操作

反映在具体的编程中，patch和put都可以用post完成，一些比较现代化的网络请求工具会直接提供这四种甚至更多的方法。

### 0.3 关于参数

在一般的网络请求中，url定位资源有几种形式，分别如下：

- 方法请求
  - 即使是同样的路径，使用不同的方法请求可能会有不同的效果，如对同一个用户使用GET和DELETE方法显然结果是不同的
- 路径请求
  - 可以把参数隐含在url路径中，如 `/user/acmore`，这里的 `acmore` 就是一个变量，可以变换其内容来获取不同的用户信息，相应的 `acmore` 就叫路径参数，在下边的API文档中，需要路径参数的会用 `{}` 标明，如 `/user/{username}`

- 参数请求
  - 这是最经典和常用的一种请求方式，对于不同的方法有不同的参数形式，分别如下：
    - GET：直接在url后附上 `?key1=value1&key2=value2...` 的请求参数
    - POST：在请求url的时候使用附加数据将json风格的键值对发送给服务器
  - 以上两种形式区别之一在于url的长度是有限制的，所以GET参数不能过长，而POST就没有这个限制

0.4 易混淆点

- `username` 和 `nickname` 不一样，前者相当于微信号，唯一且固定，后者相当于微信昵称，随时可变

1. User

名称	地址	方法
获取用户列表	<code>/user</code>	GET
获取用户信息	<code>/user/{username}</code>	GET/PUT
获取用户关注的用户	<code>/user/{username}/followees</code>	GET
获取用户的粉丝	<code>/user/{username}/followers</code>	GET
获取用户和另一位用户的友好度	<code>/user/{username}/friendship/{friend_username}</code>	GET/PUT
创建新用户	<code>/user/create</code>	PUT
登录验证 (后端地址没有)	TODO	TODO
关注好友 (后端地址没有)	<code>/user/{username}/follow/{friend_username}</code>	不限，最好是POST。 。如果成功返回值为 <code>"status": "success"</code> 否则直接异常

2. Activity

名称	地址	方法	参数
获取活动列表	<code>/activity</code>	GET	无

获取活动详情	/activity/{activity_id}	GET/UPDATE	GET:无/UPDATE:具体的
创建新的活动	/activity/create	PUT	具体参数见 <a href="#">后端地址</a> 其中的 create_time 这 后端会自动生 其中 participants 是上 表
发表评论	/activity/{activity_id}/comment	POST	<pre>{   nickname: 用   username: 月   content: 评 }</pre> 返回值是 {"status": "comment_id": 新
获取评论列表	/activity/{activity_id}/comments	GET	无。返回值是一 列表的每个元素是一个 其内容 <pre>{   id: 评论   nickname: 评论者   username: 评论者   content: 评论者   create_at: 评论 }</pre>

### 3. Game

名称	地址	方法	参数	说明
获取游戏列表	/game	GET	无	获取所有的游 按照时间排序，首
获取游戏详情	/game/{game_id}	GET/UPDATE	GET:无/UPDATE: 具体的内容见 <a href="#">后端地址</a>	获取游戏详 并且可以更新游
创建新的游戏	/game/create	PUT	具体参数见 <a href="#">后端地址</a>	通过这个接口来创

### 4. Message

名称	地址	方法	参数	
			?username=	参

获取消息列表	/message	GET	{username}	
获取消息详情	/message/{message_id}	GET/UPDATE	GET:无/UPDATE: 具体的内容见 <a href="#">后端地址</a>	
创建新的消息	/message/create	PUT	具体参数见 <a href="#">后端地址</a>	
将消息标记为已读	/message/read/{message_id}	GET	无。 返回值是 {"status": "success"}	过

## 5. Merchant

名称	地址	方法	参数	
获取商家列表	/merchant	GET	无	获取
获取商家详情	/merchant/{merchant_id}	GET/UPDATE	GET:无/UPDATE: 具体的内容见 <a href="#">后端地址</a>	获 并且可
创建新的商家	/merchant/create	PUT	具体参数见 <a href="#">后端地址</a>	通过这个

## 6. Token

名称	地址	方法	参数	说明
获取身份令牌 (可用, 但是目前不需要用)	/token	POST	<pre>{   username :   权限用户的用户名,   password :   权限用户的密码 }</pre> 返回值是 <pre>{   status :   "success",   token :   有效的令牌 }</pre>	将权限用户的登录口令POST上来之后 然后请求消息的时候加在消息头中, (目前没有在这里设限 所以不需要获取令牌就能直