## 后台接口文档

Auther：Harry Zhu Version：v2.0 Date： July 16，2015

### 图片上传操作逻辑

前端访问 后端接口/api/upload\_token，提交参数 entity\_id, 后端返回 token+img\_version 作为图片地址 前端根据七牛的上传规则，上传相应图片，

image\_id = fdbaf9d61f8148c38e62104fc9f1adda\_profile\_image\_1 比如：<http://qiankr.qiniudn.com/fdbaf9d61f8148c38e62104fc9f1adda_profile_image_1>

前端持有相应的图片地址 POST到 /api/user 或者 /api/album 或者 /api/event

在登录条件下，POST请求 /api/user 或者 /api/album 或者 /api/event 1. 用户 /api/user

主体：用户 客体：用户资料 行为：上传头像、背景图 image\_id

1. 专辑 /api/album 主体： 用户 客体： 专辑 行为：上传若干照片 image\_id
2. 任务 /api/event 主体：用户 客体：任务 行为：上传一张图片 image\_id

### 一、用户验证模块

1. 验证码服务：/verify\_phone/

请求：POST

参数：

1. phone （前端验证手机号码有效性） #手机号

返回状态：

1. 成功发送，该用户尚未使用该系统，且尚未注册
2. 成功发送，该用户曾经使用该系统，但尚未注册
3. 成功发送，该用户已经注册，该用户忘记密码 （forget\_password)
4. 拒绝发送，该用户已经注册，不能重复注册（signup)
5. 当日发送数量达到上限
6. 手机注册接口：/signup/

请求：POST

参数：

1. phone #手机号
2. password1 #密码1
3. password2 #密码2
4. verified\_code #验证码
5. current\_page {signup} #当前页面类型（hidden） 返回状态：
6. 成功注册：
7. 注册失败，用户已经注册：
8. 密码不一致或（密码违规，前端检验）
9. 验证码不正确
10. 忘记密码再登陆：/forget\_password/ 请求：POST 参数：
11. phone #手机号
12. verified\_code #验证码
13. password0 #密码0 \*\*
14. password1 #密码1 \*\*
15. current\_page {forget\_password} #当前页面类型 （hidden）

返回状态： 1. 找回密码成功，返回新密码 2. 验证码错误 3. 手机未注册

1. 微信注册登陆：/auth/wechat/ 请求：POST 参数：
2. 微博注册登陆: /auth/weibo/ 请求：POST 参数：
3. 手机登陆接口:/login/ 请求：POST 参数：
4. phone #手机号
5. password #密码

返回： 1. 登陆成功 2. 登陆失败，手机号或密码不正确，请重新输入，忘记请找回。

1. 登出接口:/logout/ 请求：GET

### 二、推荐系统模块 ### cookie判断当前用户是否登陆,除专辑推荐外其他均需要登陆

* 默认推荐模式两种：

1. 按照发布时间（自动）
2. 按照编辑排序（手动）//未来可能引入推荐算法
3. 推荐专辑列表:/albums/

请求：GET

参数：

1. current\_page : albums(hidden) #当前页面类型

返回值：

1. album\_id(type==album) #专辑实体ID
2. album\_owner\_id #摄影师ID
3. album\_publish\_datetime #发布日期、时间
4. album\_name #专辑名称
5. album\_liked\_number #点赞数量
6. album\_forwarded\_number #转发数量
7. album\_commented\_number #评论数量
8. album\_face #专辑封面
9. album\_images #专辑其余图片
10. album\_rank #专辑展示排序
11. album\_type(default:unclassified)#专辑类型
12. 推荐用户列表:/users/

请求：GET

参数：

1. current\_page : users(hidden) #当前页面类型

返回值：

1. user\_id #摄影师实体ID
2. user\_name #摄影师名称
3. user\_bio #摄影师简介
4. user\_face #摄影师头像
5. user\_title #摄影师评级（默认：千刻注册摄影师）
6. user\_rank #摄影师排序
7. user\_type(default:unclassified) #摄影师类型
8. 推荐活动列表:/events/

请求：GET

参数：

1. current\_page : events(hidden) #当前页面类型

返回值： 1. event\_id #活动实体ID 2. event\_name #活动名称 3. event\_face #活动封面 4. event\_rank #活动排序 5. event\_type(topic,match,party) #活动类型 (此处需要征求产品经理)

### 三、详情模块 \*\*\* 需要用户登录 \*\*\*

1. 大图模式:/check\_big\_image/{image\_id} [七牛云服务] 前端：下载图片+举报功能

请求：GET

参数：image\_id

返回值：

1. image\_id #图像地址
2. image\_owner\_id #所有者
3. image\_status(legal,illegal,checking)#是否合法：合法、不合法、审核中
4. 专辑详情：/album/{entity\_id}

请求：GET

参数：entity\_id

返回值：

1. album\_name #专辑名称
2. album\_face #专辑封面
3. album\_images #专辑其余图片
4. album\_introduction #专辑简介
5. album\_tag #专辑标签
6. album\_commented\_number #专辑评论总数
7. 活动详情:/event/{entity\_id}

请求：GET

参数：entity\_id

返回值： 1. event\_background #活动背景图 2. event\_name #活动名称 3. event\_introduction #活动介绍 4. event\_album #用户专辑 5. event\_album\_order #用户专辑排序 5. event\_joined\_number #参与人数 6. event\_album\_number #活动总上传专辑数（允许一张专辑只包含一张图片） 7. event\_expired\_datetime #活动截止日期时间 8. event\_begin\_datetime #活动开始日期时间

1. 用户详情:/user/{entity\_id}

请求： GET

参数：entity\_id

返回值：

1. user\_name #摄影师名称
2. user\_face #摄影师头像
3. user\_photo\_number #摄影师总图片数量
4. user\_followed\_number #摄影师总粉丝数
5. user\_following\_number #摄影师总关注数
6. user\_backgroud #摄影师资料背景图
7. album\_face #各专辑封面
8. album\_name #各专辑名称
9. album\_photo\_number #各专辑图片总数

### 四、产出内容模块 \*\*\* 需要用户登录 \*\*\* 单张专辑一次最多9张图片

1. 图片上传:/api/upload\_token/

请求：POST 参数： 1. entity\_id(type==album|event|user) #所有者类型ID

返回值： 1. upload\_token #七牛token 2. img\_version #图片版本序列

1. 专辑发布:/album\_create/

请求：POST 参数：

1. user\_id #用户ID \*\*
2. key #关键字
3. value #关键字的值

key-value的允许值：

1. album\_id(type==album) #专辑ID
2. album\_owner #专辑所有者ID
3. album\_introduction #专辑介绍
4. album\_tag #专辑标签
5. album\_publish\_datetime #专辑发布时间
6. album\_face #专辑封面
7. album\_images #专辑图片
8. 专辑编辑:/api/album

请求：POST 参数： 1. album\_id #专辑ID \*\* 2. key #关键字 3. value #关键字的值

key-value的允许值：

1. album\_images\_new #专辑照片新序列
2. album\_bio #编辑专辑简介
3. album\_name #编辑专辑名称
4. album\_face #编辑封面
5. 个人信息编辑：/api/user/

请求：POST 参数：

1. user\_id #用户ID
2. key #关键字
3. value #关键字的值 key-value的允许值
4. user\_name #用户名
5. user\_bio #个人介绍
6. user\_face #个人头像
7. user\_backgroud #背景图片
8. 任务提交：/api/events

请求：POST 参数：

1. user\_id #用户ID
2. key #关键字
3. value #关键字的值

key-value的允许值

1. event\_id #任务ID
2. event\_image #任务图片
3. event\_image\_datetime #任务图片上传时间

### 五、社交消息中心模块

1. 专辑点赞：/api/like/

请求：POST 参数： 1. user\_id #点赞用户ID 2. album\_id #被点赞专辑ID

1. 取消专辑点赞:/api/unlike/

请求：POST 参数： 1. user\_id #取消点赞用户ID 2. album\_id #取消点赞专辑ID

1. 活动参与:/api/join/

请求：POST 参数： 1. partner\_id #参与用户ID \*\* 2. event\_id #活动ID

1. 取消活动参与:/api/unjoin/

请求：POST 参数： 1. partner\_id #取消参与用户ID \*\* 2. event\_id #活动ID

1. 关注其他摄影师:/api/follow/

请求：POST 参数： 1. followed\_id #受关注的摄影师ID 2. following\_id #围观群众ID

1. 取消关注其他摄影师:/api/unfollow/

请求：POST 参数： 1.unfollowed\_id #受冷落的摄影师ID 2.unfollowing\_id #围观群众ID

1. 专辑转发到社交网络:/api/share/ ***朋友圈|微信好友|微博|其他***

请求：POST 参数：

1. 举报非法专辑:/api/report

请求：POST

参数： 1. reporter #投诉人ID （entity\_id(type==user)） 2. suspect #嫌疑人ID （entity\_id(type==user)） 3. image\_id #嫌疑照片ID 4. report\_reason #举报理由：非法图片、版权争议 5. report\_datetime #举报日期时间

返回值： 1. report\_id #产生的举报ID 2. reprot\_status #举报成功或失败

1. 投诉建议:/api/complaint

请求：POST

参数： 1. complainant\_id #当前投诉人ID 2. complaint\_content #投诉内容 3. complaint\_datetime #投诉时间

返回值： 1. complaint\_id #产生的投诉ID 2. complaint\_status #投诉成功或失败

1. 消息通知推送:/api/message

请求：GET 参数： 1. message\_type #推送内容类型{摄影获奖信息、举报|投诉反馈信息、新点赞|关注数量信息、后台编辑自定义} 2. message\_name #推送信息名称 3. message\_content #推送信息内容 4. message\_template #推送信息模板

推送举例如下： 1. 您的专辑 {{album\_name}}又获得 {{ new\_like\_number }} 个新的点赞; 2. 感谢您的举报，举报编号为{{ report\_id }}，您的举报信息正在受理中，请您耐心等待; 3. 恭喜您获得{{ event\_name }} 大赛的一等奖，奖品正在派送中！ 4. 您的专辑 {{album\_name}}有 {{ new\_comment\_number }} 个新的评论; 5. 感谢您的投诉，投诉编号为{{ complaint\_id }}，您的投诉建议正在受理中; 6. {{ event\_name}} 还有{{event\_expired\_datetime}}即将结束，快来参与赢取精美奖品吧！ 7. 已成功举报{{ report\_id }},图片/专辑{{ image\_name|album\_name }}已经成功下架！ 7. 很可惜，您的举报{{ report\_id }}失败，如仍需举报，请提供更多信息，谢谢合作！