

# 服务器域名:moran.chinacloudapp.cn。

请求的 URL 格式为[域名]/moran/web/[controller]/[action]。下面讲到具体 API 时只写 [controller]/[action]。比如登录 API 的格式给出的是/user/login,那么实际使用时请求的地址就是

moran.chinacloudapp.cn/moran/web/user/login

## RESTful API

使用 token 进行验证。每次登录成功后,服务器返回的内容中会包含一个 token,调用其它 API 时需传入这个 token 进行验证。若 token 通过不了验证,则返回提示“Invalid token”。

返回结果为 json 格式。有些 API 可传 gbid 参数,用于标识用户是哪个项目组的。gbid 的取值可以是项目组的编号,没有项目编号的同学可以用学号标识。

## 注册

/user/register

### 提交格式:

提交方式:POST。

提交参数:(该文档除却图片数据都应提交NSString类型)

username (参考示例: username = testProject3)

password (参考示例: password = 123456789)

email (参考示例: email = textProject@163.com )

gbid (参考示例: gbid = GeekBand-I150003 )

其中gbid 为各个学员的学号。

## 返回格式:

若请求成功: 应返回类似的如下信息。

```
{  
  
  "status": 1,  
  "data": {  
    "user_id": "6",  
    "user_name": "testProject3",  
    "project_id": "3"  
  },  
  "message": "Register success"  
}
```

若失败,返回相关提示。错误提示信息计有:

- 若缺失参数,提示"Missing params"
- 若密码长度不在 6 到 20 的范围内,提示"Password length must in be [6, 20]"
- 若该 email 已用于注册,提示"Email exists"

# 登录

/user/login

POST 提交 email 与 password,以及用于标注是哪个项目的参数 gbid。

## 提交格式:

提交方式:POST。

提交参数:

email (参考示例: email = textProject@163.com )

password (参考示例: password = 123456789)

gbid (参考示例: gbid = GeekBand-l150003 )

其中gbid 为各个学员的学号。

## 返回格式:

若请求成功: 应返回类似的如下信息。

```
{  
  "status": 1,  
  "data": {  
    "user_id": "2",  
    "user_name": "testProject3",  
    "token": "cec5a0a5e7a1951c87099b96d16851bf6f9f0a21",  
    "avatar": "",  
    "project_id": "1",  
    "last_login": "2015-08-30 17:35:49",
```

```
"login_times": 3
},
"message": "Login success"
}
```

**重要的返回说明：**token为登陆成功获取的令牌。（在后面的接口中需频繁使用）

**user\_id** 为登陆成功后返回的用户Id （在后面的接口中需频繁使用）

若失败,返回相关提示。错误提示信息计有:

- 若缺失参数,提示"Missing params"
- 若此邮箱未注册,提示"No such user" • 若密码不正确,提示“Wrong password”

## 上传头像

/user/avatar

### 提交格式:

提交方式:POST。

提交参数:

user\_id           (参考示例: user\_id = 3 )

token

(参考示例: token = "092f9d43fbd0a02c29c4963187ab2f1b31067f55" )

data               (参考示例: data 为头像的图片数据, **建议上传前压缩**)

其中gbid 为各个学员的学号。

### 返回格式:

若请求成功：应返回类似的如下信息。

**receive data string:**

```
"status": 1,
"data": {
  "user_id": "2",
  "user_name": "testProject3",
  "email": "textProject@163.com",
  "avatar": "VvarVwwwVhtmlVmoranVcontrollersVavatarV
39_1445220445.jpg"
},
"message": "Update success"
}
```

## 更改用户名

/user/rename

### 提交格式:

提交方式:POST。

提交参数:

user\_id (参考示例: user\_id = 3 )

token

(参考示例: token = "092f9d43fbd0a02c29c4963187ab2f1b31067f55" )

new\_name (参考示例: new\_name = 大白菜 )

参数说明: token 为登陆成功后获得的令牌。 user\_id 为登陆成功后返回的Id。

new\_name为新名称。

## 返回格式:

若请求成功：应返回类似的如下信息。

```
ReName data string:{
  "status": 1,
  "data": {
    "user_id": "2",
    "user_name": "testProject3",
    "email": "testProject3@163.com",
    "avatar": "VvarVwwwVhtmlVmoranVcontrollersVavatarV
39_1445220445.jpg"
  },
  "message": "Update success"
}
```

## 显示头像

/user/show

GET 提交 user\_id。

## 获取评论列表

/comment

GET 获取。传 user\_id,token;可选参数 pic\_id(地点与图片是 1:m 的关系,图片与评论也是 1:n 的关系),若不传则返回所有节点的评论;可选参数 page,页码,可选参数 limit,每页条数,这两个参数用于控制分页,默认分别为 1 和 10,即默认显示头 10 条。

## 查看评论

/comment/view

GET。传 user\_id,token,comment\_id,评论列表中有。

## 创建评论

/comment/create

POST 提交。参数 user\_id,token,pic\_id,comment(评论的文字内容)。

## 编辑评论

/comment/update

POST 提交。参数 user\_id,token,comment\_id,comment(评论的文字内容)。

## 删除评论

/comment/delete

POST 提交。参数 user\_id,token,comment\_id。

## 获取地理位置列表

/node/list

### 提交格式:

提交方式:GET。

提交参数:

token

(参考示例: token = "092f9d43fbd0a02c29c4963187ab2f1b31067f55" )

user\_id (参考示例: user\_id = 3)

distance (参考示例: distance = 500)

longitude (参考示例: longitude = 37.1289327 )

latitude (参考示例: latitude = 54.213791)

## 返回格式:

若请求成功: 应返回类似的如下信息。

**Square receive data string:{**

**“status”:1,**

**“data”:{“6”:{**

**“node”:{“id”:"6",**

**“distance\_in\_meters”:"0",**

**"addr": "\u6dee\u6d77\u4e2d\u8def138\u53f7\u4e0a\u6d77\u5e7f\u573a4\u697c401\u53f7\u94fa", "ST\_AsText(geom)": "POINT(121.47794 31.22516)"},**

**“pic”:[{“pic\_id”:"51", "node\_id": "6", "user\_id": "0", "title": "", "pic\_link": "http://moran.chinacloudapp.cn/moran/web/picture/read?pic\_id=51"},  
{"pic\_id": "52", "node\_id": "6", "user\_id": "0", "title": "", "pic\_link": "http://moran.chinacloudapp.cn/moran/web/picture/read?pic\_id=52"},  
{"pic\_id": "53", "node\_id": "6", "user\_id": "0", "title": "", "pic\_link": "http://moran.chinacloudapp.cn/moran/web/picture/read?pic\_id=53"},**

**----- 省略大量图片信息 -----**

**]**

**}},**

**"0":256}**



# 上传图片

/picture/create

## 提交格式:

提交方式:POST。

提交参数:

token

(参考示例: token = "092f9d43fbd0a02c29c4963187ab2f1b31067f55" )

user\_id (参考示例: user\_id = 3)

data (参考示例: data为图片数据, **建议上传前压缩**)

title (参考示例: title = 这家店真不错 )

location (参考示例: location= @“”)

longitude (参考示例: longitude = 37.3823791238123 )

latitude (参考示例: latitude = 56.123984287123 )

addr (参考示例: addr = 寿宁路98号6楼)

## 返回格式:

若请求成功: 应返回类似的如下信息。

```
receive data string:{
  "status": 1,
  "data": {
    "node_id": "449",
    "pic_id": "4182",
    "title": "\u4f60\u60f3\u8bf4\u7684\u8bdd"
  }
}
```

## 访问图片

/picture/read

GET。传入参数 user\_id,token,pic\_id。展示图片。

## 用户列表 (非学员接口)

/users

GET 获取。返回所有用户的 id,name,email,avatar,以及结果分页相关信息。返回示

例:

```
{
  "items": [
    {
      "id": 1,
      "name": "diaosj",
      "password": "ee03249b4e0b9e6c624d42f5d18cc9d",
      "email": "752156102@qq.com",
      "project_id": 0,
      "avatar": "",
      "created": "2015-08-20 02:29:35",
      "modified": "2015-08-20 02:29:37"
    },
    {
      "id": 2,
      "name": "testProject1",
      "password": "4ed56252edc00aa0c4ef893eff99e694",
      "email": "testProject1@test.com",
      "project_id": 1,
      "avatar": "",
      "created": "2015-08-20 02:29:37"
    }
  ]
}
```

```

": "2015-08-20 03:17:14", "modified": "2015-08-20
03:17:16"}, {"id":
3, "name": "testProject2", "password": "d253576dc8e6d54712878dd9
1beace4f", "email": "testProject2@test.com", "project_id":
2, "avatar": "", "created

": "2015-08-20 06:37:10", "modified": "2015-08-20
06:37:12"}], "_links": {"self": {"href": "http://moran.chinacloudapp.cn/
moran/web

/users?page=1"}}, "_meta": {"totalCount": 3, "pageCount": 1, "currentPage":
1, "perPa

ge": 20}}

```

可在请求中指定返回的字段。如/users?fields=id,email。返回示例:

## 用户详情 (非学员接口)

/users/

```

{"items": [{"id": 1, "email": "752156102@qq.com"}, {"id":
2, "email": "testProject1@t

est.com"}, {"id": 3, "email": "testProject2@test.com"}], "_links": {"self":
{"href":

"http://moran.chinacloudapp.cn/moran/web/users?fields=id
%2Cemail&page=1"}}, "_

meta": {"totalCount": 3, "pageCount": 1, "currentPage": 1, "perPage": 20}}

```

这里 id 是指 user 表的 id。返回的结果类似用户列表。