

# 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

提交方式: POST。提交参数: username, password与email, 以及gbid。比如提交的表单数据为, username=testProject3, password=testProject3, gbid=GeekBand-I150003, 则返回示例:

```
{
  "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。返回内容示例:

```
{
  "status": 1,
  "data": {
    "user_id": "2",
    "user_name": "testProjectNew1",
    "token": "cec5a0a5e7a1951c87099b96d16851bf6f9f0a21",
    "avatar": "",
    "project_id": "1",
    "last_login": "2015-08-30 17:35:49",
    "login_times": 3
  },
  "message": "Login 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"
- 若此邮箱未注册，提示"No such user"
- 若密码不正确，提示"Wrong password"

## 用户列表

/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": "ee03249b4e0b9e6c624d42f5d18cc9db",
      "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 03:17:14",
      "modified": "2015-08-20 03:17:16"
    },
    {
      "id": 3,
      "name": "testProject2",
      "password": "d253576dc8e6d54712878dd91beace4f",
      "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,
    "perPage": 20
  }
}
```

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

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

## 用户详情

/users/

这里id是指user表的id。返回的结果类似用户列表。

## 上传头像

/user/avatar

POST提交。传递参数：user\_id, token, data（图片文件）。

## 更改用户名

/user/rename

POST提交。传递参数：user\_id, token, new\_name。返回示例：

```
{
  "status": 1,
  "data": {
    "user_id": "2",
    "user_name": "testProject1",
    "email": "testProject1@test.com",
    "avatar": ""
  },
  "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

GET提交。参数user\_id, token, longitude, latitude, distance。返回地理位置的列表，以及各自的图片url列表。按距离排序。

## 上传图片

/picture/create

POST提交。参数计有：user\_id, token, longitude, latitude, data（图片数据）。上传后返回的data中会有pic\_id，可供访问图片。

## 访问图片

/picture/read

GET。传入参数user\_id, token, pic\_id。展示图片。