

# 接口文档

接口地址：

1. 生产环境： <http://47.92.164.250:3000>
2. 开发环境： <http://localhost:3000>

接口响应格式：

```
{
  code, data, msg
}
```

code: 状态码

data: 数据体

msg: 响应信息

---

## Auth相关（用户、权限）

---

Base-prefix: /auth

用户Schema：

```
const UserSchema = new mongoose.Schema({
  username: String, // 用户名
  password_sha: String, // 密码sha
  nickname: String, // 昵称
  description: String, // 描述（个人简介）
  headImage: String, // 头像
  phone: String, // 手机号
  sex: Number, // 性别 0女 1男
  birth_time: Number // 出生时间 时间戳
})
```

### 注册

注册用户，可设置用户名，密码。

Request：

参数	值	属性
Url	/register	
Params	username	string required
	password	string required
	nickname	string 不传自动设置为“用户-n”
Method	Put	

Response：

状态码	描述
200	成功
208	用户已存在

## 删除用户（单个）

根据用户名单独的删除某个用户

Request：

参数	值	属性
Url	/delete	
Params	username	string required
Method	Post	

Response：

状态码	描述
200	成功
404	用户不存在

## 登陆

登陆用户，传入用户名和密码。

Request：

参数	值	属性
Url	/login	
Params	username	string required
	password	string required
Method	Post	

Response：

状态码	描述
200	成功
404	用户不存在
401	密码错误

## 更新用户信息

更新用户基本信息，如昵称、描述等。

Request：

参数	值	属性
Url	/userinfo/update	
Params	username	string required
	info	json-object-string required
Method	Patch	

Response：

状态码	描述
200	成功
404	用户不存在
401	参数格式错误

## 获取用户信息

获取用户全部信息。

Request：

参数	值	属性
Url	/userinfo	
Params	username	string required
Method	Get	

Response：

状态码	描述
200	成功
404	用户不存在

## 修改用户密码

修改用户密码，需传入当前密码与新密码。

Request：

参数	值	属性
Url	/password/update	
Params	username	string required
	password	string required
	newPassword	string required
Method	Patch	

Response：

状态码	描述
200	成功
404	用户不存在
401	密码不正确

## 更改用户头像

上传文件的方式来修改头像。

Request：

参数	值	属性
Url	/userinfo/headimage/update	
Params	username	string required
	file	file required
Method	Post	

Response:

状态码	描述
200	成功
404	用户不存在

## Category相关（一级、二级分类、标签）

Base-prefix: /category

Schema:

```
// 一级分类
const ClassifySchema = new mongoose.Schema({
  title: String,
  groups: Array<GroupSchema>, // { title, id }
  hots: Number
})

// 二级分类
const GroupSchema = new mongoose.Schema({
  title: String,
  classify_id: String,
  hots: Number
})

// 标签
const TagSchema = new mongoose.Schema({
  title: String,
  hots: Number
})
```

### 获取全部一级分类

Request:

参数	值	属性
Url	/classify/list	
Method	Get	

Response:

状态码	描述
200	成功
404	classify不存在

## 获取某个一级分类

Request:

参数	值	属性
Url	/classify	
Params	id	classify_id
Method	Get	

Response:

状态码	描述
200	成功
404	classify不存在

## 删除某个一级分类

Request:

参数	值	属性
Url	/classify/delete	
Params	id	classify_id
Method	POST	

Response:

状态码	描述
200	成功
404	classify不存在

### 新增某个一级分类

Request:

参数	值	属性
Url	/classify/create	
Params	title	标题
Method	PUT	

Response:

状态码	描述
200	成功
208	classify已存在

### 新增一个二级分类

Request:

参数	值	属性
Url	/group/create	
Params	title	标题
Params	classifyId	所属一级分类ID
Method	PUT	

Response:

状态码	描述
200	成功
404	classify不存在
208	classify中此group已存在

### 删除一个二级分类

Request:

参数	值	属性
Url	/group/delete	
Params	groupId	
Params	classifyId	所属一级分类ID
Method	POST	

Response:

状态码	描述
200	成功
404	classify不存在
404	classify中此group不存在

新增一个标签

Request:

参数	值	属性
Url	/tag/create	
Params	title	
Method	PUT	

Response:

状态码	描述
200	成功
208	tag已存在

删除一个标签

Request:



参数	值	属性
Url	/tag/delete	
Params	id	
Method	POST	

Response：

状态码	描述
200	成功
404	tag不存在

## File相关（上传、下载）

Base-prefix: /file

### 上传图片（通用）

上传一张照片

Request：

参数	值	属性
Url	/image/upload	
Form-data	file	file required
Method	Post	

Response：

状态码	描述
200	成功
data	{ imageUrl: ' <a href="http://localhost:3000/uploads/">http://localhost:3000/uploads/</a> ' + filename }

## 有趣的接口

Base-prefix: /funny

### 随机获取一碗毒鸡汤

随机的获取一碗毒鸡汤，数据来源于github，且处理后已经保存在mongodb中了。

毒鸡汤数据Schema:

```
const SoupSchema = new mongoose.Schema({
  title: String,
  hots: Number
})
```

Request:

参数	值	属性
Url	/soup/random	
Method	Get	

Response:

状态码	描述
200	成功
data	{ title: '毒鸡汤内容', hots: '热度' }

### 有道翻译

中英文双向翻译接口，调用了有道提供的翻译公共api

Request:

参数	值	属性
Url	/translate	
Params	text	String Required
Method	Get	

Response:

状态码	描述
200	成功

Response Success Data:

```
{
  "code": 200,
  "data": {
    "type": "EN2ZH_CN",
    "errorCode": 0,
    "elapsedTime": 1,
    "translateResult": [
      [
        {
          "src": "i love u",
          "tgt": "我爱你"
        }
      ]
    ],
  },
  "msg": "success"
}
```