**1、 用户接口**

**1.登录**

**/user/login.do** post(代码需要 post 方式请求),开放 get，方便调试request

username,password

response

fail

{

"status": 1,

"msg": "密码错误"

} success {

"status": 0, "data": {

"id": 12, "username": "aaa",

"createTime": 1479048325000, "updateTime": 1479048325000

}

}

**2.注册**

/user/register.do post(代码需要 post 方式请求)

request username,password,

success

{

"status": 0,

"msg": "校验成功"

} fail {

"status": 1,

"msg": "用户已存在"

}

**3.检查用户名是否有效**

/user/check\_valid.do post(代码需要 post 方式请求),开放 get，方便调试

/check\_valid.do?str=admin&type=username 就是检查用户名。request

str,type

str 可以是用户名也可以是 email。对应的 type 是 username 和 email

response

success

{

"status": 0,

"msg": "校验成功"

}

fail

{

"status": 1,

"msg": "用户已存在"

}

**4.获取登录用户信息**

/user/get\_user\_info.do post(代码需要 post 方式请求)

request无参数response success

{

"status": 0, "data": {

"id": 12, "username": "aaa",

"createTime": 1479048325000,

"updateTime": 1479048325000

}

} fail {

"status": 1,

"msg": "用户未登录,无法获取当前用户信息"

}

**5.修改个人信息**

//user/update\_user\_info.do post(代码需要 post 方式请求)

Request

user\_id avatar

description

response success

{

"status": 0, "data": {

“avatar”:xx

“description”:ssss

......

}

}

fail {

"status": 1,

"msg": "修改信息失败"

}

**2、 文章接口**

**1.展示文章列表**

//article/list.do post(代码需要 post 方式请求)

Request

user\_id

response success

{

"status": 0, "data": {

{

“article\_id: ”

“title: ”,

”description: ”,

"createTime": ,"

}

{

“article\_id: ”

“title: ”,

”description: ”,

"createTime": ,"

}

{

“article\_id: ”

“title: ”,

”description: ”,

"createTime": ,"

}

......

}

fail {

"status": 1,

"msg": "获取列表信息失败"

}

**2.返回文章在文件服务器的路径**

//article/article\_src.do post(代码需要 post 方式请求)

Request

article\_idresponse success

{

"status": 0, "data": {

“src: ”

}

fail {

"status": 1,

"msg": "获取路径失败"

}

**3、 管理员接口**

**1.获取用户列表**

//manager/get\_users.do post(代码需要 post 方式请求)

Request

无参数response success

{

"status": 0, "data": {

{

“user\_id: ”

“user\_name: ”

“user\_avatar: ”

},

{

“user\_id: ”

“user\_name: ”

“user\_avatar: ”

},

{

“user\_id: ”

“user\_name: ”

“user\_avatar: ”

}

......

}

fail {

"status": 1,

"msg": "获取用户列表失败"

}

**2.待审核列表**

//manager/audit.do post(代码需要 post 方式请求)

Request

无参数response success

{

"status": 0, "data": {

{

“article\_id: ”

“title: ”

“author: ”

}

}

fail {

"status": 1,

"msg": "获取待审核文章列表失败"

}

**4、 评论接口**

**1.发表评论**

//comment/release.do post(代码需要 post 方式请求)

Request

Author(评论人) content article\_id

response success

{

"status": 0,

"msg": "发表评论成功"

fail {

"status": 1,

"msg": "发表评论失败"

}

**2.获取评论**

//comment/get\_comment.do post(代码需要 post 方式请求)

Request

article\_id

response success

{

"status": 0, "data": {

{

“comment: ”

“article: ”

“author: ”（评论人）

“time: ”

}

}

fail {

"status": 1,

"msg": "获取待审核文章列表失败"

}