

接口定义

1. 用户管理

URL	方法	描述	请求头	请求体	备注	响应头	响应体
/user/register	POST	注册		{ "id":string, fakename":string, "password":string}	id因为 有唯一性故 作为电话或邮箱,		{ "code": "0 (成功) /1 (失败) ", "message":string, "data":string}
/user/login	POST	登录		{ "id": string, "password":string }			{ "code": "0 (成功) /1 (失败) ", "message":string, "data":string JWT令牌}
/user/userinfo	GET	获取用户信息	'Authorization': (string)JWT令牌	无	JWT令牌一般12个小时过期		{ "code": "0 (成功) /1 (失败) ", "message":string, "data":object}
/user/update	PUT	更新用户信息	'Authorization': (string)JWT令牌	{ "id":string, fakename":string, "password":string}			{ "code": "0 (成功) /1 (失败) ", "message":string, "data":string}
/user/updateAvatar	PATCH	更新头像	'Authorization': (string)JWT令牌	{ "avatarUrl":string}			{ "code": "0 (成功) /1 (失败) ", "message":string, "data":string}
/user/randomupdate	POST	随机更新头像	'Authorization': (string)JWT令牌				{ "code": "0 (成功) /1 (失败) ", "message":string, "data":string}
/user/updateAvatar	GET	获得头像	'Authorization': (string)JWT令牌				{ "code": "0 (成功) /1 (失败) ", "message":string, "data":object}

1.1 注册

1.1.1 基本信息

- 请求路径: /user/register
- 请求方式: POST
- 接口描述: 该接口用于注册新用户

1.1.2 请求参数

请求参数格式: x-www-form-urlencoded

请求参数说明:

参数名称	说明	类型	是否必须	备注
id	电话或邮箱	string	是	只支持中国
fakename	用户名	string	是	非空字符
password	密码	string	是	非空字符

请求数据样例:

```
id=13123456789&fakename=zhangsan&password=123456
```

1.1.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

1.2 登录

1.2.1 基本信息

- 请求路径: /user/login
- 请求方式: POST
- 接口描述: 该接口用于登录

1.2.2 请求参数

请求参数格式: x-www-form-urlencoded

请求参数说明:

参数名称	说明	类型	是否必须	备注
id	电话或邮箱	string	是	只支持中国
password	密码	string	是	非空字符

请求数据样例:

```
id=13123456789&password=123456
```

1.2.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	string	必须		返回的数据, jwt令牌	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjbGFpbXMiOnsiaWQiOiJ1c2Z2JhIn0sImV4cCI6MTY5MzcxNTk3OH0.pE_RATcoF7Nm9KEp9eC3CzcBbkWAFOL0IsuMNjnZ95M"
}
```

1.2.4 备注说明

用户登录成功后，系统会自动下发JWT令牌，然后在后续的每次请求中，浏览器都需要在请求头header中携带到服务端，请求头的名称为 Authorization，值为 登录时下发的JWT令牌。

如果检测到用户未登录，则http响应状态码为401

1.3 获取用户详细信息

1.3.1 基本信息

请求路径：/user/userinfo

请求方式：GET

接口描述：该接口用于获取当前已登录用户的详细信息

1.3.2 请求参数

无

1.3.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	必须		返回的数据	
-id	string	非必须		电话或邮箱	
-fackname	string	非必须		昵称	
-userPic	string	非必须		头像地址	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": {
    "id": "135123456789",
    "fackname": "sawqw",
    "userPic": ""
  }
}
```

1.4 更新用户基本信息

1.4.1 基本信息

- 请求路径：/user/update
- 请求方式：PUT
- 接口描述：该接口用于更新已登录用户的基本信息(除头像和密码)

1.4.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
id	主键ID	string	是	电话或邮箱
fakename	用户名	string	是	非空字符
password	密码	string	是	非空字符

请求数据样例：

```
id=13123456789&fakename=zhangsan&password=123456
```

1.4.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

1.5 更新用户头像

1.5.1 基本信息

请求路径: /user/updateAvatar

请求方式: PATCH

接口描述: 该接口用于更新已登录用户的头像

1.5.2 请求参数

请求参数格式: queryString

请求参数说明:

参数名称	说明	类型	是否必须	备注
avatarUrl	头像地址	string	是	url地址

请求数据样例:

```
avatarUrl=https://big-event-gwd.oss-cn-beijing.aliyuncs.com/9bf1cf5b-1420-4c1b-91ad-e0f4631cbcd4.png
```

1.5.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

1.6 获取头像

1.6.1 基本信息

请求路径: /user/updateAvatar

请求方式: GET

接口描述: 该接口用于获取本地头像

1.6.2 请求参数

无

1.6.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": [
    "\\userpic\\little_bear.jpg",
    "\\userpic\\little_bear1.jpg"
  ]
}
```

1.7 随机更换头像

1.7.1 基本信息

请求路径: /user/updaterandom

请求方式: POST

接口描述: 该接口用于随机更换头像

1.7.2 请求参数

无

1.7.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

###

2. 聊天室管理

URL	方法	描述	请求头	请求体	响应头	响应体
/room	POST	创建聊天室	'Authorization': (string)JWT令牌	{ "roomName": string }		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":string }
/room	GET	查看现有所有聊天室列表	'Authorization': (string)JWT令牌			{ "code": "0 (成功) /1 (失败) ", "message": string, "data":object }
/room/roominfo	GET	查看聊天室信息	'Authorization': (string)JWT令牌	{ "roomName": string }		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":object }

URL	方法	描述	请求头	请求体	响应头	响应体
/room/roominfo	PUT	更新当前聊天室	'Authorization': (string)JWT令牌, 'roomname': (string)	{"roomname":string, "password":string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":string}
/room/roominfo	POST	加入聊天室	'Authorization': (string)JWT令牌, 'roomname': (string)	{ "password":string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":string}
/room/addmember	POST	为聊天室加入成员	'Authorization': (string)JWT令牌, 'roomname': (string)	{"id":string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":string}
/room/deletemember	POST	为聊天室删除成员	'Authorization': (string)JWT令牌, 'roomname': (string)	{"id":string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":string}
/room/roominfo	DELETE	删除聊天室	'Authorization': (string)JWT令牌, 'roomname': (string)	{ "roomname": string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":string}

2. 聊天室相关接口

2.1 新增聊天室

2.1.1 基本信息

请求路径：/room

请求方式：POST

接口描述：该接口用于新增聊天室

2.1.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
roomName	聊天室名	string	是	

请求数据样例：

```
roomName=zhangsan
```

2.1.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

2.2 聊天室列表

2.2.1 基本信息

请求路径：/room

请求方式：GET

接口描述：该接口用于获取当前已登录用户创建的所有聊天室

2.2.2 请求参数

无

2.2.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	List	必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": [
    "sad",
    "sak",
    "Dad"
  ]
}
```

2.3 聊天室信息

2.3.1 基本信息

请求路径：/room/roominfo

请求方式：GET

接口描述：该接口用于查看聊天室信息

2.3.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
roomName	聊天室名	string	是	

请求数据样例：

```
roomName=zhangsan
```

2.3.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": [
    {
      "roomName": "sads",
      "roomOwnerid": "dasdas",
      "numofpeople": 2,
      "membersId": "13123456789,14123456789"
    }
  ]
}
```

2.4 加入聊天室

2.4.1 基本信息

- 请求路径：/room/roominfo
- 请求方式：POST
- 接口描述：该接口用于成员加入聊天室

2.4.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
password	聊天室密码	string	否	已经在聊天室的就不用填了

请求数据样例：

```
password=zhangsan
```

2.4.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	user	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": [
    {
      "id": "13123456789",
      "fakeName": "hane",
      "cookie": null,
      "userpic": "http:adaad.com"
    }
  ]
}
```

2.5 更新当前聊天室

2.5.1 基本信息

- 请求路径：/room/roominfo
- 请求方式：PUT
- 接口描述：该接口用于更新当前聊天室

2.5.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
roomname	聊天室名字	string	否	没有改动就是原来的
owenerid	聊天室拥有者id	striing	否	没有改动就是原来的

请求数据样例：

roomName=zhangsan&&owenerid=13123456789

2.5.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	string	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

2.6 群主添加当前聊天室成员

2.6.1 基本信息

请求路径：/room/addmenber

请求方式：POST

接口描述：该接口用于群主添加当前聊天室成员

2.6.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
id	加入聊天室成员id	string	是	

请求数据样例：

id=13123456789

2.6.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	string	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

2.7 删除聊天室成员

2.7.1 基本信息

- 请求路径：/room/deletemember
- 请求方式：POST
- 接口描述：该接口用于群主删除聊天室成员

2.7.2 请求参数

请求参数格式：application/json

请求参数说明：

参数名称	说明	类型	是否必须	备注
id	删除成员的id	string	是	

请求数据样例：

```
id=13123456789
```

####

2.7.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	string	非必须		返回的数据	

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": null
}
```

3. 聊天功能

URL	方法	描述	请求头	请求体	响应头	响应体
/talk	POST	发送消息	'Authorization': (string)JWT令牌,'roomname': (string)	{"context":string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":object}
/talk	GET	显示当前聊天室的所有消息	'Authorization': (string)JWT令牌,'roomname': (string)			{ "code": "0 (成功) /1 (失败) ", "message": string, "data":object}
/talk/contextsearch	POST	查找消息	'Authorization': (string)JWT令牌,'roomname': (string)	{"context":string}		{ "code": "0 (成功) /1 (失败) ", "message": string, "data":object}

3.聊天相关接口

3.1 发送聊天内容

3.1.1 基本信息

请求路径: /talk

请求方式: POST

接口描述: 该接口用于发送聊天内容

3.1.2 请求参数

请求参数格式: application/json

请求参数说明:

参数名称	说明	类型	是否必须	备注
context	聊天内容	string	是	

请求数据样例:

```
context=13123456789
```

####

3.1.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": [
    {
      "time": "2023-12-3-16:00",
      "chatroomname": "sad",
    }
  ]
}
```

```
        "sendername": "das",
        "senderpic": " ",
        "context": "dadsdaafa"
    }
]
}
```

3.2 查找聊天内容

3.2.1 基本信息

- 请求路径: /talk/contextsearch
- 请求方式: POST
- 接口描述: 该接口用于发送聊天内容

3.2.2 请求参数

请求参数格式: application/json

请求参数说明:

参数名称	说明	类型	是否必须	备注
context	聊天内容	string	是	

请求数据样例:

```
context=jhgiugui
```

####

3.2.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	非必须		返回的数据	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": {}
}
```

```
    "data": [
      {
        "time": "2023-12-3-16:00",
        "chatroomname": "sad",
        "sendername": "das",
        "senderpic": " ",
        "context": "dadsdaafa"
      },
      {
        "time": "2023-12-3-16:03",
        "chatroomname": "sad",
        "sendername": "das",
        "senderpic": " ",
        "context": "dadsdaafadadad"
      }
    ]
  }
```

4. 其他接口

4.1 添加游戏

4.1.1 基本信息

- 请求路径: /game/addgame
- 请求方式: POST
- 接口描述: 该接口用于添加游戏

4.1.2 请求参数

请求参数格式: multipart/form-data

请求参数说明:

参数名称	说明	类型	是否必须	备注
id	唯一的id	String	是	
chatroom	房间名	String	是	
gameid	什么类型的游戏(五子棋1, 围棋2, 翻转棋3)	int	是	

请求数据样例:

无

4.1.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	必须			

响应数据样例:

```
{  
    "code": 0,  
    "message": "操作成功",  
    "data": {  
        "id": "0989090",  
        "chatroom": "32324",  
        "time": "2024-06-22 19:20:15",  
        "gameid": 1,  
        "winner": -1,  
        "turn": 0,  
        "board": "[[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],  
[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1,  
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1,  
-1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1,  
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]]",  
        "players": "18025160324",  
        "isend": 0  
    }  
}
```

4.2 更改游戏

4.2.1 基本信息

请求路径: /game/play

请求方式: POST

接口描述：该接口用于更改游戏

4.2.2 请求参数

请求参数格式: multipart/form-data

请求参数说明:

参数名称	说明	类型	是否必须	备注
id	唯一标识	String	是	
row	改变的行	int	是	
col	改变的列	int	是	

请求数据样例:

无

4.2.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	必须			

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": {
    "time": "2024-06-22 15:33:00",
    "gameid": 1,
    "winner": 1,
    "turn": 1,
    "board": "[[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, 0, 0, 1, 0, 0, 0, 0, 1, 0, -1, -1, -1, -1], [-1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, 0, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]",
    "players": "18025160324,13093160328",
  }
}
```

```
        "isend": 1
    }
}
```

4.3 获取游戏信息

4.3.1 基本信息

- 请求路径：/game/showgame
- 请求方式：POST
- 接口描述：该接口用于获取游戏信息

4.3.2 请求参数

请求参数格式：multipart/form-data

请求参数说明：

参数名称	说明	类型	是否必须	备注
id	唯一标识	String	是	

请求数据样例：

无

4.3.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	必须			

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": {
    "time": "2024-06-22 15:33:00",
    "gameid": 1,
    "winner": 1,
    "turn": 1,
    "board": "[[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, 0, 0, 1, 0, 0, 0, 0, 1, 0, -1, -1, -1, -1], [-1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, 0, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]]",
    "players": "18025160324,13093160328",
    "isend": 1
  }
}
```

4.4 加入游戏

4.4.1 基本信息

请求路径: /game/joingame

请求方式: POST

接口描述：该接口用于加入游戏

4.4.2 请求参数

请求参数格式: multipart/form-data

请求参数说明:

参数名称	说明	类型	是否必须	备注
id	唯一标识	String	是	

请求数据样例:

无

4.4.3 响应数据

响应数据类型: application/json

响应参数说明:

请求数据样例：

无

4.5.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	必须			

响应数据样例：

```
{
  "code": 0,
  "message": "操作成功",
  "data": "有人走了，删除当前游戏"
}
```

4.5 获取所有游戏

4.5.1 基本信息

- 请求路径：/game/showallgames
- 请求方式：POST
- 接口描述：该接口用于获取所有游戏

4.5.2 请求参数

无

4.5.3 响应数据

响应数据类型：application/json

响应参数说明：

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	object	必须			

响应数据样例：

```
{
```

```
"code": 0,  
"message": "操作成功",  
"data": [{  
    "id": "0989090",  
    "chatroom": "32324",  
    "time": "2024-06-22 15:33:00",  
    "gameid": 1,  
    "winner": 1,  
    "turn": 1,  
    "board": "[[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, 0, 0, 1, 0, 0, 0, 0, 1, 0, -1, -1, -1, -1], [-1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, 0, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]]",  
    "players": "18025160324,13093160328",  
    "isend": 1  
}, {  
    "id": "0989091",  
    "chatroom": "22324",  
    "time": "2024-06-23 15:33:00",  
    "gameid": 1,  
    "winner": 1,  
    "turn": 1,  
    "board": "[[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, 0, 0, 1, 0, 0, 0, 0, 1, 0, -1, -1, -1, -1], [-1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]]",  
    "players": "18025160323,13093160328",  
    "isend": 1  
}]
```

5. 其他接口

5.1 文件上传

5.1.1 基本信息

请求路径: /upload

请求方式: POST

接口描述: 该接口用于上传文件(单文件)

5.1.2 请求参数

请求参数格式: multipart/form-data

请求参数说明:

参数名称	说明	类型	是否必须	备注	
file	表单中文件请求参数的名字	file	是		

请求数据样例:

无

5.1.3 响应数据

响应数据类型: application/json

响应参数说明:

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须		响应码, 0-成功,1-失败	
message	string	非必须		提示信息	
data	string	必须		在本地的存储相对路径	

响应数据样例:

```
{
  "code": 0,
  "message": "操作成功",
  "data": "src/userpic/little_bear.jpg"
}
```