

# iteam-harmonyos-back接口文档

---

简介:iteam-harmonyos-back接口文档

HOST:localhost:8081

联系人:

Version:2.0

接口路径:/v2/api-docs

- iteam-harmonyos-back接口文档
- 好友相关接口
  - 根据userId查询申请列表
  - 根据userId查询好友列表
  - 处理好友申请
  - 发送好友申请
  - 查询当前用户和活动成员的关系：好友，已发送好友申请，未发送好友申请，被拒绝
- 文件上传接口
  - 上传文件
- 活动成员相关接口
  - 加入活动
  - 某个活动所有的成员列表
  - 获取小队信息(我加入的所有活动列表)
- 活动相关接口
  - 删除活动信息
  - 修改活动信息
  - 根据eventId获取某个活动的信息
  - 新增活动信息
  - 获取活动与用户之间的关系
  - 获取活动列表|搜索活动
  - 根据用户ID获取用户创建的活动
  - 获取附近活动列表
  - 活动推荐
- 活动评论相关接口
  - 新增评论

- 根据eventId查找所有相关评论
- 用户相关接口
  - 根据userId查询用户信息
  - 用户登录
  - 用户退出登录
  - 用户注册
  - 更新用户信息
  - 更新用户密码
  - 上传用户头像
- 聊天相关接口
  - 根据两个用户的id获取聊天记录列表
  - 根据chatId删除聊天记录
  - 新增聊天记录

## 好友相关接口

### 根据userId查询申请列表

接口地址:/friend/applyList

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«用户»»

状态码	说明	schema
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	用户
addr	地址	string	
createTime		string(date-time)	
gender	性别	string	
headImg	头像的路径	string	
id		integer(int64)	
lat	纬度	number(double)	
lon	经度	number(double)	
name	姓名	string	
password	密码	string	
username	用户名,登录的唯一凭证	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "addr": "",
      "createTime": "",
      "gender": "",
      "headImg": "",
      "id": 0,
      "lat": 0,
      "lon": 0,
      "name": "",

```

```
        "password": "",
        "username": ""
    },
    ],
    "success": true
}
```

# 根据userId查询好友列表

接口地址:/friend/getFriendList

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«用户»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	用户

参数名称	参数说明	类型	schema
addr	地址	string	
createTime		string(date-time)	
gender	性别	string	
headImg	头像的路径	string	
id		integer(int64)	
lat	维度	number(double)	
lon	经度	number(double)	
name	姓名	string	
password	密码	string	
username	用户名,登录的唯一凭证	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "addr": "",
      "createTime": "",
      "gender": "",
      "headImg": "",
      "id": 0,
      "lat": 0,
      "lon": 0,
      "name": "",
      "password": "",
      "username": ""
    }
  ],
  "success": true
}
```

# 处理好友申请

接口地址:/friend/processApply

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

# 发送好友申请

接口地址:`/friend/sendApply`

请求方式:`POST`

请求数据类型:`application/json`

响应数据类型:`*/*`

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

# 查询当前用户和活动成员的关系：好友，已发送好友申请，未发送好友申请，被拒绝

接口地址:`/friend/status`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
memberId	活动成员id	query	true	integer(int64)	
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«int»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		integer(int32)	integer(int32)
success		boolean	

响应示例:



```
{
  "info": "",
  "obj": 0,
  "success": true
}
```

# 文件上传接口

## 上传文件

接口地址:`/upload/uploadFile`

请求方式:`POST`

请求数据类型:`multipart/form-data`

响应数据类型:`*/*`

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«string»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	

参数名称	参数说明	类型	schema
obj		string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": "",
  "success": true
}
```

# 活动成员相关接口

## 加入活动

接口地址:/eventMember/join

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	

状态码	说明	schema
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

# 某个活动所有的成员列表

接口地址:`/eventMember/list`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
eventId	活动id	query	true	integer(int64)	
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«活动成员(数据传输对象)»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	活动成员(数据传输对象)
addr	用户地址	string	
apply	该用户和我的关系,0:拒绝,1:好友,2:已发送申请	integer(int32)	
headImg	用户头像路径	string	
name	用户姓名	string	
userId	用户id	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "addr": "",
      "apply": 0,
      "headImg": "",
      "name": "",
      "userId": ""
    }
  ],
  "success": true
}
```

# 获取小队信息(我加入的所有活动列表)

接口地址:`/eventMember/teamList`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«活动小队(数据传输对象)»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	活动小队(数据传输对象)
addr	活动地址	string	
createUserId	哪个用户创建的,他的id	integer(int64)	
eventId	活动的id	integer(int64)	

参数名称	参数说明	类型	schema
eventName	活动名称	string	
headImg	活动海报的路径	string	
memberCount	参加活动的人数	integer(int32)	
userInfoName	用户名	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "addr": "",
      "createUserId": 0,
      "eventId": 0,
      "eventName": "",
      "headImg": "",
      "memberCount": 0,
      "userInfoName": ""
    }
  ],
  "success": true
}
```

## 活动相关接口

### 删除活动信息

接口地址:/event/delete

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
eventId	活动id	query	true	integer(int64)	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

# 修改活动信息

接口地址:</event/edit>

请求方式:PUT

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
event	活动实体	body	true	string	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«活动»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		活动	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	



参数名称	参数说明	类型	schema
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	
state	活动状态	integer(int32)	
updateTime		string(date-time)	
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {
    "addr": "",
    "createTime": "",
    "headImg": "",
    "id": 0,
    "ifDelete": 0,
    "intro": "",
    "lat": 0,
    "lon": 0,
    "name": "",
    "startTime": "",
    "state": 0,
    "updateTime": "",
    "userId": 0
  },
  "success": true
}
```

## 根据eventId获取某个活动的信息

接口地址:/event/info

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
eventId	活动id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«活动»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		活动	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	

参数名称	参数说明	类型	schema
state	活动状态	integer(int32)	
updateTime		string(date-time)	
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {
    "addr": "",
    "createTime": "",
    "headImg": "",
    "id": 0,
    "ifDelete": 0,
    "intro": "",
    "lat": 0,
    "lon": 0,
    "name": "",
    "startTime": "",
    "state": 0,
    "updateTime": "",
    "userId": 0
  },
  "success": true
}
```

## 新增活动信息

接口地址:/event/insert

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
event	活动实体	body	true	string	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«活动»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		活动	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	
state	活动状态	integer(int32)	
updateTime		string(date-time)	

参数名称	参数说明	类型	schema
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {
    "addr": "",
    "createTime": "",
    "headImg": "",
    "id": 0,
    "ifDelete": 0,
    "intro": "",
    "lat": 0,
    "lon": 0,
    "name": "",
    "startTime": "",
    "state": 0,
    "updateTime": "",
    "userId": 0
  },
  "success": true
}
```

## 获取活动与用户之间的关系

接口地址:/event/isJoin

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
eventId	活动id	query	true	integer(int64)	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«int»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		integer(int32)	integer(int32)
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": 0,
  "success": true
}
```

# 获取活动列表|搜索活动

接口地址:/event/list

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
kw	搜索关键字	query	false	string	
userId	用户id	query	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«活动»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	
state	活动状态	integer(int32)	

参数名称	参数说明	类型	schema
updateTime		string(date-time)	
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "addr": "",
      "createTime": "",
      "headImg": "",
      "id": 0,
      "ifDelete": 0,
      "intro": "",
      "lat": 0,
      "lon": 0,
      "name": "",
      "startTime": "",
      "state": 0,
      "updateTime": "",
      "userId": 0
    }
  ],
  "success": true
}
```

## 根据用户ID获取用户创建的活动

接口地址:`/event/myEventList`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:



参数名称	参数说明	请求类型	是否必须	数据类型	schema
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«活动»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	
state	活动状态	integer(int32)	
updateTime		string(date-time)	
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "addr": "",
      "createTime": "",
      "headImg": "",
      "id": 0,
      "ifDelete": 0,
      "intro": "",
      "lat": 0,
      "lon": 0,
      "name": "",
      "startTime": "",
      "state": 0,
      "updateTime": "",
      "userId": 0
    }
  ],
  "success": true
}
```

# 获取附近活动列表

接口地址:/event/nearbyList

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
distance	距离	query	true	number(double)	
targetLat	目标精度	query	true	number(double)	
targetLont	目标维度	query	true	number(double)	

响应状态:

状态码	说明	schema
200	OK	R«List«活动»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	
state	活动状态	integer(int32)	
updateTime		string(date-time)	
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
```

```
{
    "addr": "",
    "createTime": "",
    "headImg": "",
    "id": 0,
    "ifDelete": 0,
    "intro": "",
    "lat": 0,
    "lon": 0,
    "name": "",
    "startTime": "",
    "state": 0,
    "updateTime": "",
    "userId": 0
},
"success": true
}
```

# 活动推荐

接口地址:`/event/recommend`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«活动»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		活动	活动
addr	活动地址	string	
createTime		string(date-time)	
headImg	活动海报图片的路径	string	
id		integer(int64)	
ifDelete	活动是否已被删除	integer(int32)	
intro	活动简介	string	
lat	维度	number(double)	
lon	经度	number(double)	
name	活动名称	string	
startTime	活动开始时间	string(date-time)	
state	活动状态	integer(int32)	
updateTime		string(date-time)	
userId	哪个用户创建的,他的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {
    "addr": "",
    "createTime": "",
    "headImg": "",
    "id": 0,
    "ifDelete": 0,
    "intro": "",
    "lat": 0,
    "lon": 0,
    "name": "",
    "startTime": "",
    "state": 0,
    "updateTime": "",
  }
}
```

```
        "userId": 0
    },
    "success": true
}
```

# 活动评论相关接口

## 新增评论

接口地址:/eventComment/insert

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
eventComment	评论实体,须包含 eventId,userId,content 字段	body	true	string	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

## 根据eventId查找所有相关评论

接口地址:`/eventComment/listByEventId`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
eventId	活动id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«活动评论(数据传输对象)»»
401	Unauthorized	

状态码	说明	schema
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	活动评论(数据传输对象)
content	评论内容	string	
createTime		string(date-time)	
headImg	哪个用户评论的,他的头像	string	
name	哪个用户评论的,他的name	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "content": "",
      "createTime": "",
      "headImg": "",
      "name": ""
    }
  ],
  "success": true
}
```

# 用户相关接口



# 根据userId查询用户信息

接口地址:`/user/info`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userId	用户id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

```
    "success": true
}
```

# 用户登录

接口地址:/user/login

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
user	请求体,须包含username和password两个字段	body	true	string	

响应状态:

状态码	说明	schema
200	OK	R«用户»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		用户	用户

参数名称	参数说明	类型	schema
addr	地址	string	
createTime		string(date-time)	
gender	性别	string	
headImg	头像的路径	string	
id		integer(int64)	
lat	维度	number(double)	
lon	经度	number(double)	
name	姓名	string	
password	密码	string	
username	用户名,登录的唯一凭证	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {
    "addr": "",
    "createTime": "",
    "gender": "",
    "headImg": "",
    "id": 0,
    "lat": 0,
    "lon": 0,
    "name": "",
    "password": "",
    "username": ""
  },
  "success": true
}
```

## 用户退出登录

接口地址:</user/logout>

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

# 用户注册

接口地址:/user/reg

请求方式:POST

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
user	用户实体,至少包含username和password两个字段	body	true	string	

响应状态:

状态码	说明	schema
200	OK	R«用户»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		用户	用户
addr	地址	string	
createTime		string(date-time)	
gender	性别	string	
headImg	头像的路径	string	
id		integer(int64)	
lat	维度	number(double)	

参数名称	参数说明	类型	schema
lon	经度	number(double)	
name	姓名	string	
password	密码	string	
username	用户名,登录的唯一凭证	string	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {
    "addr": "",
    "createTime": "",
    "gender": "",
    "headImg": "",
    "id": 0,
    "lat": 0,
    "lon": 0,
    "name": "",
    "password": "",
    "username": ""
  },
  "success": true
}
```

# 更新用户信息

接口地址:/user/update

请求方式:PUT

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
user	用户实体,须包含用户id和要更新的字段	body	true	string	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

# 更新用户密码

接口地址:</user/updatePassword>

请求方式:PUT

请求数据类型:application/json

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
user	用户实体,须包含用户id和要更新的字段	body	true	string	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```



```
}      "success": true
```

# 上传用户头像

接口地址:/user/uploadHeadImg

请求方式:POST

请求数据类型:multipart/form-data

响应数据类型:\*/\*

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

## 聊天相关接口

### 根据两个用户的id获取聊天记录列表

接口地址:`/chat/byUserIdFromAndTo`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`*/*`

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userIdFrom	用户1的id	query	true	integer(int64)	
userIdTo	用户2的id	query	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«聊天记录»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		array	聊天记录
content	消息内容	string	
createTime		string(date-time)	
id		integer(int64)	
statusFrom	发送方消息状态,0:正常,1:删除	integer(int32)	
statusTo	接收方消息状态,0:正常,1:删除	integer(int32)	
userIdFrom	发送方用户的id	integer(int64)	
userIdTo	接收方用户的id	integer(int64)	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": [
    {
      "content": "",
      "createTime": "",
      "id": 0,
      "statusFrom": 0,
      "statusTo": 0,
      "userIdFrom": 0,
      "userIdTo": 0
    }
  ],
  "success": true
}
```

## 根据chatId删除聊天记录

接口地址:`/chat/delete`

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:\*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
chatId	chat的id	query	true	integer(int64)	
userId	userId	header	false	integer(int64)	
userIdFrom	如果是发送方删除消息,发送方的id,注意只能传一方	query	false	integer(int64)	
userIdTo	如果是接收方删除消息,接收方的id,注意只能传一方	query	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```

```
    "obj": {},  
    "success": true  
}
```

# 新增聊天记录

接口地址: /chat/insert

请求方式: POST

请求数据类型: application/json

响应数据类型: \*/\*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
chat	聊天记录chat实体	body	true	string	
userId	userId	header	false	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«object»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
info		string	
obj		object	

参数名称	参数说明	类型	schema
success		boolean	

响应示例:

```
{
  "info": "",
  "obj": {},
  "success": true
}
```