

# 前后端接口说明书

## 序

为保证安全性，所有接口请使用https访问，将考虑在网关禁用所有http访问。

约定API描述中，用一对大括号 {、} 包裹的内容为当前位置参数的描述，特别地，{BASE\_URL} 表示域名。被一对 [、] 包裹的内容标识可选的内容。

例如以下描述：

```
1 | {BASE_URL}/api/question?qid={qid}
```

实际使用时，将转换为：

```
1 | https://aiape.icu/api/question?qid=1024
2 |
3 | 权限分为三种：`游客`、`用户`、`管理员`，权限等级依次提高，权限较高的用户可以进行全校较低的用户的操作。其中`用户`以上的权限需要登录获得，在请求相关方式时，请求头中需包含`Authorization`项，值为`Bearer {token}`，其中`token`为登录成功后服务器返回的[RFC7519 - JWT](https://tools.ietf.org/html/rfc7519#section-4)令牌，使用参考[JWT](https://jwt.io/introduction)。描述中权限一栏描述为`用户`时，`管理员`即可访问，而`用户`/`管理员`一般表示该接口`用户`有权以自己为目标调用，`管理员`有权对所有人为目标调用，具体情况会有详细描述。
4 |
5 | 关于时间格式，参考[自定义日期和时间格式字符串](https://docs.microsoft.com/zh-cn/dotnet/standard/base-types/custom-date-and-time-format-strings)，一般值为`yyyy/MM/dd-HH:mm:ss`，4位年份、2位月份、2位日期用`/`隔开来表示日期，2位时（24小时制）、2位分、2位秒用`:`隔开来表示时间，日期和时间用`-`连接，所以位数不足时高位用`0`补齐。
```

索引(并且为了方便，将省略{BASE\_URL}):

URL	类型	权限	功能简介
<a href="#">/api/bot/start</a>	POST	用户	开始机器人对话
<a href="#">/api/bot/message</a>	POST	用户	向机器人发送消息
<a href="#">/api/user/signup</a>	POST	游客	注册用户
<a href="#">/api/user/login</a>	POST	游客	用户登录
<a href="#">/api/user/public info</a>	GET	游客	获取用户公开信息
<a href="#">/api/user/internal info</a>	GET	用户	获取用户内部信息
<a href="#">/api/user/full info</a>	GET	用户 / 管理员	获取用户详细信息
<a href="#">/api/user/modify</a>	PUT	用户 / 管理员	修改用户信息
<a href="#">/api/user/questions</a>	GET	用户	获取用户创建的所有问题
<a href="#">/api/user/answers</a>	GET	用户	获取用户创建的所有回答
<a href="#">/api/user/fresh</a>	POST	游客	刷新令牌
<a href="#">/api/questions/question</a>	GET	游客	获取问题的详细信息
<a href="#">/api/questions/answer</a>	GET	游客	获取回答的详细信息
<a href="#">/api/questions/tag</a>	GET	游客	获取标签的详细信息
<a href="#">/api/questions/questionlist</a>	GET	游客	获取问题列表
<a href="#">/api/questions/taglist</a>	GET	游客	获取全部标签
<a href="#">/api/questions/add question</a>	POST	用户	添加问题
<a href="#">/api/questions/add answer</a>	POST	用户	添加回答

URL	类型	权限	功能简介
<a href="#">/api/questions/add_tag</a>	POST	用户	添加标签
<a href="#">/api/questions/modify_question</a>	PUT	用户 / 管理员	修改问题
<a href="#">/api/questions/modify_answer</a>	PUT	用户 / 管理员	获取回答的详细信息
<a href="#">/api/questions/modify_tag</a>	PUT	用户 / 管理员	获取标签的详细信息
<a href="#">/api/questions/delete_question</a>	DELETE	用户 / 管理员	删除问题
<a href="#">/api/questions/delete_answer</a>	DELETE	用户 / 管理员	获取回答的详细信息
<a href="#">/api/questions/delete_tag</a>	DELETE	用户 / 管理员	获取标签的详细信息

## 机器人相关

### 创建机器人对话 /api/bot/start

- 权限：用户
- url格式：

```
1 | {BASE_URL}/api/bot/start
```

- 类型：POST
- 请求体：

```
1 | {
2 |     // 目前是空的
3 | }
```

- 请求体各参数含义：
  - 目前是空的
- 响应体：
  - 参见[机器人响应](#)

### 发送机器人消息 /api/bot/message

- 权限：用户
- url格式：

```
1 | {BASE_URL}/api/bot/message
```

- 类型：POST
- 请求体：

```
1 {  
2   "message" : "用户发送的消息"  
3 }
```

- 请求体各参数含义：
  - `message`：用户发送的消息；
- 响应体：
  - 参见[机器人响应](#)

## 机器人响应

- 可能的状态码：
  - `200 OK` 正常，此时响应体内的JSON有效；
  - `401 Unauthorized` 未登录；

```
1 {  
2   "message": "这里是主体消息",  
3   "prompt":  
4     [  
5       "Car",  
6       "Bus"  
7     ]  
8 }
```

- 各字段含义（注意，除非声明，任何字段都可能为空）
  - `message`：纯文本信息，字符串；
  - `prompt`：建议选项，字符串列表。表示建议用户选择的选项，用户可以选择不输入该选项；

## 用户相关

### 注册用户 /api/user/signup

- 权限：`游客`
- url格式：

```
1 | {BASE_URL}/api/user/signup
```

- 类型：POST
- 请求体：

```
1 {  
2   "name" : "昵称",  
3   "email" : "sample@buaa.edu.cn",  
4   "password" : "09azAZ@$!%*#_~?&^"  
5 }
```

- 请求体各参数含义：

- `name`: 昵称, 字符串。字符数量为2~18, 支持汉字、大小写字母、数字。不可与其他用户重复;
- `email`: 邮箱地址, 字符串。仅允许ASCII中可打印字符的合法的邮箱地址, 根据[RFC5321-SMTP](#)标准第 4.5.3.1.1. 条至 4.5.3.1.3. 条, 用户名 (@ 前) 长度不超过64字节, 总长度不超过254字节;
- `password`: 密码, 字符串。仅允许大小写字母、数字、`@$!%*#_~?&^`, 长度为8~16。明文传输, 请务必确保以https发送;
- 响应体:

```
1 {  
2   "status" : "success",  
3   "message" : "signup success"  
4 }
```

- 响应各参数含义:
  - `status`: 字符串, 表示处理状态, `success` 表示成功, 其余表示失败;
  - `message`: 信息, 作为 `status` 的扩充信息, 更易于阅读;
- `status` 取值及其含义:

取值	状态码	含义
<code>success</code>	200 OK	成功注册
<code>nameInvalid</code>	200 OK	昵称格式不合法
<code>nameExisted</code>	200 OK	昵称已经存在
<code>emailInvalid</code>	200 OK	邮件地址格式不合法
<code>emailExisted</code>	200 OK	邮件地址已经存在
<code>passwordInvalid</code>	200 OK	密码格式不合法

## 用户登录 `/api/user/login`

- 权限: `游客`
- url格式:

```
1 {BASE_URL}/api/user/login
```

- 类型: POST
- 请求体:

```
1 {  
2   "email" : "sample@buaa.edu.cn",  
3   "password" : "09azAZ@$!%*#_~?&^"  
4 }
```

- 请求体各参数含义:



- `status`: 字符串, 表示处理状态, `success` 表示成功, 其余表示失败;
- `name`: 昵称, 字符串。字符数量为2~18, 支持汉字、大小写字母、数字。不可与其他用户重复;
- `status` 取值及其含义:

取值	状态码	含义
<code>success</code>	200 OK	成功
<code>userNotExist</code>	404 Not Found	<code>uid</code> 对应的用户不存在

## 获取用户内部信息 /api/user/internal\_info

- 权限: `用户`
- url格式:

```
1 | {BASE_URL}/api/user/internal_info[?uid={uid}]
```

- 类型: GET
- 简介: 获取 `uid` 指定的用户的公开用户信息。当 `uid` 为-1或未提供 `uid` 参数时, 将视为查询当前用户。
- 响应体:

```
1 | {
2 |   "status" : "success",
3 |   "uid" : 12345,
4 |   "name" : "昵称",
5 |   "email" : "example@buaa.edu.cn"
6 | }
```

- 响应各参数含义:
  - `status`: 字符串, 表示处理状态, `success` 表示成功, 其余表示失败;
  - `uid`: 用户ID, uint;
  - `name`: 昵称, 字符串。字符数量为2~18, 支持汉字、大小写字母、数字。不可与其他用户重复;
  - `email`: 邮箱地址, 字符串, **若用户未公开邮箱, 则为空字符串**。仅包含ASCII中可打印字符的合法的邮箱地址, 用户名 (@ 前) 长度不超过64字节, 总长度不超过254字节;
- `status` 取值及其含义:

取值	状态码	含义
<code>success</code>	200 OK	成功
<code>userNotExist</code>	404 Not Found	<code>uid</code> 对应的用户不存在
	401 Unauthorized	权限不足

# 获取用户详细信息 /api/user/full\_info

- 权限： 用户 / 管理员
- url格式：

```
1 | {BASE_URL}/api/user/full_info[?uid={uid}]
```

- 类型： GET
- 简介： 获取uid 指定的用户的详细用户信息。当 uid 为-1或未提供 uid 参数时， 将视为查询当前用户；当查询目标为当前用户时， 只需要 用户 权限；当查询目标不是当前用户时， 需要 管理员 权限。
- 响应体：

```
1 | {
2 |     "status" : "success",
3 |     "uid" : 12345,
4 |     "name" : "昵称",
5 |     "email" : "example@buaa.edu.cn",
6 |     "auth" : 0
7 | }
```

- status 取值及其含义：
  - status： 字符串，表示处理状态， success 表示成功， 其余表示失败；
  - name： 昵称，字符串。字符数量为2~18，支持汉字、大小写字母、数字。不可与其他用户重复；
  - email： 邮箱地址，字符串。仅包含ASCII中可打印字符的合法的邮箱地址，用户名（@前）长度不超过64字节，总长度不超过254字节；
  - uid： 用户ID， uint；
  - auth： 权限等级， int。 1 表示用户， 2 表示管理员；
- status 取值及其含义：

取值	状态码	含义
success	200 OK	成功
userNotExist	404 Not Found	uid 对应的用户不存在
	401 Unauthorized	权限不足

# 修改用户信息 /api/user/modify

- 权限： 用户 / 管理员
- url格式：

```
1 | {BASE_URL}/api/user/modify
```



- 类型：PUT
- 简介：修改用户的详细信息，要么全部成功，要么全部失败。
- 请求体：**所有项目均为可选项**

```
1 {
2     "uid" : 1234,
3     "name" : "新昵称",
4     "password" : "09azAZ@$!%*#_~?&^",
5     "auth" : 1
6 }
```

- 请求体各参数含义：
  - uid：待修改目标的ID，int。uid 为负数或不存在时，表示当前用户；当修改当前用户时，只需要 用户 权限，当修改目标不是当前用户时，需要 管理员权限；
  - name：新昵称，字符串，不包括此项表示不更改昵称。字符数量为2~18，支持汉字、大小写字母、数字。不可与其他用户重复；
  - auth：新权限等级，int，**只有管理员可以更改该字段**。1 表示用户，2 表示管理员。不可以修改当前用户的权限等级；
  - password：新密码，字符串。仅允许大小写字母、数字、@ \$ ! % \* # \_ ~ ? & ^，长度为8~16。不允许与旧密码相同。明文传输，请务必确保以https发送；
- 响应体：

```
1 [
2     {
3         "status" : "tryChangeSelfAuth",
4         "message" : "trying to change the auth of yourself"
5     },
6     {
7         "status" : "nameInvalid",
8         "message" : "the format of name is invalid"
9     }
10 ]
```

- 响应各参数含义：
  - 当HTTP状态码为 200 OK 时，表示修改成功；当状态码为 401 Unauthorized 时，表示权限不足，即未登录，或者非 管理员用户 尝试修改其他用户的信息，或者非 管理员用户 尝试修改 auth；当状态码为 409 Conflict 时，表示未发生权限错误，仅发生数据错误，将通过响应体返回所有的错误信息；
  - status：字符串，表示处理状态，success 表示成功，其余表示失败；
  - message：信息，作为 status 的扩充信息，更易于阅读；
- 各个错误 status 取值及其含义：

取值	状态码	含义
tryChangeSelfAuth	409 Conflict	尝试修改当前用户的权限
userNotExist	409 Conflict	用户不存在（只有 管理员 才可能触发此错误）
nameInvalid	409 Conflict	昵称格式不合法
nameExisted	409 Conflict	昵称已经存在
passwordInvalid	409 Conflict	密码格式不合法

<code>passwordSame</code>	<code>409_Conflict</code>	密码格式不合法
	401 <code>Unauthorized</code>	权限错误
	200 OK	成功修改

## 获取用户创建的所有问题 /api/user/questions

- 权限：用户
- url格式：

```
1 | {BASE_URL}/api/user/questions[&uid={uid}]
```

- 类型：GET
- 简介：查询某个用户创建的所有问题的 `qid`；
- 请求各参数含义：
  - `uid`：待查询的用户，uint。若为空，则表示查询当前用户；
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "query question information success",
4 |   "questions":
5 |   [
6 |     120, 100, 98, 21
7 |   ]
8 | }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
  - `questions`：用户创建的所有问题的 `qid`。按照最后修改时间由新到旧排序；
- `status` 取值及其含义：

取值	状态码	含义
success	200 OK	成功
userNotExist	404 Not Found	uid 对应的用户不存在

## 获取用户创建的所有回答 /api/user/answers

- 权限：用户
- url格式：

```
1 | {BASE_URL}/api/user/answers[&uid={uid}]
```

- 类型：GET
- 简介：查询某个用户创建的所有回答的 qid/aid；
- 请求各参数含义：
  - uid：待查询的用户，uint。若为空，则表示查询当前用户；
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "query question information success",
4 |   "answers":
5 |   [
6 |     "120/2", "100/3", "98/0", "21/3"
7 |   ]
8 | }
```

- 响应各参数含义：
  - status：字符串，表示处理状态，success 表示成功，其余表示失败；
  - message：信息，作为 status 的扩充信息，更易于阅读；
  - answers：用户创建的所有回答的 qid/aid，是用 / 分割的字符串。按照最后修改时间由新到旧排序；
- status 取值及其含义：

取值	状态码	含义
success	200 OK	成功
userNotExist	404 Not Found	uid 对应的用户不存在

## 刷新令牌 /api/user/fresh

- 权限：游客
- url格式：

```
1 | {BASE_URL}/api/user/fresh
```

- 类型：POST
- 简介：在JWT令牌超时前，获取新的令牌；
- 请求体：

```
1 | {
2 |   "token" :
   "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbn2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWlscy9uYW1lIjoimTIzIiwiaXhwIjoxnJE5MDYwMTM4LCJpc3MiOiJ0ZXN0LmNuIiwiaXVkiJoidGVzdCJ9.aci1lSQj01flQXkrATE70Q8KhH2SYBeAPFDIyktFefI"
3 | }
```

- 请求体各参数含义：
  - `token`：旧的令牌，需要未超时的令牌；
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "token fresh success",
4 |   "token" :
   "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbn2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWlscy9uYW1lIjoimTIzIiwiaXhwIjoxnJE5MDYwMTM4LCJpc3MiOiJ0ZXN0LmNuIiwiaXVkiJoidGVzdCJ9.aci1lSQj01flQXkrATE70Q8KhH2SYBeAPFDIyktFefI",
5 |   "timeout" : 600
6 | }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
  - `token`：[JSON Web Token](#)，作为用户信息的唯一标识；
  - `timeout`：令牌过期时间，uint，单位秒。应当在过期前调用[刷新令牌](#)；
- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	<code>200 OK</code>	成功
	<code>401 Unauthorized</code>	权限错误（可能令牌已过期）

## 问答相关

# 获取问题的详细信息 /api/questions/question

- 权限: 游客
- url格式:

```
1 | {BASE_URL}/api/questions/question?qid={qid}
```

- 请求URL各参数含义:
  - qid: 问题的ID, uint;
- 类型: GET
- 响应体:

```
1 | {
2 |   "status" : "success",
3 |   "message" : "query question information success",
4 |   "question" :
5 |   {
6 |     "question" : "问题的题目? ",
7 |     "remarks" : "这里是题目的具体描述",
8 |     "creator" : 1234,
9 |     "createTime" : "yyyy/MM/dd-HH:mm:ss",
10 |    "modifyTime" : "yyyy/MM/dd-HH:mm:ss",
11 |    "best" : 1,
12 |    "tags":
13 |    {
14 |      "语法" : 0,
15 |      "循环" : 1
16 |      // map of tag-name -> tid
17 |    },
18 |    "answers" :
19 |    [
20 |      0, 1, 2, 3 // list of aid
21 |    ]
22 |  }
23 | }
```

- 响应各参数含义:
  - status: 字符串, 表示处理状态, success 表示成功, 其余表示失败;
  - message: 信息, 作为 status 的扩充信息, 更易于阅读;
  - question: 问题信息对象;
  - question.question: 问题的题目, 字符串, 不超过20个字符;
  - question.remarks: 问题的详细描述, 字符串;
  - question.creator: 创建问题的用户的 uid;
  - question.createTime: 创建时间, 字符串;
  - question.modifyTime: 最后一次修改的时间, 字符串;
  - question.best: 最佳回答的 aid, int。值为负数时, 表示目前没有标记最佳回答;
  - question.tags: 问题的标签, 字符串->uint 映射。表示包含的标签名和对应的标签的 tid, 标签名不超过8个字符。如果只关心标签名, 可以忽略本映射的value, 只关注key;
  - question.answers: 现有回答, uint列表, 当前存在的回答的 aid;
- status 取值及其含义:

取值	状态码	含义
success	200 OK	成功
questionNotExist	404 Not Found	qid 对应的问题不存在

## 获取回答的详细信息 /api/questions/answer

- 权限：游客
- url格式：

```
1 | {BASE_URL}/api/questions/answer?qid={qid}&aid={aid}
```

- 请求URL各参数含义：
  - qid：该回答所属的问题，uint；
  - aid：该回答在所属问题内的ID，由[/api/question/question](#)获得，uint；
- 类型：GET
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "query question information success",
4 |   "answer" :
5 |   {
6 |     "content" : "这里是回答的内容",
7 |     "creator" : 1234,
8 |     "createTime" : "yyyy/MM/dd-HH:mm:ss",
9 |     "modifyTime" : "yyyy/MM/dd-HH:mm:ss"
10 |   }
11 | }
```

- 响应各参数含义：
  - status：字符串，表示处理状态，success 表示成功，其余表示失败；
  - message：信息，作为 status 的扩充信息，更易于阅读；
  - answer：回答信息对象；
  - answer.content：回答的内容，字符串；
  - answer.creator：回答问题的用户的 uid；
  - answer.createTime：创建时间，字符串；
  - answer.modifyTime：最后一次修改的时间，字符串；
- status 取值及其含义：

取值	状态码	含义
success	200 OK	成功
answerNotExist	404 Not Found	qid 和 aid 所确定的回答不存在

# 获取标签的详细信息 /api/questions/tag

- 权限：游客
- url格式：

```
1 | {BASE_URL}/api/questions/tag?tid={tid}
```

- 类型：GET
- 请求URL各参数含义：
  - tid：标签的ID, uint;
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "query question information success",
4 |   "tag" :
5 |     {
6 |       "name": "语法",
7 |       "desc": "编程语言语法相关问题：语句结构，关键字用法，类型系统，声明/定义语法结
8 |         构....."
9 |     }
10 | }
```

- 响应各参数含义：
  - status：字符串，表示处理状态，success表示成功，其余表示失败；
  - message：信息，作为status的扩充信息，更易于阅读；
  - tag：标签信息对象；
  - tag.name：标签名，字符串，不长于8个字符；
  - tag.desc：标签详细描述，字符串，不长于128字符；
- status取值及其含义：

取值	状态码	含义
success	200 OK	成功
tagNotExist	404 Not Found	tid 对应的标签不存在

## 获取问题列表 /api/questions/questionlist

- 权限：游客
- url格式：

```
1 | {BASE_URL}/api/questions/questionlist
```

- 类型：**POST**（因为参数较多，不采用GET方法）
- 简介：若指定了锚点，则返回创建时间早于锚点的若干问题的 `qid`（不包括锚点）；否则，返回创建时间最早的若干问题。可以指定标签，所有返回的 `qid` 对应的问题都包括所有指定的标签。可以指定返回的 `qid` 数量的最大值，但无论最大值有多少，最多只能返回64个。
- 请求体：

```
1 | {
2 |   "tags" :
3 |   [
4 |     1, 3, 4, 5
5 |   ],
6 |   "pt" : 123,
7 |   "number" : 20
8 | }
```

- 请求体各参数含义：
  - `tags`：用于筛选的标签，返回的问题都包括所有标签；
  - `pt`：锚点；
  - `number`：返回的 `qid` 数量的最大值；
- 响应体：

```
1 | [
2 |   200,
3 |   198,
4 |   197
5 | ]
```

- 响应含义：
  - `qid` 的列表，uint列表，按照创建时间由新到旧排序；
  - 若列表长度不足 `number` 且不足64，则说明不存在这样一个问题：它的 `qid` 不在响应列表中，且它的创建时间早于 `pt`；

## 获取全部标签 /api/questions/taglist

- 权限：游客
- url格式：



```
1 | {BASE_URL}/api/questions/taglist
```

- 类型: GET
- 简介: 获取所有标签的 tid;
- 响应体:

```
1 | {
2 |     "语法" : 0,
3 |     "循环" : 1
4 |     // map of tag-name -> tid
5 | }
```

- 响应体说明:
  - 响应体为映射, 类型为 字符串->uint, 表示标签名和 tid 的对应关系;

## 添加问题 /api/questions/add\_question

- 权限: 用户
- url格式:

```
1 | {BASE_URL}/api/questions/add_question
```

- 类型: POST
- 请求体:

```
1 | {
2 |     "question" : "问题的题目?",
3 |     "remarks" : "这里是题目的具体描述",
4 |     "tags":
5 |     [
6 |         0, 1 // list of tid
7 |     ],
8 | }
```

- 请求体各参数含义:
  - question: 问题的题目, 字符串, 不超过20个字符;
  - remarks: 问题的详细描述, 字符串;
  - tags: 标签列表, 内容为 tid, uint字符串;
- 响应体:

```
1 {  
2   "status" : "success",  
3   "message" : "new question add success"  
4 }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	200 OK	成功
<code>questionTooLong</code>	200 OK	题目过长

## 添加回答 /api/questions/add\_answer

- 权限：用户
- url格式：

```
1 | {BASE_URL}/api/questions/add_answer
```

- 类型：POST
- 请求体：

```
1 {  
2   "qid" : 123,  
3   "content" : "这里是回答的内容"  
4 }
```

- 请求体各参数含义：
  - `qid`：问题的ID，uint。同一个用户不能在同一个问题下添加多个回答；
  - `content`：回答的内容，字符串；
- 响应体：

```
1 {  
2   "status" : "success",  
3   "message" : "new answer add success"  
4 }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；

- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	200 OK	成功
<code>answerHasExist</code>	200 OK	当前用户已经为 <code>qid</code> 指定的问题添加过回答了
<code>questionTooLong</code>	200 OK	题目过长

## 添加标签 `/api/questions/add_tag`

- 权限：管理员
- url格式：

```
1 | {BASE_URL}/api/questions/add_tag
```

- 类型：POST
- 请求体：

```
1 | {
2 |   "name" : "语法",
3 |   "desc": "编程语言语法相关问题：语句结构，关键字用法，类型系统，声明/定义语法结构....."
4 | }
```

- 请求体各参数含义：
  - `name`：标签名，字符串，不超过8个字符，不能与任何已存在的标签拥有相同的名字；
  - `desc`：详细描述，字符串，不长于128字符；
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "new tag add success"
4 | }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
- `status` 取值及其含义：

取值	状态码	含义
success	200 OK	成功
nameTooLong	200 OK	标签名过长
nameHasExists	200 OK	标签名已经存在了
descTooLong	200 OK	描述过长

## 修改问题 /api/questions/modify\_question

- 权限：用户 / 管理员
- url格式：

```
1 | {BASE_URL}/api/questions/modify_question
```

- 类型：PUT
- 请求体：

```
1 | {
2 |   "qid" : 123,
3 |   "question" : "这里是新的题目",
4 |   "remarks" : "这里是新的内容",
5 |   "tags":
6 |   [
7 |     0, 1 // list of tid
8 |   ],
9 | }
```

- 请求体各参数含义：
  - qid：要修改的回答所在的问题的ID, uint;
  - question：新的题目，字符串，不超过20个字符。若不修改则不包括此项；
  - remarks：新的详细描述，字符串。若不修改则不包括此项；
  - tags：新的标签列表，内容为 tid，uint字符串。若不修改则不包括此项；
- 权限检查：
  - 若 qid 所确定的问题的创建者不是当前用户，且当前用户不是 管理员，则会报错权限错误；
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "question modification success",
4 | }
```

- 响应各参数含义：

- `status`: 字符串, 表示处理状态, `success` 表示成功, 其余表示失败;
- `message`: 信息, 作为 `status` 的扩充信息, 更易于阅读;
- `status` 取值及其含义:

取值	状态码	含义
<code>success</code>	<code>200 OK</code>	成功
<code>questionNotExist</code>	<code>404 Not Found</code>	<code>qid</code> 对应的问题不存在
	<code>401 Unauthorized</code>	权限不足

## 修改回答 `/api/questions/modify_answer`

- 权限: `用户 / 管理员`
- url格式:

```
1 | {BASE_URL}/api/questions/modify_answer
```

- 类型: PUT
- 请求体:

```
1 | {
2 |   "qid" : 123,
3 |   "aid" : 2,
4 |   "content" : "这里是新的内容"
5 | }
```

- 请求体各参数含义:
  - `qid`: 要修改的回答所在的问题的ID, uint;
  - `aid`: 要修改的回答在问题中的ID, uint;
  - `content`: 新的内容, 字符串;
- 权限检查:
  - 若 `qid`、`aid` 所确定的回答的创建者不是当前用户, 且当前用户不是 `管理员`, 则会报错权限错误;
- 响应体:

```
1 | {
2 |   "status" : "success",
3 |   "message" : "answer modification success"
4 | }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	<code>200 OK</code>	成功
<code>answerNotExist</code>	<code>404 Not Found</code>	<code>qid</code> 和 <code>aid</code> 所确定的回答不存在
	<code>401 Unauthorized</code>	权限不足

## 修改标签 `/api/questions/modify_tag`

- 权限：`管理员`
- url格式：

```
1 | {BASE_URL}/api/questions/modify_tag
```

- 类型：
- 请求体：

```
1 | {
2 |   "tid" : 123,
3 |   "name" : "语法",
4 |   "desc": "编程语言语法相关问题：语句结构，关键字用法，类型系统，声明/定义语法结构....."
5 | }
```

- 请求体各参数含义：
  - `tid`：要修改的标签的ID，uint；
  - `name`：新的标签名，字符串，不超过8个字符，不能与任何已存在的标签拥有相同的名字。  
**若不修改则不包括此项；**
  - `desc`：新的详细描述，字符串，不长于128字符。**若不修改则不包括此项；**
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "tag modification success"
4 | }
```

- 响应各参数含义：

- `status`: 字符串, 表示处理状态, `success` 表示成功, 其余表示失败;
- `message`: 信息, 作为 `status` 的扩充信息, 更易于阅读;
- `status` 取值及其含义:

取值	状态码	含义
<code>success</code>	200 OK	成功
<code>nameHasExists</code>	200 OK	标签名已经存在了
<code>nameTooLong</code>	200 OK	标签名过长
<code>descTooLong</code>	200 OK	描述过长
<code>tagNotExist</code>	404 Not Found	<code>tid</code> 对应的标签不存在
	401 Unauthorized	权限不足

## 删除问题 `/api/questions/delete_question`

- 权限: 用户 / 管理员
- url格式:

```
1 | {BASE_URL}/api/questions/delete_question
```

- 类型: DELETE
- 请求体:

```
1 | {
2 |   "qid"
3 | }
```

- 请求体各参数含义:
  - `qid`: 需要删除的问题的ID;
- 权限检查:
  - 若 `qid` 所确定的问题的创建者不是当前用户, 且当前用户不是 管理员, 则会报错权限错误;
- 响应体:

```
1 | {
2 |   "status" : "success",
3 |   "message" : "question removed success"
4 | }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	<code>200 OK</code>	成功
<code>questionNotExist</code>	<code>404 Not Found</code>	<code>qid</code> 对应的问题不存在
	<code>401 Unauthorized</code>	权限不足

## 删除回答 `/api/questions/delete_answer`

- 权限：`用户 / 管理员`
- url格式：

```
1 | {BASE_URL}/api/questions/delete_answer
```

- 类型：
- 请求体：

```
1 | {
2 |   "qid" : 123,
3 |   "aid" : 2,
4 | }
```

- 请求体各参数含义：
  - `qid`：要删除的回答所在的问题的ID, uint;
  - `aid`：要删除的回答在问题中的ID, uint;
- 权限检查：
  - 若 `qid`、`aid` 所确定的回答的创建者不是当前用户，且当前用户不是 `管理员`，则会报错权限错误；
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "answer removed"
4 | }
```



- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	<code>200 OK</code>	成功
<code>answerNotExist</code>	<code>404 Not Found</code>	<code>qid</code> 和 <code>aid</code> 所确定的回答不存在
	<code>401 Unauthorized</code>	权限不足

## 删除标签 `/api/questions/delete_tag`

- 权限：管理员
- url格式：

```
1 | {BASE_URL}/api/questions/delete_tag
```

- 类型：DELETE
- 请求体：

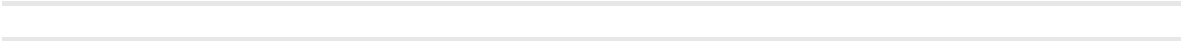
```
1 | {
2 |   "tid" : 123
3 | }
```

- 请求体各参数含义：
  - `tid`：要删除的标签的ID, uint;
- 响应体：

```
1 | {
2 |   "status" : "success",
3 |   "message" : "tag removed"
4 | }
```

- 响应各参数含义：
  - `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
  - `message`：信息，作为 `status` 的扩充信息，更易于阅读；
- `status` 取值及其含义：

取值	状态码	含义
success	200 OK	成功
tagNotExist	404 Not Found	tid 对应的标签不存在
	401 Unauthorized	权限不足



# 模板

## /api/

- 权限：
- url格式：

```
1 | {BASE_URL}/api/
```

- 类型：
- 请求体：

```
1 | {
2 | }
```

- 请求体各参数含义：
  -
- 响应体：

```
1 | {
2 |   "status" : "success",
3 | }
```

- 响应各参数含义：

- `status`：字符串，表示处理状态，`success` 表示成功，其余表示失败；
- 
- `status` 取值及其含义：

取值	状态码	含义
<code>success</code>	<code>200 OK</code>	成功