

蛋博Web API文档

版本20201122.9

UPD: 更改博客内容API

目录

[一、简介](#)

[二、登陆API](#)

- [0.确认登陆](#)
- [1.注册](#)
- [2.发送注册验证码](#)
- [3.找回密码](#)
- [4.发送找回密码验证码](#)
- [5.登陆](#)
- [6.登出](#)

[三、用户API](#)

- [1.修改地址](#)
- [2.修改生日](#)
- [3.修改性别](#)
- [4.修改昵称](#)
- [5.修改头像](#)
- [6.修改签名](#)
- [7.修改密码](#)
- [8.获取地址](#)
- [9.获取生日](#)
- [10.获取性别](#)
- [11.获取昵称](#)
- [12.获取头像](#)
- [13.获取签名](#)
- [14.获取邮箱](#)

[四、关注API](#)

- [1.关注](#)
- [2.取关](#)
- [3.获取关注列表](#)
- [4.获取被关注列表](#)

[五、博客API](#)

- [1.发布博客](#)
- [2.转发博客](#)
- [3.点赞](#)
- [4.取消点赞](#)
- [5.评论](#)
- [6.获取信息流](#)
- [7.获取用户博客](#)

- [8. 获取博客点赞](#)
- [9. 获取博客评论](#)

一、简介

1. 请求使用HTTP协议的multipart/form-data格式，需要包含相应的键值对。格式描述如下：

name:type

name表示键值名，type表示键值类型，包括text文本和file文件两种。

2. 登出、修改个人信息、发送博客和获取用户名必须包含一个Cookie，这个Cookie在登陆时加入，在登出时删除。

3. 返回值按如下JSON格式发送：

```
{
  "err_code": <int, 200 means success, otherwise fail>,
  "message": <str, human-readable message replied from the server>,
  "data": <data, this part is different from api to api>
}
```

4. 错误码规范为：

- 前两位标明功能40 注册 41登陆 42 修改密码 43修改属性 44获取属性
- 第三位标明错误1 不存在 2 不正确 3 格式错误
- 200表示成功执行。

5. 图像获取的是相对路径，绝对路径为：MEDIA_ROOT + 相对路径，MEDIA_ROOT = 127.0.0.1:8000/media/ 部署在不同服务器上时，请用真实IP地址和端口

6. 本系统仅支持学邮注册，也即邮箱必须以@fudan.edu.cn结尾。

二、登陆API

登陆API都以 /user/ 开头，包含用户的登陆，注册，找回密码和确认登陆等功能，注意这里的找回密码不需要登陆。

0.确认登陆

每个页面必须先调用这个接口确认用户有无登陆。

请求地址：/user/getUsername

请求方式：POST

请求内容：需要Cookie

返回内容:

```
{"error_code":441,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "获取用户名成功", "data": nickname}
```

1.注册

请求地址: /user/register

请求方式: POST

请求内容:

username:text

password:text

r_password:text

email:text

code:text

返回内容:

```
{"error_code":401,"message":"用户名已注册","data":None}
```

```
{"error_code":401,"message":"邮箱已注册","data":None}
```

```
{"error_code": 402, "message": "密码只能由大小写字母，数字组成，且长度应在6-20", "data": None}
```

```
{"error_code":402,"message":"两次输入的密码不一致","data":None}
```

```
{"error_code": 403, "message": "邮箱格式不正确", "data": None}
```

```
{"error_code": 403, "message": "验证码不正确或已过期", "data": None}
```

```
{"error_code": 200, "message": "注册成功", "data": None}
```

2.发送注册验证码

向用户邮箱发送注册验证码

请求地址: /user/sendRegisterCode

请求方式: POST

请求内容:

email:text

返回内容:

```
{"error_code":401,"message":"邮箱已注册","data":None}
```

```
{"error_code": 403, "message": "邮箱格式不正确", "data": None}
```

```
{"error_code": 200, "message": "邮件发送成功", "data": None}
```

3.找回密码

请求地址: /user/modifyPwd

请求方式: POST

请求内容:

username:text

password:text

r_password:text

email:text

code:text

返回内容:

```
{"error_code":421,"message":"用户名不存在","data":None}
```

```
{"error_code": 423, "message": "密码只能由大小写字母，数字组成，且长度应在6-20", "data": None}
```

```
{"error_code":422,"message":"两次输入的密码不一致","data":None}
```

```
{"error_code": 423, "message": "邮箱格式不正确", "data": None}
```

```
{"error_code": 422, "message": "该邮箱不是您注册时填写的邮箱", "data": None}
```

```
{"error_code": 422, "message": "验证码不正确或已过期", "data": None}
```

```
{"error_code": 200, "message": "密码修改成功", "data": None}
```

4.发送找回密码验证码

向用户邮箱发送登录时找回密码的验证码

请求地址: /user/sendLoginCode

请求方式: POST

请求内容:

email:text

返回内容:

```
{"error_code": 423, "message": "邮箱格式不正确", "data": None}
```

```
{"error_code": 422, "message": "该邮箱不是您注册时填写的邮箱", "data": None}
```

```
{"error_code": 200, "message": "邮件发送成功", "data": None}
```

5. 登陆

请求地址: /user/login

请求方式: POST

请求内容:

username:text

password:text

返回内容:

```
{"error_code": 411, "message": "用户不存在", "data": None}
```

```
{"error_code": 412, "message": "密码不正确", "data": None}
```

```
{"error_code": 200, "message": "登录成功", "data": None}
```

6. 登出

请求地址: /user/logout

请求方式: POST

请求内容: 需要Cookie

返回内容:

```
{"error_code": 411, "message": "当前未登录", "data": None}
```

```
{"err_code": 200, "message": "成功退出登录", "data": None}
```

三、用户API

用户API都以 /user/ 开头, 包含用户的信息修改, 信息获取等功能, 注意这里的修改密码仅在登陆状态下有效。

1. 修改地址

请求地址: /user/modifyAddress

请求方式: POST

请求内容: 需要Cookie

address:text

返回内容:

```
{"error_code":431,"message":"用户名不存在","data":None}
```

```
{"error_code": 433, "message": "地址长度应小于40个字符", "data": None}
```

```
{"error_code": 200, "message": "地址修改成功", "data": None}
```

2.修改生日

请求地址: /user/modifyBirthday

请求方式: POST

请求内容: 需要Cookie

birthday:text

The format of birthday is"YYYY-MM-DD"

返回内容:

```
{"error_code":431,"message":"用户名不存在","data":None}
```

```
{"error_code": 433, "message": "昵称长度应小于40个字符", "data": None}
```

```
{"error_code": 200, "message": "生日修改成功", "data": None}
```

3.修改性别

请求地址: /user/modifyGender

请求方式: POST

请求内容: 需要Cookie

gender:text

gender has a limited value to '男','女' or '保密'

返回内容:

```
{"error_code":431,"message":"用户名不存在","data":None}
```

```
{"error_code": 433, "message": "性别错误", "data": None}
```

```
{"error_code": 200, "message": "性别修改成功", "data": None}
```

4.修改昵称

请求地址: /user/modifyNickname

请求方式: POST

请求内容：需要Cookie

nickname:text

返回内容：

```
{"error_code":431,"message":"用户名不存在","data":None}
```

```
{"error_code": 433, "message": "昵称长度应小于20个字符，且不能为空", "data": None}
```

```
{"error_code": 200, "message": "昵称修改成功", "data": None}
```

5.修改头像

请求地址：/user/modifyProfile

请求方式：POST

请求内容：需要Cookie

profile:file

返回内容：

```
{"error_code":431,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "头像修改成功", "data": None}
```

6.修改签名

请求地址：/user/modifySignature

请求方式：POST

请求内容：需要Cookie

signature:text

返回内容：

```
{"error_code":431,"message":"用户名不存在","data":None}
```

```
{"error_code": 433, "message": "签名长度应小于30个字符", "data": None}
```

```
{"error_code": 200, "message": "签名修改成功", "data": None}
```

7.修改密码

请求地址：/user/modifyPwdLogin

请求方式：POST

请求内容：需要Cookie

password:text

r_password:text

code:text

返回内容:

```
{"error_code":421,"message":"用户名不存在","data":None}
```

```
{"error_code": 423, "message": "密码只能由大小写字母，数字组成，且长度应在6-20", "data": None}
```

```
{"error_code":422,"message":"两次输入的密码不一致","data":None}
```

```
{"error_code": 422, "message": "验证码不正确或已过期", "data": None}
```

```
{"error_code": 200, "message": "密码修改成功", "data": None}
```

8.获取地址

请求地址: /user/getAddress

请求方式: POST

请求内容:

username:text

返回内容:

```
{"error_code":441,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "获取地址成功", "data": address}
```

9.获取生日

请求地址: /user/getBirthday

请求方式: POST

请求内容:

username:text

返回内容:

```
{"error_code":441,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "获取生日成功", "data": birthday}
```

10.获取性别

请求地址: /user/getGender

请求方式: POST

请求内容:

username:text

返回内容:

{"error_code":441,"message":"用户名不存在","data":None}

{"error_code": 200, "message": "获取性别成功", "data": gender}

11.获取昵称

请求地址: /user/getNickname

请求方式: POST

请求内容:

username:text

返回内容:

{"error_code":441,"message":"用户名不存在","data":None}

{"error_code": 200, "message": "获取昵称成功", "data": nickname}

12.获取头像

获取用户头像相对路径。

请求地址: /user/getProfile

请求方式: POST

请求内容:

username:text

返回内容:

{"error_code":441,"message":"用户名不存在","data":None}

{"error_code": 200, "message": "获取头像路径成功", "data": profile_path}

13.获取签名

请求地址: /user/getSignature

请求方式: POST

请求内容:

username:text

返回内容:

```
{"error_code":441,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "获取签名成功", "data": signature}
```

14.获取邮箱

请求地址: /user/getEmail

请求方式: POST

请求内容:

username:text

返回内容:

```
{"error_code":441,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "获取邮箱成功", "data": signature}
```

四、关注API

关注API都以 /user/ 开头, 包括关注, 取关, 获取关注列表, 获取被关注列表等操作。

1.关注

请求地址: /user/follow

请求方式: POST

请求内容: 需要Cookie

to_username:text

返回内容:

```
{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}
```

```
{"error_code": 431, "message": "关注对象不存在", "data": None}
```

```
{"error_code": 432, "message": "请不要重复关注", "data": None}
```

```
{"error_code": 200, "message": "关注成功", "data": None}
```

2.取关

请求地址: /user/cancelFollow

请求方式: POST

请求内容: 需要Cookie

to_username:text

返回内容:

{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}

{"error_code": 431, "message": "取消关注的对象不存在", "data": None}

{"error_code": 433, "message": "当前还未关注", "data": None}

{"error_code": 200, "message": "取消关注成功", "data": None}

3.获取关注列表

请求地址: /user/getFollowers

请求方式: POST

请求内容:

username:text

返回内容:

{"error_code":441,"message":"用户名不存在","data":None}

{"error_code": 200, "message": "获取关注列表成功", "data": signature}

4.获取被关注列表

请求地址: /user/getFollowees

请求方式: POST

请求内容:

username:text

返回内容:

{"error_code":441,"message":"用户名不存在","data":None}

{"error_code": 200, "message": "获取关注者列表成功", "data": signature}

五、博客API

博客API都以 /blog/ 开头, 包括发布和获取博客, 点赞, 转发, 评论等功能。

1.发布博客

请求地址: /blog/releaseBlog

请求方式: POST

请求内容: 需要Cookie

content:text

pictures:file

pictures:file

...

You may attach several pictures in one post.

返回内容:

```
{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}
```

```
{"error_code":433, "message":"正文内容不能超过256字", "data":None}
```

```
{"error_code":433, "message":"图片最多只能上传9张", "data":None}
```

```
{"error_code":433, "message":"博客内容不能为空", "data":None}
```

```
{"error_code": 200, "message": "博客发布成功", "data": None}
```

2.转发博客

请求地址: /blog/repostBlog

请求方式: POST

请求内容: 需要Cookie

content:text

blog_id:text

返回内容:

```
{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}
```

```
{"error_code":433, "message":"转发正文内容不能超过100字", "data":None}
```

```
{"error_code": 442, "message": "转发的目标博客不存在", "data": None}
```

```
{"error_code": 200, "message": "博客转发成功", "data": None}
```

3.点赞

请求地址: /blog/giveLike

请求方式：POST

请求内容：需要Cookie

blog_id:text

返回内容：

{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}

{"error_code": 441, "message": "点赞的目标博客不存在", "data": None}

{"error_code": 442, "message": "请不要重复点赞", "data": None}

{"error_code": 200, "message": "点赞成功", "data": None}

4.取消点赞

请求地址：/blog/cancelLike

请求方式：POST

请求内容：需要Cookie

blog_id:text

返回内容：

{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}

{"error_code": 441, "message": "取消点赞的目标博客不存在", "data": None}

{"error_code": 442, "message": "当前还未点赞", "data": None}

{"error_code": 200, "message": "点赞成功", "data": None}

5.评论

请求地址：/blog/comment

请求方式：POST

请求内容：需要Cookie

content:text

blog_id:text

返回内容：

{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}

{"error_code": 433, "message": "评论内容不能超过30字", "data": None}

{"error_code": 443, "message": "评论的目标博客不存在", "data": None}

```
{"error_code":433, "message":"评论内容不能为空", "data":None}
```

```
{"error_code": 200, "message": "评论成功", "data": None}
```

6.获取信息流

刷新动态主页，获得自己和关注对象的所有博客

请求地址：/blog/refreshBlogs

请求方式：POST

请求内容：需要Cookie

返回内容：

```
{"error_code": 431, "message": "用户名不存在或当前未登录", "data": None}
```

```
{"error_code": 200, "message": "获取博客成功", "data": data}
```

7.获取用户博客

获取某一用户发布过的所有博客

请求地址：/blog/getBlog

请求方式：POST

请求内容：

username:text

返回内容：

```
{"error_code":441,"message":"用户名不存在","data":None}
```

```
{"error_code": 200, "message": "获取博客成功", "data": data}
```

这里 data 是一个字典结构，键值是博客ID，已经按发布顺序逆序排序：

```
data[b.id] = {
    'time': time,
    'origin_user': b.user.username,
    'origin_content': b.content,
    'users': users
    'contents': contents
    'pictures': picture_paths,
}
```

users 含有各级转发的用户的用户名

contents 含有各级转发的博客的内容

pictures 是原博客的图片

8. 获取博客点赞

请求地址: /blog/getLikes

请求方式: POST

请求内容:

blog_id:text

返回内容:

{"error_code": 442, "message": "目标博客不存在", "data": None}

{"error_code": 200, "message": "点赞获取成功", "data": like_usernames}

9. 获取博客评论

请求地址: /blog/getComments

请求方式: POST

请求内容:

blog_id:text

返回内容:

{"error_code": 442, "message": "目标博客不存在", "data": None}

{"error_code": 200, "message": "评论获取成功", "data": data}

这里 data 是一个词典结构, 键值是评论ID, 已经按发布顺序逆序排序:

```
data[cmt.id] = {
    'time': time,
    'username': cmt.user.username,
    'blog_id': cmt.id,
    'content': cmt.content,
}
```