

Flask中的错误处理

Flask中的错误处理

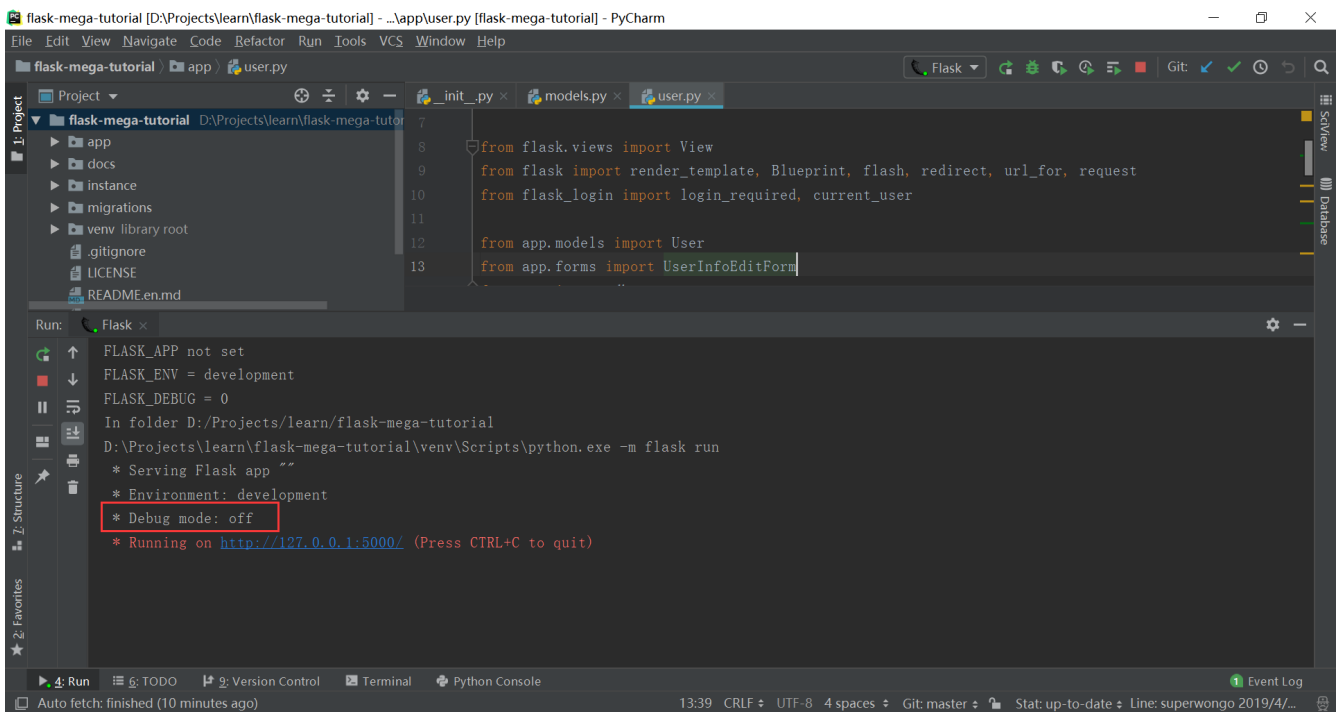
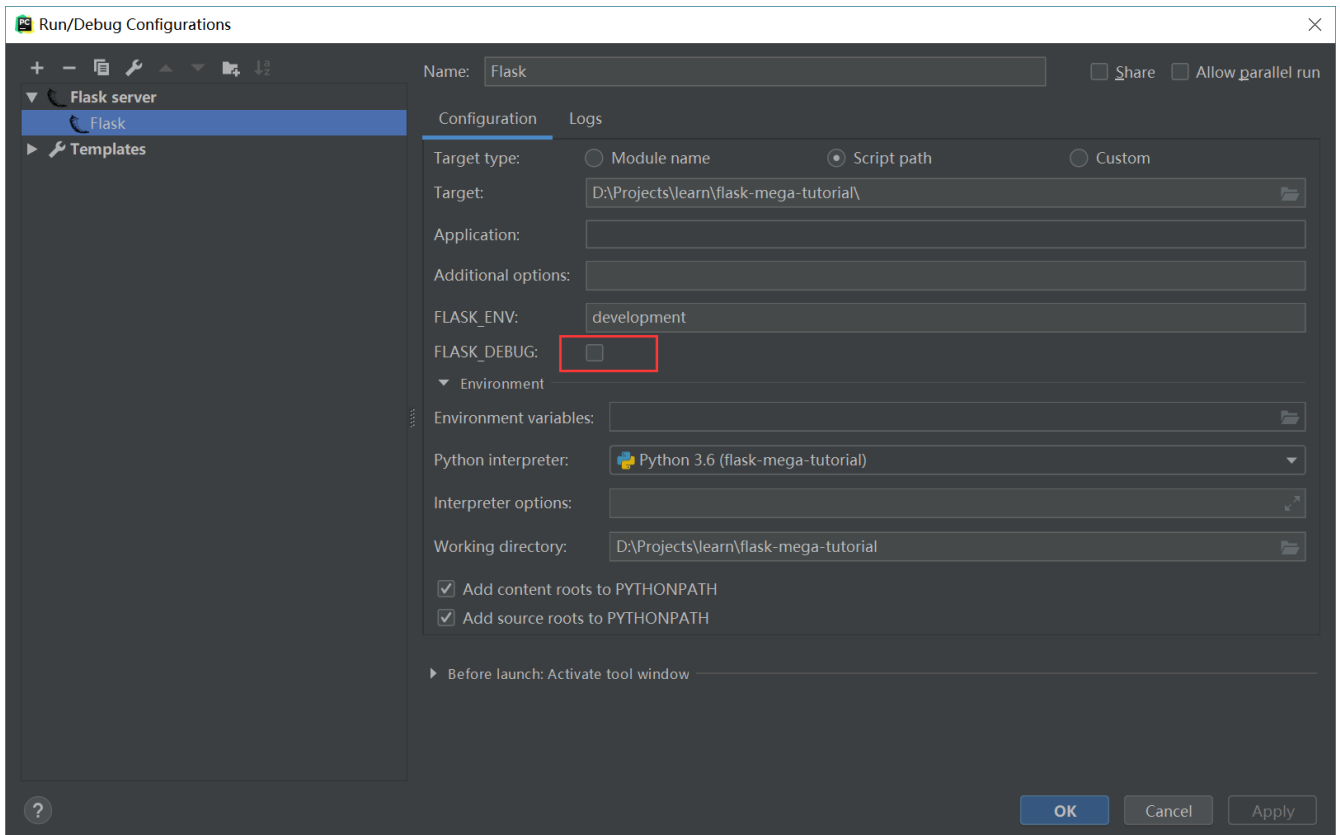
1. 当前已成功功能的bug再现
 - 1.1. 将测试参数调整为非Debug模式
 - 1.2. 将测试参数调整为Debug模式
2. 自定义错误页面
 - 2.1. 定义错误处理函数
 - 2.2. 新建404错误页面模板
 - 2.3. 新建500错误页面模板
 - 2.4. 启动服务测试结果
3. 通过电子邮件发送错误信息
 - 3.1. 配置电子邮箱参数
 - 3.2. 编写SMTPHandler实例
 - 3.3. 初始化邮件实例
 - 3.4. 编写RotatingFileHandler日志文件记录器
 - 3.5. 初始化日志文件记录器
 - 3.6. pycharm配置环境变量测试邮件发送
4. 解决该异常情况
 - 4.1. 表单增加已使用用户名校验
 - 4.2. 实例化表单时传入原用户名
 - 4.3. 启动服务测试

1. 当前已成功功能的bug再现

当前已成功功能中，用户资料修改存在一个bug，若本地存在两个用户admin和admin1，将admin的用户名修改为admin1，就会出现报错信息。对于此类服务器错误不应该简单粗暴的进行展示，而应该设置特定的页面进行错误的提示。

1.1. 将测试参数调整为非Debug模式

在非debug模式下，页面不会展示具体的错误信息。

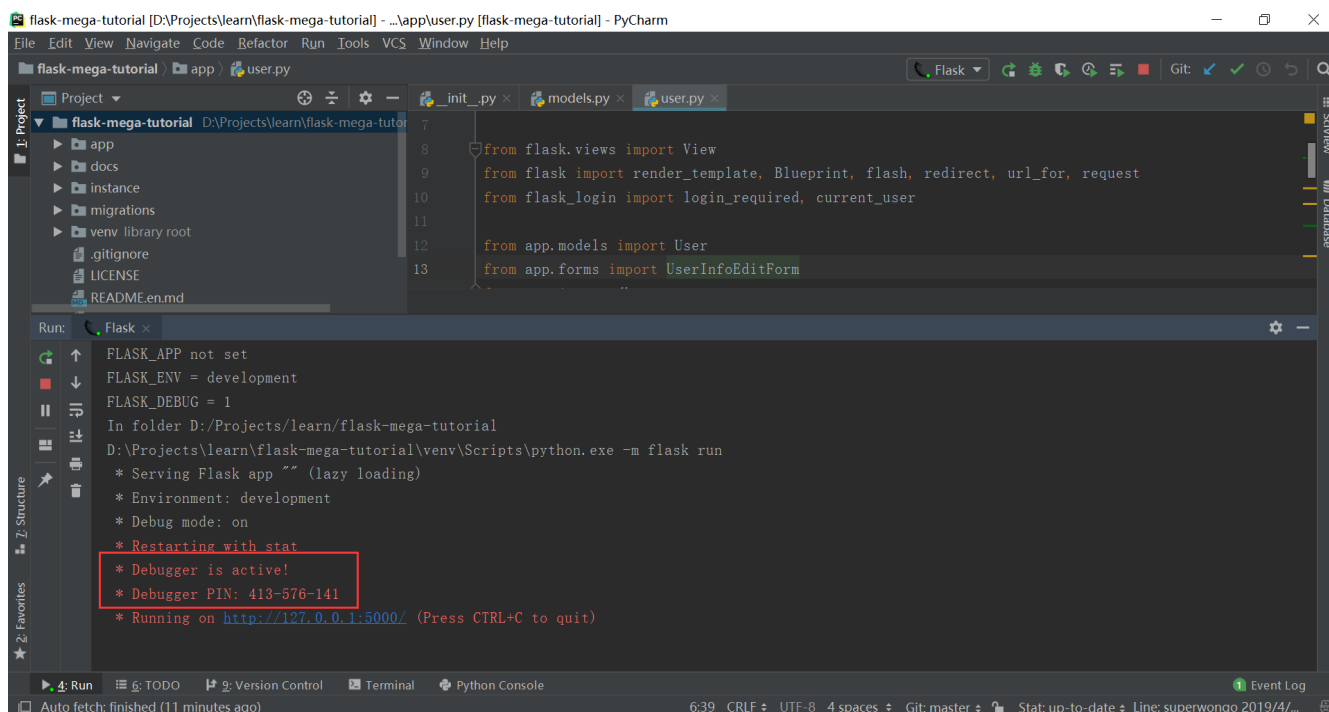


Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

1.2. 将测试参数调整为Debug模式

在debug模式下，可以展示具体的错误信息。



The screenshot shows the PyCharm IDE interface. The top pane displays the code for `user.py`, which includes imports for `View`, `render_template`, `Blueprint`, `flash`, `redirect`, `url_for`, `request`, `login_required`, `current_user`, `User`, and `UserInfoEditForm`. The bottom pane shows the Run console output for the Flask application. The output indicates that the application is running in debug mode, with the debugger active and the PIN 413-576-141. The application is running on `http://127.0.0.1:5000/`.

```
flask-mega-tutorial [D:\Projects\learn\flask-mega-tutorial] - ...app\user.py [flask-mega-tutorial] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
flask-mega-tutorial flask-mega-tutorial app user.py
Project flask-mega-tutorial D:\Projects\learn\flask-mega-tutorial
  app
  docs
  instance
  migrations
  venv library root
  .gitignore
  LICENSE
  README.en.md
Run: Flask
FLASK_APP not set
FLASK_ENV = development
FLASK_DEBUG = 1
In folder D:/Projects/learn/flask-mega-tutorial
D:\Projects\learn\flask-mega-tutorial\venv\Scripts\python.exe -m flask run
* Serving Flask app "" (lazy loading)
* Environment: development
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 413-576-141
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

sqlalchemy.exc.IntegrityError

sqlalchemy.exc.IntegrityError: (sqlite3.IntegrityError) UNIQUE constraint failed: users.username
 [SQL: UPDATE users SET username=?, about_me=? WHERE users.id = ?]
 [parameters: ('admini', '这是测试信息', 1)]
 (Background on this error at: <http://sqlalche.me/e/gkpi>)

Traceback (most recent call last)

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\sqlalchemy\engine\base.py", line 1244, in _execute_context

cursor, statement, parameters, context

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\sqlalchemy\engine\default.py", line 552, in do_execute

cursor.execute(statement, parameters)

The above exception was the direct cause of the following exception:

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 2309, in __call__

return self.wsgi_app(environ, start_response)

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 2295, in wsgi_app

response = self.handle_exception(e)

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 1741, in handle_exception

reraise(exc_type, exc_value, tb)

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask_compat.py", line 35, in reraise

raise value

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 2292, in wsgi_app

response = self.full_dispatch_request()

File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 2025, in full_dispatch_request

2. 自定义错误页面

使用 `@errorhandler` 装饰器，来声明自定义错误处理程序。

2.1. 定义错误处理函数

修改 `app/__init__.py` 脚本，编写并注册自定义处理函数。

```
def create_app():
    .....

    @application.errorhandler(404)
    def not_found_error(error):
        return render_template('error/404.html'), 404

    @application.errorhandler(500)
    def internal_error(error):
        return render_template('error/500.html'), 500

    return application
```

2.2. 新建404错误页面模板

新建 `app/templates/error/404.html` 模板文件，增加404错误提示。

```
{% extends 'base.html' %}

{% block content %}
    <h1>网页未找到</h1>
    <p><a href="{{ url_for('index') }}">返回首页</a></p>
{% endblock %}
```

2.3. 新建500错误页面模板

新建 `app/templates/error/500.html` 模板文件，增加500错误提示。

```
{% extends 'base.html' %}

{% block content %}
    <h1>发生意外错误</h1>
    <p>已通知管理员，给您带来的不便，请多多原谅</p>
    <p><a href="{{ url_for('index') }}">返回首页</a></p>
{% endblock %}
```

2.4. 启动服务测试结果

将Debug模式关闭后启动服务，修改已存在的 `username` 时还是无法跳转至500错误页面，查看后台报错。

← → ↻ 127.0.0.1:5000/user/edit

Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

报错内容如下：

```
Traceback (most recent call last):
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\werkzeug\serving.py",
line 302, in run_wsgi
    execute(self.server.app)
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\werkzeug\serving.py",
line 290, in execute
    application_iter = app(environ, start_response)
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line
2309, in __call__
    return self.wsgi_app(environ, start_response)
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line
2295, in wsgi_app
    response = self.handle_exception(e)
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line
1748, in handle_exception
    return self.finalize_request(handler(e), from_error_handler=True)
  File "D:\Projects\learn\flask-mega-tutorial\app\__init__.py", line 102, in internal_error
    return render_template('error/500.html'), 500
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\templating.py",
line 135, in render_template
    context, ctx.app)
  File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\templating.py",
line 117, in _render
    rv = template.render(context)
```

```
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\jinja2\asyncsupport.py", line 76, in render
    return original_render(self, *args, **kwargs)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\jinja2\environment.py", line 1008, in render
    return self.environment.handle_exception(exc_info, True)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\jinja2\environment.py", line 780, in handle_exception
    reraise(exc_type, exc_value, tb)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\jinja2\_compat.py",
line 37, in reraise
    raise value.with_traceback(tb)
File "D:\Projects\learn\flask-mega-tutorial\app\templates\error\500.html", line 1, in
top-level template code
    {% extends 'base.html' %}
File "D:\Projects\learn\flask-mega-tutorial\app\templates\base.html", line 18, in top-
level template code
    <a href="{{ url_for('user.user_info', username=current_user.username) }}">个人资料</a>
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\jinja2\environment.py", line 430, in getattr
    return getattr(obj, attribute)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\attributes.py", line 276, in __get__
    return self.impl.get(instance_state(instance), dict_)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\attributes.py", line 677, in get
    value = state._load_expired(state, passive)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\state.py", line 660, in _load_expired
    self.manager.deferred_scalar_loader(self, toload)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\loading.py", line 979, in load_scalar_attributes
    only_load_props=attribute_names,
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\loading.py", line 208, in load_on_ident
    identity_token=identity_token,
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\loading.py", line 282, in load_on_pk_identity
    return q.one()
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\query.py", line 3275, in one
    ret = self.one_or_none()
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\query.py", line 3244, in one_or_none
    ret = list(self)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\query.py", line 3317, in __iter__
    return self._execute_and_instances(context)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\query.py", line 3339, in _execute_and_instances
    querycontext, self._connection_from_session, close_with_result=True
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\query.py", line 3354, in _get_bind_args
```

```

mapper=self._bind_mapper(), clause=querycontext.statement, **kw
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\query.py", line 3332, in _connection_from_session
conn = self.session.connection(**kw)
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\session.py", line 1123, in connection
execution_options=execution_options,
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\session.py", line 1129, in _connection_for_bind
engine, execution_options
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\session.py", line 407, in _connection_for_bind
self._assert_active()
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-
packages\sqlalchemy\orm\session.py", line 294, in _assert_active
% self._rollback_exception
sqlalchemy.exc.InvalidRequestError: This Session's transaction has been rolled back due to
a previous exception during flush. To begin a new transaction with this Session, first
issue Session.rollback(). Original exception was: (sqlite3.IntegrityError) UNIQUE
constraint failed: users.username
[SQL: UPDATE users SET username=?, about_me=? WHERE users.id = ?]
[parameters: ('admin1', '这是测试信息', 1)]
(Background on this error at: http://sqlalche.me/e/gkpj)

```

通过错误提示可以看出，由于首先是 username 修改时，数据库提交异常但未进行session会话回滚，导致 base.html 页面中 current_user 获取失败。因此修改 app/user.py 脚本，增加数据库异常回滚处理。

```

class UserInfoEditView(View):
    """用户资料编辑"""
    methods = ['GET', 'POST']
    decorators = [login_required]

    def dispatch_request(self):
        form = UserInfoEditForm()
        # 验证通过更新当前登录用户信息
        if form.validate_on_submit():
            current_user.username = form.username.data
            current_user.about_me = form.about_me.data
            # db.session.commit()
            try:
                db.session.commit()
            except Exception as e:
                db.session.rollback()
                raise e
            flash('您的修改已保存')
            return redirect(url_for('user.user_info_edit'))
        .....

```

修改后，重启服务测试结果。

博客: [首页](#) [个人资料](#) [退出](#)

发生意外错误

已通知管理员，给您带来的不便，请多多原谅

[返回首页](#)

3. 通过电子邮件发送错误信息

3.1. 配置电子邮箱参数

修改 `app/config.py` 配置文件，增加电子邮箱相关参数。其中 `MAIL_USERNAME` 为邮箱地址，`MAIL_PASSWORD` 非邮箱登录地址，而是用于第三方邮件客户端专用的授权码。其中授权码需要自行设置。

126 网易免费邮
www.126.com

2@126.com

电脑客户端 | 升级VIP | 升级服务 | 设置 | 帮助 | 退出

首页 | 通讯录 | 应用中心 | 收件箱 | 网易严选 | 免费课程 | 设置

来信分类

帐号与邮箱中心

邮箱安全设置

邮箱手机服务

反垃圾/黑白名单

POP3/SMTP/IMAP

客户端授权密码

文件夹和标签

多标签窗口

换肤

手机号码邮箱



授权码

授权码是用于登录第三方邮件客户端的专用密码。
适用于登录以下服务: POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务。

设置客户端授权码: ☒ 开启 ☐ 关闭 (默认)

您已启用授权码，请使用授权码登录第三方邮件客户端

重置授权码

启用时间	停用时间
2019-04-28 16:02:09	未停用

启用授权码，避免密码泄漏造成邮箱安全隐患，使用邮件客户端更安心。[了解更多>>](#)


```
# -----EMAIL相关配置-----#
# 电子邮箱服务器
MAIL_SERVER = os.environ.get('MAIL_SERVER')
# 电子邮箱端口, 标准端口为25
MAIL_PORT = int(os.environ.get('MAIL_PORT') or 25)
# 电子邮件服务器凭证默认不使用
MAIL_USE_TLS = os.environ.get('MAIL_USE_TLS') is not None
# 电子邮箱服务器用户名
MAIL_USERNAME = os.environ.get('MAIL_USERNAME')
# 电子邮箱服务器密码
MAIL_PASSWORD = os.environ.get('MAIL_PASSWORD')
# 电子邮箱邮件接收地址
MAIL_ADMINS = ['your-email@example.com']
```

3.2. 编写SMTPHandler实例

新增 app/logger.py 脚本, 编写创建电子邮件实例 SMTPHandler 的函数。

```
import logging
from logging.handlers import SMTPHandler

def init_email(app):
    if app.config['MAIL_SERVER']:
        auth = None
        if app.config['MAIL_USERNAME'] or app.config['MAIL_PASSWORD']:
            auth = (app.config['MAIL_USERNAME'], app.config['MAIL_PASSWORD'])
        secure = None
        if app.config['MAIL_USE_TLS']:
            secure = ()
        mail_handler = SMTPHandler(
            # 电子邮箱服务器地址
            mailhost=(app.config['MAIL_SERVER'], app.config['MAIL_PORT']),
            # 邮件发送地址
            fromaddr=app.config['MAIL_USERNAME'],
            # 邮件接收地址
            toaddrs=app.config['MAIL_ADMINS'],
            # 邮件标题
            subject='flask-mega-tutorial博客异常',
            # 邮箱验证信息
            credentials=auth,
            # 是否启用加密
            secure=secure
        )
        mail_handler.setLevel(logging.ERROR)
        app.logger.addHandler(mail_handler)
```

3.3. 初始化邮件实例

修改 app/__init__.py 脚本, 新增邮件实例初始化处理。

```
def create_app():
    .....
    # 非DEBUG模式下, 异常日志通过电子邮件发送
    if not application.debug:
        from app.logger import init_email
        init_email(application)
    .....
    return application
```

3.4. 编写RotatingFileHandler日志文件记录器

修改 `app/logger.py` 脚本, 编写 `RotatingFileHandler` 日志文件记录器。

```
import os
import logging
from logging.handlers import SMTPHandler, RotatingFileHandler

def init_logger(app):
    # 创建logs目录, 用于存放日志文件
    if not os.path.exists('logs'):
        os.mkdir('logs')

    # 设置RotatingFileHandler类, 最大日志文件大小为100kb, 只保留10个备份文件, 其会自动进行日志文件的切割和清理
    file_handler = RotatingFileHandler('logs/microblog.log', maxBytes=102400,
    backupCount=10)
    # logging.Formatter类为日志消息提供自定义格式
    # 分别记录了时间戳、日志记录级别、消息、日志来源的源代码文件和行号
    file_handler.setFormatter(logging.Formatter(
        '%(asctime)s %(levelname)s: %(message)s [in %(pathname)s:%(lineno)d]'
    ))
    # 设置日志类别: 分别是DEBUG、INFO、WARNING、ERROR和CRITICAL
    file_handler.setLevel(logging.INFO)
    app.logger.addHandler(file_handler)

    # 每次服务重新启动, 都会登记一条日志
    app.logger.setLevel(logging.INFO)
    app.logger.info('微博已启动')
```

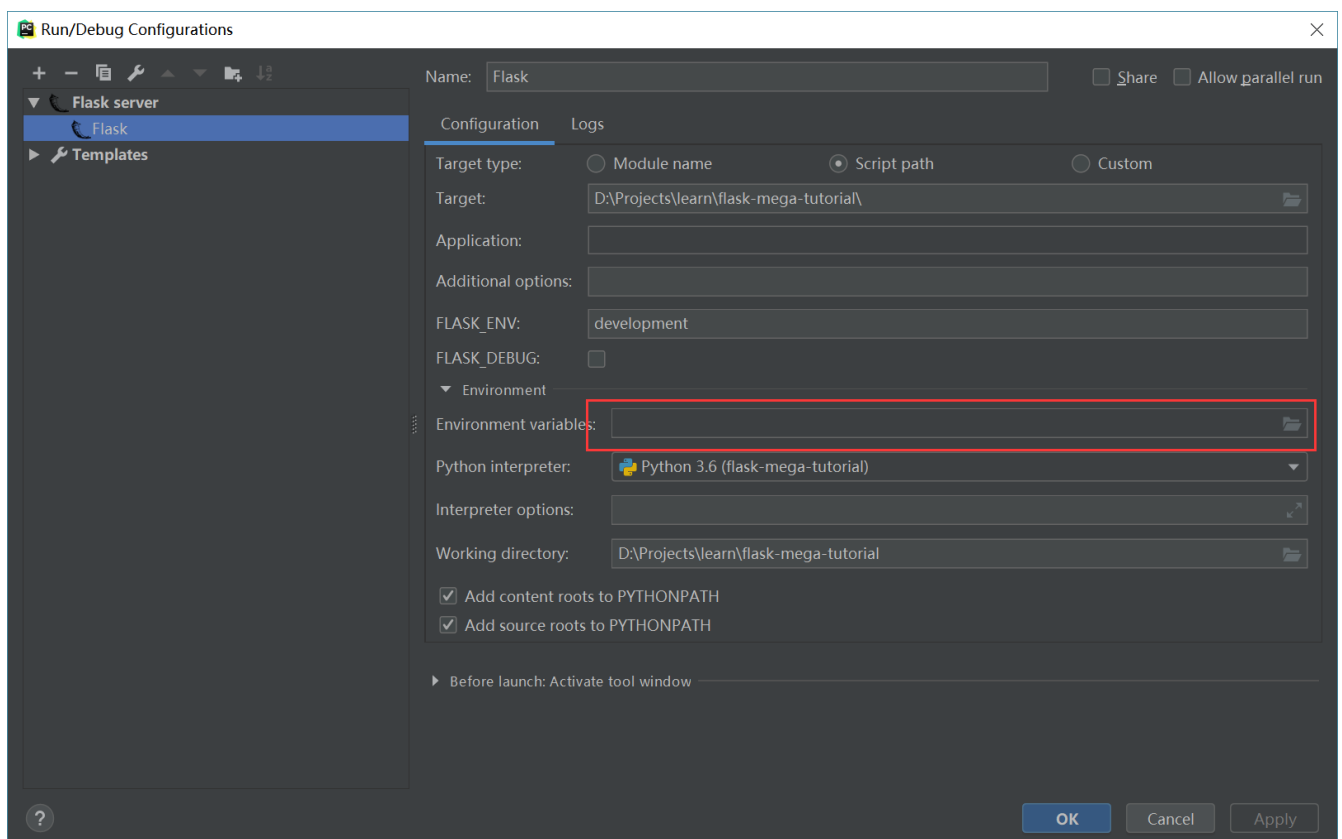
3.5. 初始化日志文件记录器

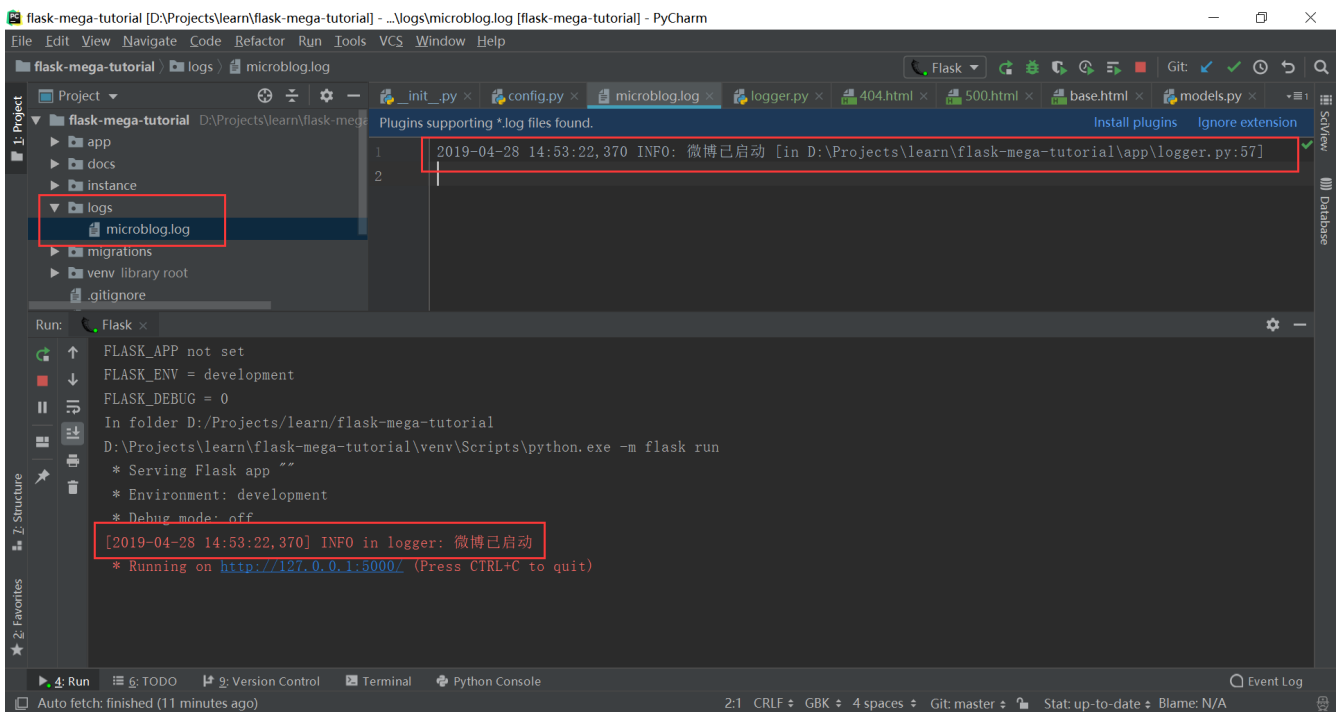
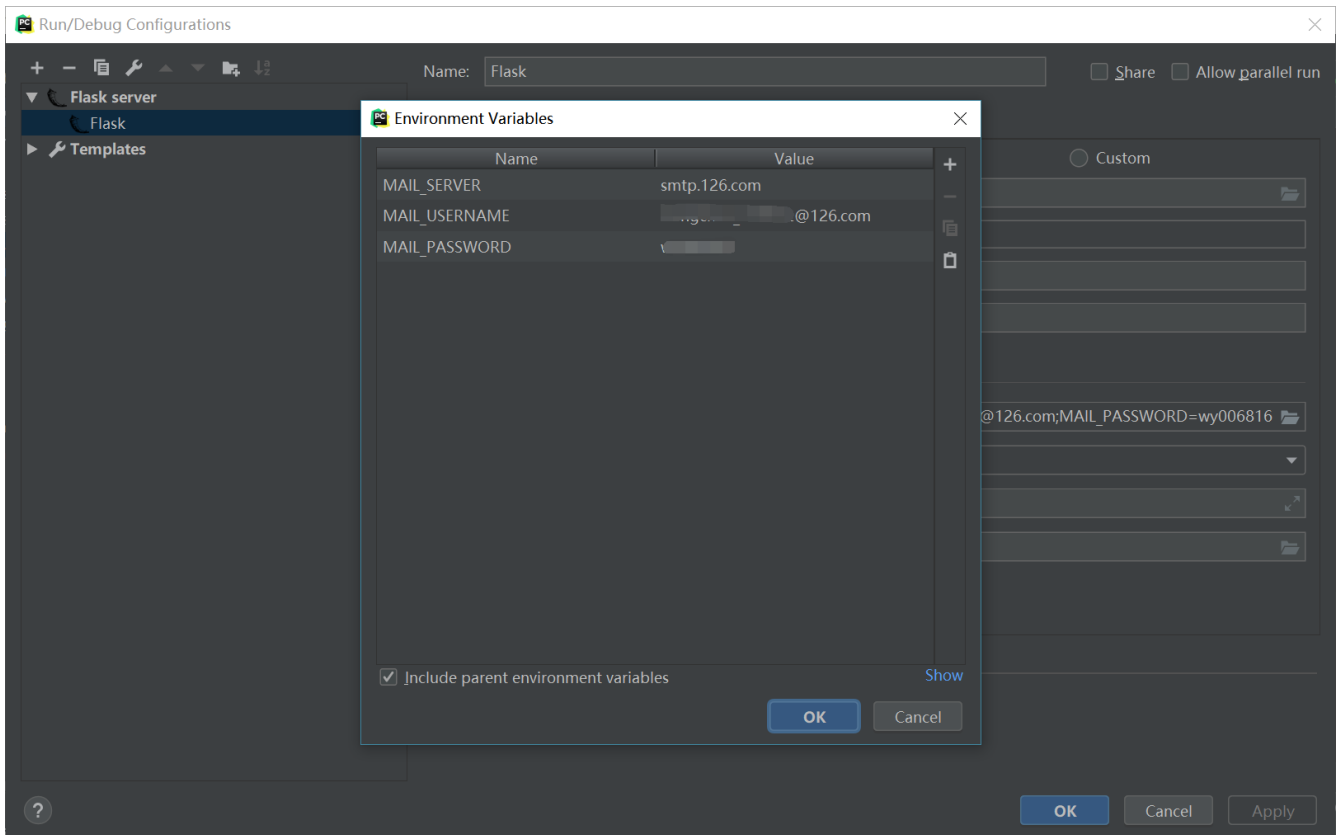
修改 `app/__init__.py` 脚本, 增加 `RotatingFileHandler` 日志文件记录器注册。

```
def create_app():
    .....
    # 非DEBUG模式下, 异常日志通过电子邮件发送
    if not application.debug:
        from app.logger import init_email, init_logger
        # 异常日志邮件提醒初始化
        init_email(application)
        # 日志记录器初始化
        init_logger(application)
    .....
    return application
```

3.6. pycharm配置环境变量测试邮件发送

修改pycharm中启动服务时的环境变量。





博客: [首页](#) [个人资料](#) [退出](#)

发生意外错误

已通知管理员, 给您带来的不便, 请多多原谅

[返回首页](#)

[« 返回](#) [回复](#) [回复全部](#) [转发](#) [删除](#) [彻底删除](#) [举报](#) [拒收](#) [标记为...](#) [移动到...](#) [上一封](#) [下一封](#)

flask-mega-tutorial 博客异常 ☆
发件人: <@126.com>
时间: 2019年4月28日(星期天) 下午4:22
收件人: @qq.com>
[纯文本](#) | [打印](#) [分享](#) [收藏](#) [举报](#)

Exception on /user/edit [POST]
Traceback (most recent call last):
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\sqlalchemy\engine\base.py", line 1244, in _execute_context
cursor, statement, parameters, context
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\sqlalchemy\engine\default.py", line 552, in do_execute
cursor.execute(statement, parameters)
sqlite3.IntegrityError: UNIQUE constraint failed: users.username

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 2292, in wsgi_app
response = self.full_dispatch_request()
File "D:\Projects\learn\flask-mega-tutorial\venv\lib\site-packages\flask\app.py", line 1815, in full_dispatch_request
rv = self.handle_user_exception(e)

4. 解决该异常情况

4.1. 表单增加已使用用户名校验

修改 app/forms.py 脚本, 增加用户名验证函数。实例化用户资料编辑表单时, 增加原用户名字段, 用于区分是否做出用户名修改。同时根据新的用户名判断该用户是否已被使用。

```
class UserInfoEditForm(FlaskForm):
    """用户信息编辑表单"""
    username = StringField('用户名', validators=[DataRequired()])
    about_me = TextAreaField('个人简介', validators=[Length(min=0, max=140)])
    submit = SubmitField('提交')

    def __init__(self, original_username, *args, **kwargs):
        super(UserInfoEditForm, self).__init__(*args, **kwargs)
        self.original_username = original_username

    def validate_username(self, username):
        if username.data != self.original_username:
```

```
user = User.query.filter_by(username=self.username.data).first()
if user is not None:
    raise ValidationError('请使用其他用户名')
```

4.2. 实例化表单时传入原用户名

修改 `app/user.py` 脚本，初始化表单时传入当前用户名。

```
class UserInfoEditView(View):
    """用户资料编辑"""
    methods = ['GET', 'POST']
    decorators = [login_required]

    def dispatch_request(self):
        form = UserInfoEditForm(current_user.username)
        .....
```

4.3. 启动服务测试

← → ↻ ⓘ 127.0.0.1:5000/user/edit

博客: [首页](#) [个人资料](#) [退出](#)

个人资料编辑

用户名

admin1

[请使用其他用户名]

个人简介

这是测试信息

提交