

[登录](#)

2019年04月19日 阅读 302

[关注](#)

Flask框架从入门到精通之参数配置(二)

知识点：1、参数配置

一、概况

上一篇我们已经把 **Flask** 第一个程序运行起来了,那么这一篇主要讲一下 **Flask** 参数的配置。

二、配置参数

Flask 参数配置方式有很多种,每一种都可以达到结果,在合适的场景选择合适的配置方式。

- 配置文件 在文件加中新建一个 **config.cfg** 配置文件,在配置文件上写上下面语句,表示开启调试模式。这样当我们程序出错的时候,就能知道错误在多少行,方便我们快速定位错误。

```
DEBUG = True
```

[复制代码](#)

在程序中加载配置文件

```
from flask import Flask

'''
__name__:表示当前的模块名字
创建Flask对象, Flask会以传入模块的位置当做家目录
'''
app = Flask(__name__)

@app.route('/') # 代表首页
def hello_world(): # 视图函数
    print(1/0)
    return 'Hello World!' # 返回内容

# 配置文件
app.config.from_pyfile('config.cfg')

if __name__ == '__main__':
    app.run() # 运行程序
```

[复制代码](#)

我们可以看出,这个程序是有错的,因为1/0在生活中,0是不能当做除数,所以会出现异常。当我们在浏览器访问的时候,可以看到

builtins.ZeroDivisionError

ZeroDivisionError: division by zero

Traceback (most recent call last)

```
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 2309, in __call__
    return self.wsgi_app(environ, start_response)
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 2295, in wsgi_app
    response = self.handle_exception(e)
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 1741, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/_compat.py", line 35, in reraise
    raise value
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 2292, in wsgi_app
    response = self.full_dispatch_request()
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 1815, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 1718, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/_compat.py", line 35, in reraise
    raise value
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 1813, in full_dispatch_request
    rv = self.dispatch_request()
File "/Users/xiaoyuan/.virtualenvs/flask_1.0/lib/python3.5/site-packages/flask/app.py", line 1799, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/Users/xiaoyuan/Desktop/study_flask/helloworld.py", line 12, in hello_world
    print(1/0)
```

• 类方式配置

#类方式配置

```
class Config(object):
    DEBUG = True
```

```
app.config.from_object(Config)
```

复制代码

• 直接操作配置 直接操作适用于配置参数较少的场景, config 本质其实就一个字典。

#直接操作

```
app.config["DEBUG"] = True
```

复制代码

• 对象上配置

```
app.debug = True
```

复制代码

• 参数传入配置 (只限debug参数)

```
app.run(debug=True) # 运行程序
```

复制代码

三、取配置参数

有很多情况, 我们会有需求, 想在视图函数中, 把配置参数取到并使用。这里面提供了两种拿到的办法:

```
from flask import Flask, current_app
```

```
'''
```

__name__:表示当前的模块名字

创建Flask对象, Flask会以传入模块的位置当做家目录

```
'''
```

```
app = Flask(__name__)
```

复制代码

```
@app.route('/') # 代表首页
def hello_world(): # 视图函数
    # 如果能拿到app对象,就可以通过app取自定义的配置参数
    print(app.config.get("TEST"))
    # 如果拿不到,可以通过current_app这个app的代理人拿到
    print(current_app.config.get("TEST"))
    return 'Hello World!' # 返回内容

# 直接操作
app.config["DEBUG"] = True
app.config["TEST"] = 'test'

if __name__ == '__main__':
    app.run() # 运行程序
```

支持外部访问

我们修改调用 `run()` 的方法使我们的服务器公开可用, 如下:

```
# 0.0.0.0代表任何能代表这台机器的地址都可以访问
app.run(host='0.0.0.0', port=5000) # 运行程序
```

复制代码

这个时候我们可以通过 <http://127.0.0.1:5000/> 可以访问我们的网站, 也可以通过自己的本机ip去访问 <http://192.168.1.101:5000/>。可以通过 `ifconfig` 命令查看自己的本机IP。

欢迎关注我的公众号：



文章分类

[前端](#)

[后端](#)

[数据库](#)

文章标签

[Python](#)

[Flask](#)



相关推荐

