

[登录](#)

2019年05月04日 阅读 904

[关注](#)

Flask框架从入门到精通之请求钩子(二十二)

知识点: 1、请求钩子

一、概况

Flask的请求钩子指的是在执行视图函数前后执行的一些函数,我们可以在这些函数里面做一些操作。Flask利用装饰器给我们提供了四种钩子函数。

- `before_first_request`:在处理第一个请求前执行。比如链接数据库操作
- `before_request`:在每次请求前执行。比如权限校验
- `after_request`:每次请求之后调用,前提是没有未处理的异常抛出
- `teardown_request`:每次请求之后调用,即使有未处理的异常抛出

Flask里面的钩子跟Django里面的中间件类似。

二、使用

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    print('视图函数执行')
    return 'index page'

# 在第一次请求之前运行.
@app.before_first_request
def before_first_request():
    print('before_first_request')

# 在每一次请求前都会执行
@app.before_request
def before_request():
    print('before_request')

# 在请求之后运行
@app.after_request
```

[复制代码](#)

```
def after_request(response):
    # response: 就是前面的请求处理完毕之后, 返回的响应数据, 前提是视图函数没有出现异常
    # 如果需要对响应做额外处理, 可以再这里进行
    # json.dumps 配置请求钩子
    # response.headers["Content-Type"] = "application/json"
    print('after_request')
    return response

# 无论视图函数是否出现异常, 每一次请求之后都会调用, 会接受一个参数, 参数是服务器出现的错误信息
@app.teardown_request
def teardown_request(error):
    print('teardown_request: error %s' % error)

if __name__ == '__main__':
    app.run(debug=True)
```

我们在浏览器第一次访问：

```
before_first_request
before_request
视图函数执行
after_request
teardown_request: error None
```

[复制代码](#)

我们在刷新浏览器试一下：

```
before_request
视图函数执行
after_request
teardown_request: error None
```

[复制代码](#)

欢迎关注我的公众号：



文章分类

前端

后端



文章标签



Flask



相关推荐

