**2.6Flask基础知识·Flask全局请求拦截**

所有进入路由方法前，都优先经过的处理方法，在Python里面叫装饰器，Flask里面已经有实现，可直接使用。

# 作用

拦截用户的请求，对请求做预先的判断处理工作，如登录拦截、权限拦截等。

# 修改app.py文件

from flask import Flask, request  
  
from controllers.role\_controller import role  
from controllers.user\_controller import user  
  
app = Flask(\_\_name\_\_)  
  
# 注册用户模块  
app.register\_blueprint(user)  
# 注册角色模块  
app.register\_blueprint(role)  
  
  
@app.before\_request  
def auth():  
 WHITE\_LIST = ['/sys/user/detail', '/sys/user/page']  
 # 简单处理一下，非白名单路由，提示登录  
 if request.path not in WHITE\_LIST:  
 return "请先登录"  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(host="0.0.0.0", port=5000)

# 运行Flask服务

flask run

# 使用curl命令访问

## 访问/sys/user/detail

curl -X POST http://localhost:5000/sys/user/detail

## 访问/sys/user/page

curl -X POST http://localhost:5000/sys/user/page

## 访问/sys/user/save

curl -X POST http://localhost:5000/sys/user/save

## 访问/sys/user/update

curl -X POST http://localhost:5000/sys/user/update

## 访问/sys/user/remove

curl -X POST http://localhost:5000/sys/user/remove

# 使用ApiFox/Postman接口测试工具访问

略