**Flask\_线程隔离的g对象使用详解：**

保存为全局对象g对象的好处：

g对象是在整个Flask应用运行期间都是可以使用的。

并且也跟request一样，是线程隔离的。

这个对象是专门用来存储开发者自己定义的一些数据，方便在整个Flask程序中都可以使用。

一般使用就是，将一些经常会用到的数据绑定到上面，以后就直接从g上面取就可以了，而不需要通过传参的形式，这样更加方便。

**g对象使用场景：**有一个工具类utils.py 和 用户办理业务：

|  |
| --- |
| **def** funa(uname):  print(**'funa %s'** % uname)  **def** funb(uname):  print(**'funb %s'** % uname)  **def** func(uname):  print(**'func %s'** % uname) |

用户办理业务：

|  |
| --- |
| **from** flask **import** Flask,request  **from** utils **import** funa,funb,func  app = Flask(\_\_name\_\_)  *#Flask\_线程隔离的g对象使用详解*  @app.route(**"/profile/"**)  **def** my\_profile():  *#从url中取参*  uname = request.args.get(**'uname'**)  *#调用功能函数办理业务*  funa(uname)  funb(uname)  func(uname)  *#每次都得传参 麻烦，引入g对象进行优化*  **return "办理业务成功"**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

**优化工具类utils.py：**

|  |
| --- |
| **from** flask **import** g  **def** funa():  print(**'funa %s'** % g.uname)  **def** funb():  print(**'funb %s'** % g.uname)  **def** func():  print(**'func %s'** % g.uname) |

**优化用户办理业务：**

|  |
| --- |
| **from** flask **import** Flask,request,g  **from** utils **import** funa,funb,func  app = Flask(\_\_name\_\_)  *#Flask\_线程隔离的g对象使用详解*  @app.route(**"/profile/"**)  **def** my\_profile():  *#从url中取参*  uname = request.args.get(**'uname'**)  *#调用功能函数办理业务*  *# funa(uname)*  *# funb(uname)*  *# func(uname)*  *#每次都得传参 麻烦，引入g对象进行优化*  g.uname = uname  funa()  funb()  func()  **return "办理业务成功"**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |