**Flask\_信号使用场景\_存储用户登录日志：**

**1.信号使用场景**

定义一个登录的信号，以后用户登录进来以后

就发送一个登录信号，然后能够监听这个信号

在监听到这个信号以后，就记录当前这个用户登录的信息

用信号的方式，记录用户的登录信息即登录日志

**2.编写一个signals.py文件创建登录信号**

|  |
| --- |
| **from** blinker **import** Namespace  **from** datetime **import** datetime  **from** flask **import** request,g  namespace = Namespace()  *#创建登录信号*  login\_signal = namespace.signal(**'login'**)  **def** login\_log(sender):  *# 用户名 登录时间 ip地址*  now = datetime.now()  ip = request.remote\_addr  log\_data = **"{uname}\*{now}\*{ip}"**.format(uname=g.uname, now=now, ip=ip)  **with** open(**'login\_log.txt'**,**'a'**) **as** f:  f.write(log\_data + **"\n"**)  f.close()  *#监听信号*  login\_signal.connect(login\_log) |

**3.使用信号存储用户登录日志**

|  |
| --- |
| **from** flask **import** Flask,request,g  **from** signals **import** login\_signal  app = Flask(\_\_name\_\_)  @app.route(**'/login/'**)  **def** login():  # 通过查询字符串的形式来传递uname这个参数  uname = request.args.get(**'uname'**)  **if** uname:  g.uname = uname  # 发送信号  login\_signal.send()  **return '登录成功！'**  **else**:  **return '请输入用户名！'**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |