**Flask\_内置信号:**

flask内置了10个常用的信号。

**1.** template\_rendered：模版渲染完成后的信号。

**2.** before\_render\_template：模版渲染之前的信号。

**3.** request\_started：请求开始之前，在到达视图函数之前发送信号。

**4.** request\_finished：请求结束时，在响应发送给客户端之前发送信号。

**5.** request\_tearing\_down：请求对象被销毁时发送的信号，即使在请求过程中发生异常也会发送信号。

**6.** got\_request\_exception：在请求过程中抛出异常时发送信号，异常本身会通过exception传递到订阅（监听）的函数中。一般可以监听这个信号，来记录网站异常信息。

**7.** appcontext\_tearing\_down：应用上下文被销毁时发送的信号。

**8.** appcontext\_pushed：应用上下文被推入到栈上时发送的信号。

**9.** appcontext\_popped：应用上下文被推出栈时发送的信号。

**10.** message\_flashed：调用了Flask的`flash`方法时发送的信号。

**代码演示：**

template\_rendered的使用：

|  |
| --- |
| **from** flask **import** Flask,request,g,template\_rendered,got\_request\_exception,render\_template  app = Flask(\_\_name\_\_)  *#内置信号*  *#模版渲染完成后的信号。*  **def** template\_rendered\_func(sender,template,context):  print(sender) *#发送者*  print(template) *#跳转到的模版名称*  print(context) *#跳转到模版时带过去的参数*  template\_rendered.connect(template\_rendered\_func)  @app.route(**'/'**)  **def** hello\_world():  **return** render\_template(**"index.html"**,data=**"momo"**)#去看发送信号的底层  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

got\_request\_exception的使用：

|  |
| --- |
| **from** flask **import** Flask,request,g,template\_rendered,got\_request\_exception,render\_template  app = Flask(\_\_name\_\_)  #内置信号  #got\_request\_exception：在请求过程中抛出异常时发送信号，异常本身会通过exception传递到订阅（监听）的函数中。  # 一般可以监听这个信号，来记录网站异常信息。  # def request\_exception\_log(sender,\*args,\*\*kwargs): #掌握写参数技巧  # print(sender)  # print(args)  # print(kwargs)  **def** request\_exception\_log(sender,exception):  print(sender)  print(exception) # division by zero  got\_request\_exception.connect(request\_exception\_log)  @app.route(**'/'**)  **def** hello\_world():  #制造bug  a = 1/0  **return** render\_template(**"index.html"**,data=**"momo"**)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |