

02-Flask会话技术和Flask模板语言

会话技术

Cookie

客户端的会话技术

cookie本身由浏览器保存，通过Response将cookie写到浏览器上，下一次访问，浏览器会根据不同的规则携带cookie过来

特点：

- 客户端会话技术，浏览器的会话技术
- 数据全都是存储在客户端中
- 存储使用的键值对结构进行的存储
- 特性
 - 支持过期时间
- 默认会自动携带本网站的所有cookie
 - 根据域名进行cookie存储
 - 不能跨域名
 - 不能跨浏览器
- Cookie是通过服务器创建的Response来创建的

设置cookie：

```
response.set_cookie(key,value[,max_age=None,expires=None])
```

max_age：整数，指定cookie过期时间
expires：整数，指定过期时间，可以指定一个具体日期时间
max_age和expires两个选一个指定

获取cookie：

```
request.cookies.get(key)
```

删除cookie

```
response.delete_cookie(key)
```

session

服务器端会话技术，依赖于cookie

特点：

- 服务端的会话技术
- 所有数据存储在服务器中
- 默认存储在内存中
- 存储结构也是key-value形势，键值对
- session 是离不开cookie的

Flask中的session是全局对象

(之前的request也是Flask的一个全局对象)

常用操作:

设置session

```
session['key'] = 'value'
```

获取session

```
session.get(key, default=None) 根据键获取会话的值
```

删除session

```
session.pop(key) 删除某一值
```

```
session.clear() 清除所有
```

cookie和session的区别:

cookie:

1. 在浏览器存储
2. 安全性较低
3. 可以减轻服务器压力

session:

1. 在服务器端存储
2. 安全性高
3. 对服务器要求较高
4. 依赖cookie

模板Template

模板是呈现给用户的界面

在MVT中充当T的角色, 实现了MT的解耦, 开发中vT有这N:M的关系, 一个v可以调用任意T, 一个T可以被任意v调用
模板处理分为两个过程

1. 加载HTML
2. 模板渲染(模板语言)

模板代码包含两个部分

1. 静态HTML
2. 动态插入的代码段(模板语法)

Jinja2

Flask中使用Jinja2模板引擎

Jinja2由Flask作者开发

一个现代化设计和友好的Python模板语言

模仿Django的模板引擎

优点

速度快，被广泛使用

HTML设计和后端Python分离

减少Python复杂度

非常灵活，快速和安全

提供了控制，继承等高级功能

模板语法

模板语法主要分为两种

变量

标签

模板中的变量 `{{ var }}`

视图传递给模板的数据

前面定义出来的数据

变量不存在，默认忽略

模板中的标签 `{% tag %}`

控制逻辑

使用外部表达式

创建变量

宏定义

结构标签

block 块操作

父模板挖坑，子模板填坑

```
{% block xxx %}
```

```
{% endblock %}
```

extends 继承

```
{% extends 'xxx' %}
```

继承后保留块中的内容

```
{{ super() }}
```

include

包含，将其他html包含进来

```
{% include 'xxx' %}
```

marco 【了解】

宏定义，可以在模板中定义函数，在其它地方调用

```
{% macro hello(name) %}  
    {{ name }}  
{% endmacro %}
```

宏定义可导入

```
{% from 'xxx' import xxx %}
```

循环

for循环

```
{% for item in cols %}  
    AA  
{% else %}  
    BB  
{% endfor %}
```

可以使用和Python一样的for...else

也可以获取循环信息 loop

loop.first: 判断是否是第一个元素

loop.last: 判断是否是最后一个元素

loop.index: 1开始的下标

loop.index0: 0开始的下标

loop.revindex: 反向下标,不包括0

loop.revindex0: 反向下标,包括0

过滤器（扩展）

语法

```
{{ 变量 | 过滤器 | 过滤器... }}
```

capitalize

lower

upper

title

trim

reverse

striptags 渲染之前，将值中标签去掉

safe

default(1)

last

first

length

sum

sort

...

掌握

1. 熟练掌握会话技术Cookie和Session
2. 能轻松写出基于Cookie和Session的登录功能
3. 掌握模板语法