

Flask Session与Cookie

一. Flask设置与使用Cookie

1.1 Flask设置Cookie, Flask在HTTP响应中为客户端设置Cookie. 示例代码:

```
In [ ]:
from flask import Flask, request
from flask import make_response
app = Flask(__name__)
@app.route('/set_cookie')
def get_name():
    resp = make_response('cookie setup success!')
    resp.set_cookie("username", "the value")
    return resp
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8902)
```

测试:
请求URL: http://172.16.70.233:8902/set_cookie, 并在chrome浏览器中检查设置的cookies

←
172.16.70.233 本地存储的数据

全部删除

Hm_lvt_080836300300be57b7f34f4b3e97d911	▼	×
username	^	×
<div>名称</div> <div>username</div>		
<div>内容</div> <div>"the value"</div>		
<div>域名</div> <div>172.16.70.233</div>		

1.2 客户端浏览器会在每次的HTTP请求中把与该网站相关的所有cookie信息发送给WEB服务器, 在服务器端可以通过request.cookies来查看, 示例代码:

In [1]:

```

from flask import Flask, request
from flask import make_response
app = Flask(__name__)
@app.route('/get_cookie')
def get_person():
    s = request.cookies
    print(s)
    return 'Look up flask shell'
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8902)

```

* Running on http://0.0.0.0:8902/ (http://0.0.0.0:8902/) (Press CTRL+C to quit)

测试:

请求URL: http://172.16.70.233:8902/get_cookie, 最终可以在Flask的SHELL中查看到HTTP请求的cookies信息.

```

(proj_b) [root@backup-platform web]# python demo10.py
* Serving Flask app "demo10" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8902/ (Press CTRL+C to quit)
{'Hm_lvt_080836300300be57b7f34f4b3e97d911': '1562470445,1563173641,1563173657', 'username': 'he value'}
172.16.70.1 - - [30/Jul/2019 02:27:52] "GET /get_cookie HTTP/1.1" 200 -

```

二. Flask中应用Session

2.1 session模块安装

```
pip install Flask-Session -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com
```

2.2 Flask设置Session

In []:

```

from flask import session, Flask, request
from flask import make_response
from flask_session import Session
app = Flask(__name__)
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_FILE_DIR'] = 'session'
Session(app)
@app.route('/set_session')
def get_name():
    session["foo"] = "bar"
    return 'session set success!'
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8902)

```

2.3 Flask获取Session

In []:

```

from flask import session, Flask, request
from flask import make_response
from flask_session import Session
app = Flask(__name__)
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_FILE_DIR'] = 'session'
Session(app)
@app.route('/get_session')
def get_person():
    print(session)
    return 'Show flask shell'
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8902)

```

```

(proj_b) [root@backup-platform web]# python demo11.py
* Serving Flask app "demo11" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8902/ (Press CTRL+C to quit)
<FileSystemSession {'_permanent': True, 'foo': 'bar'}>
172.16.70.1 - - [30/Jul/2019 02:57:10] "GET /get_session HTTP/1.1" 200 -

```

在浏览器端查看sessionid，在随后的HTTP请求中与该网站相关的所有cookie信息都会随着HTTP请求发送到WEB服务器，服务器查看到了cookie中的sessionid信息，会通过查询自身所维护的Session表，找到与该ID相对应的位于本地磁盘中的session文件。

← 172.16.70.233 本地存储的数据

全部删除

Hm_lvt_080836300300be57b7f34f4b3e97d911



session



名称

session

内容

56b0349c-282b-4980-af9b-b36df96530d4

In []: