Flask Cookies

The cookies are stored in the form of text files on the client's machine. Cookies are used to track the user's activities on the web and reflect some suggestions according to the user's choices to enhance the user's experience.

Cookies are set by the server on the client's machine which will be associated with the client's request to that particular server in all future transactions until the lifetime of the cookie expires or it is deleted by the specific web page on the server.

In flask, the cookies are associated with the Request object as the dictionary object of all the cookie variables and their values transmitted by the client. Flask facilitates us to specify the expiry time, path, and the domain name of the website.

from flask import \*  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/cookie')  
def cookie():  
 res = make\_response("<h1>cookie is set</h1>")  
 res.set\_cookie('foo', 'bar')  
 return res  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

## Login application in Flask

Login.py

from flask import \*  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/error')  
def error():  
 return "<p><strong>Enter correct password</strong></p>"  
  
  
@app.route('/')  
def login():  
 return render\_template("login.html")  
  
  
@app.route('/success', methods=['POST'])  
def success():  
 if request.method == "POST":  
 email = request.form['email']  
 password = request.form['pass']  
  
 if password == "jtp":  
 resp = make\_response(render\_template('success.html'))  
 resp.set\_cookie('email', email)  
 return resp  
 else:  
 return redirect(url\_for('error'))  
  
  
@app.route('/viewprofile')  
def profile():  
 email = request.cookies.get('email')  
 resp = make\_response(render\_template('profile.html', name=email))  
 return resp  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

login.html

<html>   
<head>   
 <title>login</title>   
</head>   
<body>   
 <form method = "post" action = "http://localhost:5000/success">   
 <table>   
 <tr><td>Email</td><td><input type = 'email' name = 'email'></td></tr>   
 <tr><td>Password</td><td><input type = 'password' name = 'pass'></td></tr>   
 <tr><td><input type = "submit" value = "Submit"></td></tr>   
 </table>   
 </form>   
</body>   
</html>

Success.html

<html>   
<head>   
<title>success</title>   
</head>   
<body>   
 <h2>Login successful</h2>   
 <a href="/viewprofile">View Profile</a>   
</body>   
</html>

Profile.html

<html>   
<head>   
 <title>profile</title>   
</head>   
<body>   
 <h3>Hi, {{name}}</h3>   
</body>   
</html>

# Flask Session

The concept of a session is very much similar to that of a cookie. However, the session data is stored on the server.

The session can be defined as the duration for which a user logs into the server and logs out. The data which is used to track this session is stored into the temporary directory on the server.

The session data is stored on the top of cookies and signed by the server cryptographically.

In the flask, a session object is used to track the session data which is a dictionary object that contains a key-value pair of the session variables and their associated values.

The following syntax is used to set the session variable to a specific value on the server.

Session[<variable-name>] = <value>

To remove a session variable, use the pop() method on the session object and mention the variable to be removed.

session.pop(<variable-name>, none)

from flask import \*  
  
app = Flask(\_\_name\_\_)  
app.secret\_key = "abc"  
  
  
@app.route('/')  
def home():  
 res = make\_response("<h4>session variable is set, <a href='/get'>Get Variable</a></h4>")  
 session['response'] = 'session#1'  
 return res;  
  
  
@app.route('/get')  
def getVariable():  
 if 'response' in session:  
 s = session['response'];  
 return render\_template('getsession.html', name=s)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

getsession.html

<html>   
<head>   
<title>getting the session</title>   
</head>   
<body>   
<p>The session is set with value: <strong>{{name}}</strong></p>   
</body>   
</html>

Login.py

from flask import \*  
  
app = Flask(\_\_name\_\_)  
app.secret\_key = "ayush"  
  
  
@app.route('/')  
def home():  
 return render\_template("home.html")  
  
  
@app.route('/login')  
def login():  
 return render\_template("login.html")  
  
  
@app.route('/success', methods=["POST"])  
def success():  
 if request.method == "POST":  
 session['email'] = request.form['email']  
 return render\_template('success.html')  
  
  
@app.route('/logout')  
def logout():  
 if 'email' in session:  
 session.pop('email', None)  
 return render\_template('logout.html');  
 else:  
 return '<p>user already logged out</p>'  
  
  
@app.route('/profile')  
def profile():  
 if 'email' in session:  
 email = session['email']  
 return render\_template('profile.html', name=email)  
 else:  
 return '<p>Please login first</p>'  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

home.html

<html>   
<head>   
<title>home</title>   
</head>   
<body>   
<h3>Welcome to the website</h3>   
<a href = "/login">login</a><br>   
<a href = "/profile">view profile</a><br>   
<a href = "/logout">Log out</a><br>   
</body>   
</html>

Login.html

<html>   
<head>   
 <title>login</title>   
</head>   
<body>   
 <form method = "post" action = "http://localhost:5000/success">   
 <table>   
 <tr><td>Email</td><td><input type = 'email' name = 'email'></td></tr>   
 <tr><td>Password</td><td><input type = 'password' name = 'pass'></td></tr>   
 <tr><td><input type = "submit" value = "Submit"></td></tr>   
 </table>   
 </form>   
</body>   
</html>

Success.html

<html>   
<head>   
<title>success</title>   
</head>   
<body>   
 <h2>Login successful</h2>   
 <a href="/profile">View Profile</a>   
</body>   
</html>

Logout.html

< html >  
< head >  
< title > logout < / title >  
< / head >  
  
< body >  
< p > logout  
successful, click < a  
href = "/login" > here < / a > to  
login  
again < / p >  
< / body >  
  
< / html >