Sessions

In this flask tutorial I show you how to use the HTTP request methods Post and Get. The POST method will allow us to retrieve data from forms on our web page. In later videos we'll get into more advanced topics relating to login sessions and using POST methods to retrieve secure information like passwords.

Sessions vs Cookies

You may have heard of a web feature called *cookies*. I'd just like to quickly explain the difference between a cookie and a session to clear up any confusion.

Cookie: Is stored client side (locally in the web browser) and is NOT a secure way to store sensitive information like passwords. It is often used to store things like where a user left on a page, or their username so it can be auto-filled in the next time they visit the page. It is technically possible for someone to modify cookie data.

Session: Is stored server side (on the web server) in a temporary directory. It is encrypted information and is a secure way to store information. Sessions are often used to store information that should not be seen by the user or should not be tampered with.

Setting up a Session

Create new python file called ***tutorial5.py***

tutorial5.py

```
from flask import Flask, redirect, url_for, render_template,  request

app = Flask(__name__)

@app.route("/")
def home():
   return render_template("index.html")

@app.route("/login", methods=["POST", "GET"])
def login():
   if request.method  == "POST":
      user = request.form["nm"]
      return redirect(url_for("user", usr=user))
   else:
      return render_template("login.html")

@app.route("/<usr>")
def user(usr):
   return f"<h1>{usr}</h1>"

if __name__  == "__main__":
   app.run(debug=True)
```

Setting up a Session

To illustrate how a session works we will walk through a basic example where a user logs in and their username will be stored in a session until they logout.

We will start by importing *session* from flask.

```
from flask import Flask, redirect, url_for, render_template, request, session
```

While we are at it we will import another module that we'll use later.

```
from datetime import timedelta
```

Since the session information will be encrypted on the server we need to provide flask with a *secret key* that it can use to encrypt the data.

At the top of our program we'll write the following:

```
app.secret_key = "hello"
```

Now we can start saving data!

Creating Session Data

Saving information to a session is actually pretty easy. Sessions are represented in python as dictionaries. This means we can access values using **keys**. To save a new value in a session just create a new dictionary key and assign it a value.

```
session["my key"] = "my value"
```

Getting information can then be done the following way:

```
if "my key" in session:
    my_value = session["my key"]
```

For this example we will save a user's username after they login. Then the next time they visit the login page we will see if they are logged in by checking their session data. If they are there is no need for them to login again and we can redirect them to the user page.

```
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session.permanent = True
        user = request.form["nm"]
        session["user"] = user
        return redirect(url_for("user"))
    else:
        if "user" in session:
            return redirect(url_for("user"))

        return render_template("login.html")
```

Creating Session Data

Now from our /user page we can display the users name by simply grabbing the information from the session. If they have not signed in yet we will see that they have no username in their session and we can redirect them to the login page.

```
@app.route("/user")
def user():
   if "user"  in session:
      user = session["user"]
      return f"<h1>{user}</h1>"
   else:
      return redirect(url_for("login"))
```

And finally time to code the logout page! When a user goes to /logout we need to clear their session data. To do this we can use a method called *session.pop("key", What to do if key doesn't exist)*. The pop method will try to remove and return the key from the session data and will return the second argument if that key doesn't exist. In our case we try to remove the key "user" and if it doesn't exist we will return None.

```
@app.route("/logout")
def logout():
   session.pop("user",  None)
   return redirect(url_for("login"))
```

Session Duration

So now that we know how to create, add and remove data from sessions we should probably talk about how long they last. By default a session lasts as long as your browser is open. However, there is a way to change that from flask. We can set the duration of a session by creating a **permanent session**. Creating a permanent session allows us to define how long that session lasts. The default duration of a permanent session is 30 days.

We will start by defining the duration at the beginning of our program.

```
app.permanent_session_lifetime = timedelta(minutes=5)
```

In our example we’ve made our session last 5 minutes.

Next we will make the users session permanent as soon as they log in.

```
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session.permanent = True
        user = request.form["nm"]
        session["user"] = user
        return redirect(url_for("user"))
    else:
        if "user" in session:
            return redirect(url_for("user"))

        return render_template("login.html")
```

Session Duration

So now that we know how to create, add and remove data from sessions we should probably talk about how long they last. By default a session lasts as long as your browser is open. However, there is a way to change that from flask. We can set the duration of a session by creating a **permanent session**. Creating a permanent session allows us to define how long that session lasts. The default duration of a permanent session is 30 days.

We will start by defining the duration at the beginning of our program.

```
app.permanent_session_lifetime = timedelta(minutes=5)
```

In our example we've made our session last 5 minutes.

Next we will make the users session permanent as soon as they log in.

```
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session.permanent = True
        user = request.form["nm"]
        session["user"] = user
        return redirect(url_for("user"))
    else:
        if "user" in session:
            return redirect(url_for("user"))

        return render_template("login.html")
```

Full Python Code

tutorial5.py

```
from flask import Flask, redirect, url_for, render_template, request, session
from datetime import timedelta

app = Flask(__name__)
app.secret_key = "hello"
app.permanent_session_lifetime = timedelta(minutes=5)

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session.permanent = True
        user = request.form["nm"]
        session["user"] = user
        return redirect(url_for("user"))
    else:
        if "user" in session:
            return redirect(url_for("user"))

        return render_template("login.html")

@app.route("/user")
def user():
    if "user" in session:
        user = session["user"]
        return f"<h1>{user}</h1>"
    else:
        return redirect(url_for("login"))

@app.route("/logout")
def logout():
    session.pop("user", None)
    return redirect(url_for("login"))

if __name__ == "__main__":
    app.run(debug=True)
```