

Advanced Programming with Python

sessions and cookies

Pepe García jgarciah@faculty.ie.edu



SCHOOL OF
SCIENCE &
TECHNOLOGY

Plan for today

- Review last day's homework
- HTML Forms
- Sessions and Cookies
- Authentication

Forms in HTML are pieces of the UI which we use to send data from client to server.

Forms in HTML are pieces of the UI which we use to send data from client to server.

```
<form action="/handle-login" method="POST">
```

```
</form>
```

Forms in HTML are pieces of the UI which we use to send data from client to server.

```
<form action="/handle-login" method="POST">  
    <input type="text" name="username"/>  
  
</form>
```

Forms in HTML are pieces of the UI which we use to send data from client to server.

```
<form action="/handle-login" method="POST">  
    <input type="text" name="username"/>  
    <input type="password" name="password"/>  
  
</form>
```

Forms in HTML are pieces of the UI which we use to send data from client to server.

```
<form action="/handle-login" method="POST">
    <input type="text" name="username"/>
    <input type="password" name="password"/>
    <input type="submit"/>
</form>
```

Whenever a form is submitted, it will send an HTTP requests to the URL specified in `action`, with the method specified in `method`, and sending the form data in the body of the request

Receiving form data in flask

```
from flask import request

@app.route("/handle-login", method=["POST"])
def handle_login():
    username = request.form["username"]
    password = request.form["password"]

    # Now, do whatever you want with user and password, for ex
    # new session for the user
```



Adding login to Twitter for dogs!

Adding login to Twitter for dogs!

Let's add login functionality to Twitter for dogs!!

There's a feature of most web applications that we haven't yet discussed, **sessions**.

Sessions help us authenticate users given some piece of data in their request. This piece of data can be:

- A cookie
- An API token
- A JWT
- ...

There's a feature of most web applications that we haven't yet discussed, **sessions**.

Sessions help us authenticate users given some piece of data in their request. This piece of data can be:

- **A cookie** - An API token - A JWT - ...

Sessions can be used to authenticate users by relying on cookies

Cookies are headers that the server sends alongside the HTTP response, that the client **will send back** in subsequent requests to that host!

Whiteboard

Let's whiteboard the whole flow of cookies and sessions

Using sessions in flask

sessions in flask are handled by importing the session object from flask:

```
from flask import request, session
```

```
@app.route("/handle-login", method=["POST"])
```

```
def handle_login():
```

```
    username = request.form["username"]
```

```
    password = request.form["password"]
```

The session object behaves like a dictionary

```
session["username"] = username # this way, we can identify  
# the user
```



Using sessions in flask

We also use the session object to log the user out.

```
from flask import session, redirect
```

```
@app.route("/logout")
```

```
def handle_logout():
```

```
    # removing a session object by their key, like in a  
    # normal dictionary
```

```
    session.pop("username")
```

```
    return redirect("/")
```



Using sessions in flask

sessions in flask are handled by importing the session object from flask:

```
from flask import session
```

```
@app.route("/")
```

```
def index():
```

```
    if "user_id" in session: # session behaves like a dictionary
```

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

```
    else:
```

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



Signing users up

make it possible to create new accounts in barker. When creating an account, you must ask for user and password, and hash the password before storing it.