

Flask : Sessions again

Ref : pythonise.com/series/learning-flask/

flask-session-object

Session object : → store an encoded

cookie in user's browser

uses a secret key you provide to encode --
→ decode cookie on every request

one way to get a secret key

```
import secrets
```

```
secrets.token_urlsafe(16)
```

any number

→ random string which you can use as secret key

```
from flask import render_template,
```

```
request, session, redirect, url_for
```

```
app.config["SECRET_KEY"] = random_string
```

(Flask will use this to encode session object)

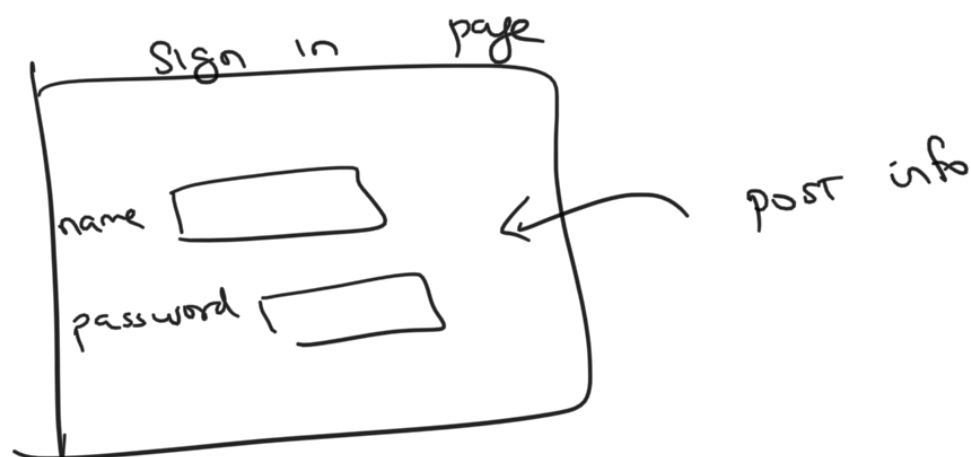
```
users = {
```

```
    "aaa" : { "name" : "aaa"
```

dummy database

```
    "bbb": {
      "name": "bbb"
      "password": "0000"
      "other": "x..."
    }
  }

  "password" : "..."
  "other" : ...
```



```
@ app.route ("/sign-in", methods = ["GET", "POST"]
```

```
def sign-in ():
```

```
    get username, password
```

```
    check they match, else
```

```
    return redirect(request.url)
```

returns to the url from which request originated

```
    suppose valid user + password
    ... enter sessions!
```

```
    user = users [username]
```

dictionary itself

```
    session ["USERNAME"] = user ["name"]
```

↑
whatever you store in
session variable, gets encoded
and posted as a cookie in
client's web browser

This is not secure, can be easily
decoded, so for instance, don't store
passwords in session variables!
might even just try to store
unique-id of database instead of
user name in sessions---

```
return redirect(url_for("profile"))
```

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


```
def profile():
```

```
if session.get("USERNAME", None) is  
not None:
```

```
username = session.get("USERNAME")
```

```
user = users[username]
```

```
return render_template("public/  
profile.html",  
user=user)
```



```
mydict.get(key, default_val)
```

$$= \begin{cases} \text{mydict[key]} & \text{if key in mydict} \\ \text{default_val} & \text{else} \end{cases}$$

Sign-out

pop username from sessions!

session = pop(key, None)