COMP1531

4.2 - Auth, State

Iteration 2 Topics

- **State**: Today
- **Testing with state**: Today
- **Authentication**: Today
- **Authorisation**: Today
- **Handling errors**: Today
- **Email & Reset**: Next week
- **Timers**: Next week

State

Let's look at **point.py** and **pointutil.py** together:

- Splitting up Flask wrapper from functions
- Using a "getData" method

State (Testing)

Let's look at **point.py** and **pointutil.py** together:

What problems with state will we run into when we test this?

Auth vs Auth

Authentication: Process of verifying the identity of a user

Authorisation: Process of verifying an identity's access privileges

Authentication

Naive method:

- User registers, we store their password
- When user logs in, we compare their input password to their stored password

Let's observer *auth.py*

(found in 19T3-lectures)

Authentication

What's wrong with this?

Authentication

Using **hashlib** to create a hash

hash.py

```python
import hashlib
print("mypassword")
print("mypassword".encode())
print(hashlib.sha256("mypassword".encode()))
print(hashlib.sha256("mypassword".encode()).hexdigest())
```

Authentication

Now let's improve auth.py

Authorisation

What is a "token"?

A packet of data used to authorise the user.

What's the simplest token?

Authorisation

What is a "token"?

A packet of data used to authorise the user.

What's the simplest token?

A user's user id! It tells us who they are.

What's the issue with just passing around a raw user_id though?

Authorisation

What is a "token"?

A packet of data used to authorise the user.

What's the simplest token?

A user's user id! It tells us who they are.

What's the issue with just passing around a raw user_id though?

Authentication can be **faked**

What is a JWT?

"JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties."

They are lightweight ways of encoding and decoding private information via a secret

Play around:

https://jwt.io/

Let's practice with python

Using a JWT in python:

https://pyjwt.readthedocs.io/en/latest/

```python
import jwt

SECRET = 'sempai'

encoded_jwt = jwt.encode({'some': 'payload'}, SECRET, algorithm='HS256').decode('utf-8')
print(jwt.decode(encoded_jwt.encode('utf-8'), SECRET, algorithms=['HS256']))
```

Let's practice with python

Now let's improve auth.py

ValueError and AccessError

Let's look at **auth.py** and see how to handle errors