Advanced Programming with Python

REST APIs

Pepe García jgarciah@faculty.ie.edu

SCHOOL OF SCIENCE & TECHNOLOGY

Plan for today

- REST
- REST in the server with Flask
- consuming REST APIs wiht requests

SCHOOL OF SCIENCE & TECHNOLOGY

What's an API

From Wikipedia

An application programming interface is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification. Wikipedia

SCHOOL OF SCIENCE & TECHNOLOGY

What's an API

Basically, a way of exposing a web server so that it can be used **from other software components**.

In order to be used from other software components, they must use a data interchange format. In our examples we'll use JSON.

SCHOOL OF SCIENCE & TECHNOLOGY

Creating APIs in Flask

Example

Let's see an example of a REST API in `rest.py`.
run it and visit `/users`

What's REST

We say an API is a REST API when:

- It uses **URIs for resources**
- It's **stateless**
- It uses HTTP semantically (verbs, status codes...)

SCHOOL OF SCIENCE & TECHNOLOGY

What's REST. URIs for resources

Resources in your API are represented as URIs.

Different operations on them are represented with different HTTP methods on that URI.

- `POST` for creating a resource
- `GET` for reading a resource
- `PUT` for updating a resource
- `DELETE` for deleting a resource
  We call these 4 operations CRUD (Create, Read, Update, Delete)

Example URIs

```
http://api.pepe.com/games/1
http://aws.amazon.com/ec2/instances/0f9304a4-e4f8-4d95-acce-bc
```

SCHOOL OF SCIENCE & TECHNOLOGY

What's REST. Statelessness

This means that the HTTP Request must come with everything needed in order to fulfill it.

Let's see the impact statelessness has in our server applications

There shouldn't be any expected state left in the server waiting for the next request. Traditional session cookies are discouraged, since they exist in the server's memory, they're stateful.

SCHOOL OF SCIENCE & TECHNOLOGY

What's REST. Statelessness

We can authenticate requests in a stateless way, let's take a look at HTTP Basic Auth.

```
from flask import request

username = request.authorization["username"]
password = request.authorization["password"]
```

SCHOOL OF SCIENCE & TECHNOLOGY

What's REST. Statelessness

On the client side, we can leverage requests to use basicauth too:

```
from requests.auth import HTTPBasicAuth

response = requests.get(
    create_url("/users"),
    auth=HTTPBasicAuth("pepe", "password")
)
```

Under the hood, basic auth passes user and password encoded in base64 in an Authorization header.

SCHOOL OF SCIENCE & TECHNOLOGY

What's REST. Statelessness

What's the biggest con of stateless sessions? (passing user and pass per request)

SCHOOL OF SCIENCE & TECHNOLOGY

What's REST. Statelessness

What's the biggest con of stateless sessions? (passing user and pass per request)

We'll need to query the DB for user authentication on every request. JWTs try to solve this problem, but introduce some other tradeoffs.

SCHOOL OF SCIENCE & TECHNOLOGY

Consuming REST APIs with requests

All HTTP methods can be used with the **requests** library.

```
import requests

requests.get("http://localhost:8080/users")
requests.post("http://localhost:8080/users", json={"a": 1})
requests.put("http://localhost:8080/users", json={"a": 1})
requests.delete("http://localhost:8080/users")
```

SCHOOL OF SCIENCE & TECHNOLOGY

Creating REST APIs in Flask

Example

Let's go over creating new users in Tweeter together

SCHOOL OF SCIENCE & TECHNOLOGY

Creating REST APIs in Flask

Exercise

implement the `GET /users/<username>/tweets` endpoint, that returns a JSON list of all tweets by that user.

Creating REST APIs in Flask

Exercise

Create a endpoint in our API that allows creating a tweet

Alternatives to Flask to build REST APIs

Nowadays it's very common to use FastAPI instead of Flask for REST APIs. It's inspired by Flask, and has lots of interesting functionality, check it out!