PokéAPI Exercise

# Introduction

Build a Spring Boot Application that enables a user to make HTTP requests to the PokeAPI. The system must require a user to be created **before** the PokeAPI can be used and **importantly** the search history of these users must be recorded.

This system needs to show your knowledge of:

* Spring Boot quality coding and best practice.
* Testing with Spring Boot, at least 70% test coverage.
* Micro Services Architecture.
* Use of a JMS queue.

You can find the PokeAPI at:

<https://pokeapi.co/>

## Planning

Plan out this application by:

1. Creating an architecture diagram.
2. Creating rough class diagrams for each of your services.

## User Stories:

**User Story:** As a user, I can create a new user by providing a *memberNumber* and a *name.*

**User Story:** As a user, I can search for a Pokemon by **name** using my *userId.*

**User Story:** As a user, I can search for a Pokemon by **number** using my *userId.*

**User Story:** As an auditer, I can search for all the search terms that have been used, this is so that I can see who has made the search and at what time it was made.

**User Story:** As a user, I want the system to inform me when I am trying to search for a Pokemon with a user that does not exist.

## Code Sample:

If you have problems sending a GET request to the PokeAPI from Spring Boot please use the code snippet below.

*@Override*

public ResponseEntity<Object> findByNumber(String number) {

HttpHeaders headers = new HttpHeaders();

headers.add("user-agent",

"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36");

HttpEntity<String> entity = new HttpEntity<String>("parameters", headers);

return restTemplate.exchange("https://pokeapi.co/api/v2/pokemon/"+number, *HttpMethod*.***GET***, entity, Object.class);

}

# Stretch

Spring Boot is the main focus of this exercise, however once you have completed all the aspects above there are multiple avenues to how you can expand this project that will all be fantastic practice for your group Project starting next week.

* You will need a record of the results of your searches for the Group Project. It is likely you will need to search by a particular user or search term. You can practice introducing this functionality here.
* You can build a React Front-end for this application; you will need to build one of these out for the Group Project.
* You can Dockerise the application and deploy it in the cloud, you need to do this for your group Project.
* You can write the documentation for your project.
* You can build a Jenkins server to automate the above deployment.