Microservices lab 2

Goals

- Create another microservice that's going to communicate with the one created during the first lab. You can find an example here.
- Make it work even if the new microservice isn't reachable or working properly.

Steps

1. Create a second microservice that exposes a simple REST api. At first, you're only going to need some GET endpoints (*find all* and *find one*).
   Don't forget that the microservice architecture allows you to build your new webapp **in another language** (Nodejs, Go, Python,...) and/or **another framework** (Play, Ktor,...).

2. Make a request to the first microservice triggers a request to the second one.

3. Add a circuit-breaker in order to make requests to the first service working even if the second one isn't ok. There are at least three cases to handle :
   a. Service B is unreachable (network down or app down).
   b. Service B is very slow (timeout).
   c. Service B is unreliable (retry).

In order to add the circuit-breaker pattern to your app I recommend you to use this starter.