Cat Facts

nifr@kea.dk

September 2020

1 Introduction

Today we are going to create an application that finds cat facts via an online API, maps the facts to a java object, sorts & orders them.
I highly recommend you working in groups or pairs.

2 Notes

- CatFactAPI: https://cat-fact.herokuapp.com
- Endpoint for a single fact: /facts/random
- Inspiration on how to consume the API map the result from JSON to a POJO:
  https://github.com/nicklasfrederiksenkea/Dat20v1CatFacts
- NOTE: We need a new dependency in pom.xml (gson) which is used for mapping

3 Requirements

We are going to create 5 end-points, that each will have a specific task in mind. You can use the same Controller class. Annotate the controller methods with @ResponseBody.

3.1 /

This end-point welcomes the user and prints out a short description of your application

3.2 /getSingle

This end-point calls a service, that fetches a single fact from cat-fact API, maps the result to a CatFact class and prints the fact to the browser.

3.3 /getTen

This end-point calls a service, that fetches ten facts from cat-fact API, maps the result to a CatFact class, adds to a Arraylist that can only contain CatFacts and prints the result to the browser (hint: we can call the API several times and a GET-mapping-method can return an ArrayList if required.

3.4 /getTenSortByDate

This end-point calls a service that fetches ten facts from cat-fact API, maps the result to a CatFact class, adds to a CatFact Arraylist and prints the result to the browser - sorted in ascending order by Date of creation.

3.5 /contains?char=x&amount=n

This end-point requires two parameters. The parameters required are a character & an integer n. The end-point will print the fact if the character is contained n amount of times.
Otherwise it will print: ”Sorry no luck”
Hint: https://www.baeldung.com/spring-request-param