



Projeto 1: API RESTful para baralho de cartas

05/08/2024

1 Objetivo

Com base na aula sobre Serviços *Web* e no Laboratório 4 - Serviços Web REST em Java, o objetivo desse projeto é desenvolver uma API RESTful para um baralho de cartas. A API deve ser desenvolvida em Java com o uso do *framework* Spring Boot e deve permitir as seguintes operações:

1. Criar um novo baralho de cartas

- Todo baralho deve possuir um identificador único, que será retornado na resposta da requisição (documento JSON). O identificador não deve ser previsível e deve ser gerado a cada requisição
- Baralho deve ser composto por 52 cartas, sendo 13 de cada naipe (copas, espadas, ouros e paus) e cada naipe com 13 valores (Ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, Valete, Dama e Rei).
- As cartas sempre serão criadas na ordem: Ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, Valete, Dama e Rei e os naipes sempre serão criados na ordem: copas, espadas, ouros e paus

2. Listar todas as cartas de um baralho novo

- Retornar todas as cartas do baralho (identificador do baralho deve ser informado na URL) e retornar um documento JSON com todas as cartas do baralho
- As cartas devem conter um código único da carta, o naipe, o valor e a URL de uma imagem da carta
- Se o baralho estiver embaralhado, então deve-se retornar um documento JSON informando que o baralho está embaralhado e não é possível listar todas as cartas

3. Embaralhar as cartas

- Embaralhar as cartas do baralho (identificador do baralho deve ser informado na URL), retornar um documento JSON com indicação de sucesso e quantas cartas ainda estão no baralho

4. Retirar n cartas do baralho

- Retirar n cartas do baralho (identificador do baralho deve ser informado na URL) e retornar um documento JSON com as cartas retiradas (código único da carta, o naipe, o valor e a URL da imagem), além de quantas cartas ainda estão no baralho
- Se o baralho estiver vazio, retornar um documento JSON com uma mensagem informando que o baralho está vazio

5. Excluir um baralho

- Excluir um baralho (identificador do baralho deve ser informado na URL) e retornar código de status 204 (No Content) em caso de sucesso ou 404 (Not Found) caso o baralho não exista

2 Requisitos

- ☐ Indicar no arquivo `Readme.md` quais requisitos foram atendidos e quais não foram atendidos
- ☐ Desenho da API RESTful respeitando os verbos HTTP e os códigos de status
- ☐ Implementação da API RESTful em Java com a lógica correta para cada operação
- ☐ Documentação da API RESTful com o uso do OpenAPI (ver Laboratório 4)
- ☐ Arquivo `Readme.md` com instruções com o cURL para testar cada operação da API
- ☐ Arquivo `.gitignore` adequado para projetos Java com gradle
- ☐ Após clonar o repositório, ser possível compilar e executar usando o `gradle bootRun`

3 Material de apoio

Na [Listagem 1](#) tem um exemplo de como transferir uma imagem com o *framework* Spring. Neste exemplo, as imagens estão no diretório `src/main/resources/static/cartas`. A URL para acessar a imagem é <http://localhost:8080/carta/1c.png> (de acordo com os nomes dos arquivos das cartas contidas no repositório <https://github.com/poo29004/jogo-cartas>).

Listagem 1: Exemplo de como transferir uma imagem com o *framework* Spring

```
1 package engtelecom.std.labrest;
2
3 import java.io.InputStream;
4 import org.springframework.core.io.InputStreamResource;
5 import org.springframework.http.MediaType;
6 import org.springframework.http.ResponseEntity;
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.PathVariable;
9 import org.springframework.web.bind.annotation.ResponseBody;
10 import org.springframework.web.bind.annotation.RestController;
11
12 @RestController
13 public class Teste {
14
15     @GetMapping(value= "/carta/{carta}", produces = MediaType.IMAGE_PNG_VALUE)
16     @ResponseBody
17     public ResponseEntity<InputStreamResource> getImage(@PathVariable String carta){
18         InputStream is = Teste.class.getClassLoader().getResourceAsStream("static/cartas/"+carta);
19         return ResponseEntity.ok().body(new InputStreamResource(is));
20     }
21 }
```

Plágio não é tolerado



Você deve ser o único(a) responsável por fazer a entrega para essa atividade. Todo o código ou texto deverá ser produzido exclusivamente por você, exceto trechos de códigos que possam ter sido fornecidos como parte do enunciado.

Você pode discutir com outros estudantes com o intuito de esclarecer pontos, isso é até incentivado, porém você não poderá copiar trechos de códigos, textos ou soluções de qualquer fonte (e.g. colegas da mesma turma ou de turmas anteriores, repositórios de códigos na Internet ou soluções providas por serviços como Copilot e ChatGPT).