Java – Spring Framework – Respostas

Objetivos:

•Identificar a estrutura de um serviço REST.
•Compreender o papel do Spring Framework na implementação de serviços REST.
•Configurar projeto Spring Boot para a implementação de serviços REST.
Atividades:

1)Siga o starter-guide "Building a RESTful Web Service" disponível em https://spring.io/guides/gs/rest-service/ .

2)Utilize uma ferramenta para testar e observar a requisição e reposta do protocolo HTTP ao acessar o serviço implementado no exercício anterior. Sugere-se o uso do "Postman", disponível em https://www.getpostman.com/.

3)Responda as seguintes questões:

a)Qual o propósito da anotação @RestController? É uma anotação que indica que uma classe é o controle REST, é atravez disso que o String sabe que a classe está preparada para receber requisição

b)Qual o propósito da anotação @RequestMapping? Qual a sua relação com as anotações @GetMapping, @PostMapping, @PutMapping, @DeleteMapping, e @PatchMapping?
É a anotação utilizada tradicionalmente para implementar URL handler, ela suporta os métodos Post, Get, Put, Delete e Pacth.

c)Qual o propósito da anotação @RequestParam? É utilizado para pegar uma parâmetro de query da url, por exemplo:

http://localhost:8080/topicos?curso=Java

O parâmetro curso é um parâmetro de url e para você recuperá-lo no controller, você deve utilizar a anotação @RequestParam:

```
@GetMapping("/topicos")
public void exemplo(@RequestParam String curso) {
   //...
}
```

d) De que maneira se configura o acesso aos elementos componentes de uma requisição HTTP?

e)Qual o propósito da bilbioteca "Jackson"?

No Java existem algumas bibliotecas para fazer conversão Java <-> JSON, dentre elas: Jackson, Jettison e Gson.

O Spring por padrão utiliza o Jackson, mas é possível substituir por essas outras, caso você queira.

O Jackson funciona da seguinte maneira:
ObjectMapper mapper = new ObjectMapper();
TopicoDto topico = mapper.readValue(json, TopicoDto.class);

4)Siga o tutorial "Step-By-Step Spring Boot RESTful Web Service Complete Example" disponível em https://dzone.com/articles/spring-boot-restful-web-service-complete-example .

5)Nenhum dos exemplos anteriores realizou a configuração/customização de mensagens de erro no caso de exceções serem geradas pela camada de negócio da aplicação. Leia o material em https://www.amitph.com/spring-rest-service-exception-handling/ para entender as diferentes opções disponíveis no Spring MVC.

6)Revise o processo de configuração das APIs de validação do Spring MVC em um exemplo de serviço REST em https://dzone.com/articles/implementing-validation-for-restful-services-with.

Veja Também:

- Understanding REST https://spring.io/understanding/REST
- MDN Web Docs HTTP https://developer.mozilla.org/pt-BR/docs/Web/HTTP
- Padrão Front Controller https://martinfowler.com/eaaCatalog/frontController.html
- Padrão Service Layer https://martinfowler.com/eaaCatalog/serviceLayer.html

- Padrão Remote Facade https://martinfowler.com/eaaCatalog/remoteFacade.html
- Padrão Data Transfer Object https://martinfowler.com/eaaCatalog/dataTransferObject.html
- Spring Framework Web MVC (capítulos 1.1 e 1.3) https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html