



Java Advanced

Criando uma API RESTful
Prof. Luiz Gustavo S. Real



Agenda

- Projeto desta aula (uma API RESTful)
 - O que é uma API?
 - REST vs. RESTful
 - CRUD
 - Próximos passos

Quem é o professor?

- Luiz Gustavo dos Santos Real
 - Por que escolhi a TI?
 - Formação
 - Experiência
 - O que esperar das aulas?
 - LinkedIn



Projeto desta aula (uma API RESTful)

- O que é API?
 - API significa "Application Programming Interface", ou Interface de Programação de Aplicativos.
 - É um conjunto de definições e protocolos que permite a comunicação entre diferentes softwares.
- REST vs. RESTful
 - REST é um estilo de arquitetura de software que define um conjunto de restrições para a criação de serviços web. Pensem em REST como um conjunto de boas práticas para a comunicação entre sistemas.
 - Já RESTful se refere a um serviço web que segue os princípios da arquitetura REST. Ou seja, um serviço web RESTful é aquele que implementa as restrições definidas pelo REST.

Projeto desta aula (uma API RESTful)

- As principais características de uma API RESTful são:
 - **Uso de HTTP:** A comunicação acontece através dos verbos HTTP, como GET, POST, PUT, DELETE, cada um com um significado específico.
 - **Stateless:** Cada requisição é independente da anterior. O servidor não guarda estado da comunicação.
 - **Formato de dados:** APIs RESTful normalmente usam **JSON** ou XML para troca de dados.

Projeto desta aula (uma API RESTful)

- O que é CRUD?
 - CRUD é um acrônimo para Create, Read, Update e Delete (Criar, Ler, Atualizar e Deletar). Essas são as operações básicas que podemos realizar com dados. APIs RESTful geralmente implementam operações CRUD, usando os verbos HTTP para mapear as ações:
 - **POST:** Criar um novo recurso.
 - **GET:** Ler um recurso ou uma coleção de recursos.
 - **PUT:** Atualizar um recurso existente.
 - **DELETE:** Deletar um recurso.



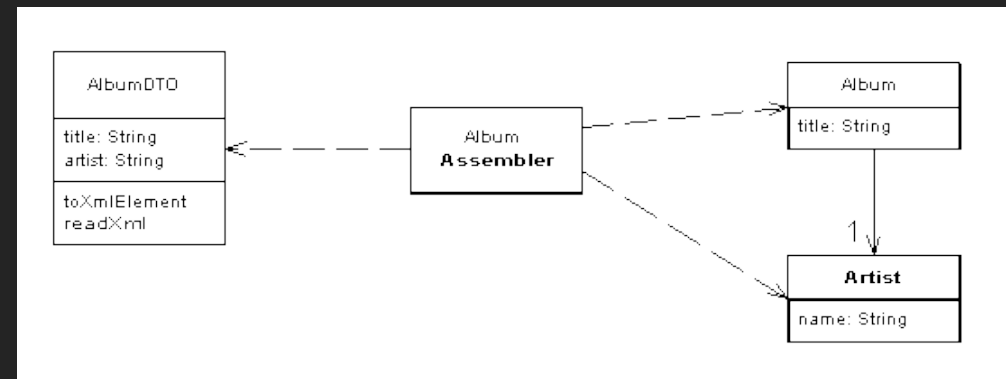
Próximos passos

- DTO
- Paginação
- Validação
- Tratamento de exceção



DTO – Data Transfer Object

- DTOs são objetos que carregam dados entre processos a fim de reduzir o número de chamadas a métodos. O padrão foi introduzido por Martin Fowler.
- Pode ser usado para consolidar atributos para exibir na API



Pageable – Paginação

- O Pageable é uma interface do Spring Data JPA que facilita a implementação da paginação de dados. Você pode dividir grandes conjuntos de dados em partes menores (páginas) para melhorar a performance e a usabilidade da aplicação.
- Elementos chave do Pageable:
 - Número da página: Define qual página você deseja acessar, iniciando em zero.
 - Tamanho da página: Define a quantidade de elementos que cada página deve conter.
 - Ordenação: Define como os elementos devem ser ordenados na página (exemplo: nome em ordem decrescente).

Bean Validation – Validação

- O Bean Validation define diversas anotações para aplicar restrições aos atributos de suas classes. Algumas são:
 - @NotNull: Garante que o valor não seja nulo.
 - @NotBlank: Garante que o valor não seja nulo ou uma string vazia.
 - @Size: Define o tamanho mínimo e/ou máximo de strings, arrays e coleções.
 - @Min: Define o valor mínimo permitido para números.
 - @Max: Define o valor máximo permitido para números.
 - @Email: Valida se o valor é um endereço de email válido.
 - @Pattern: Valida se o valor corresponde a um padrão regex definido.
- Para ativar a validação, basta adicionar a anotação @Valid aos argumentos do controlador que deseja validar.

Exception Handler – Tratamento de Exceção

- O Spring automaticamente intercepta as exceções lançadas pela validação (@Valid) e as converte em respostas HTTP 400 (Bad Request). Você também pode personalizar o tratamento de erros e retornar mensagens personalizadas usando um @ExceptionHandler



Próximos passos

- HATEOAS
- Documentação



HATEOAS

- Hypermedia As The Engine Of Application State
- Um dos princípios fundamentais para uma API ser considerada RESTful
- Spring oferece suporte nativo
- Em vez de entregar respostas estáticas, a API RESTful que implementa HATEOAS enriquece cada resposta com links contextuais. Esses links indicam ao cliente quais ações podem ser executadas a seguir e quais recursos estão relacionados ao que ele está visualizando.

Documentação



- Swagger OpenAPI
 - OpenAPI define um formato padrão para documentar APIs
 - Swagger fornece uma interface web interativa para visualizar e explorar a documentação OpenAPI
- Mantém a API sempre alinhada com o código
- Torna o processo de documentação mais rápido
- Possui muitos recursos de personalização



Próximos passos

- Testes
- Spring Security
- Deploy

