Criação de uma API com Java e Spring Boot

Nesta aula, vamos explorar a criação de uma API com Java e Spring Boot, conhecendo o projeto prático da disciplina e iniciando sua implementação. Utilizando o Spring Boot, integraremos os módulos Spring MVC, Spring Data JPA e Spring Security para desenvolver uma aplicação web robusta e segura.

O projeto consistirá na criação de um mini blog, onde usuários autenticados poderão criar postagens, adicionar comentários e utilizar tags como hashtags para filtragem. As entidades principais do projeto serão Post, User, Comment e Tag, todas modeladas em um diagrama de entidade-relacionamento que auxiliará na compreensão da estrutura e dos relacionamentos no banco de dados.

Ao longo desta aula, você aprenderá a inicializar um projeto Java com Spring Boot, configurar as dependências necessárias e implementar as operações básicas de CRUD (Create, Read, Update, Delete) para cada uma das entidades. Esse conhecimento fundamental permitirá que você construa aplicações web escaláveis e seguras, ampliando suas habilidades como desenvolvedor.

Apresentação do Projeto

Vamos começar criando uma API com Java e Spring Boot. Você conhecerá o projeto da disciplina e começará a colocar em prática seus conhecimentos sobre APIs e o framework Spring Boot. O objetivo desta aula é apresentar o projeto que será desenvolvido ao longo do curso.

O projeto consiste na criação de uma aplicação web utilizando Spring Boot, com os módulos Spring MVC, Spring Data JPA e Spring Security. Esses módulos são voltados para o desenvolvimento de aplicações web, utilizando o padrão de design Model-View-Controller (MVC), a persistência de dados com Hibernate e a implementação de segurança com autenticação e autorização.

O projeto será um mini blog, que permitirá a criação de usuários autenticados, postagens, comentários e a utilização de tags como hashtags para filtragem. As quatro entidades principais do projeto são: Post, User, Comment e Tag. Essas entidades serão modeladas em um diagrama de entidade-relacionamento, que estará disponível no GitHub da disciplina.

O diagrama de entidade-relacionamento ajuda a entender a estrutura e os relacionamentos entre as tabelas do banco de dados. No nosso caso, teremos as tabelas Tag, Post, User e Comment. Cada tabela terá um ID único, incrementado automaticamente pelo JPA. A tabela Tag conterá o ID e o nome da tag. A tabela Post conterá o ID, título, conteúdo, data da postagem e uma chave estrangeira para identificar o usuário que fez a postagem. A tabela User terá o ID, nome, e-mail, senha e role (papel), indicando se o usuário é administrador ou não. A tabela Comment terá o ID, conteúdo, data, a identificação da postagem e do usuário que fez o comentário.

Essa estrutura permitirá a criação de postagens com múltiplos comentários, usuários que podem criar várias postagens e comentários, e a utilização de tags em múltiplas postagens. O projeto será detalhado e atualizado conforme necessário durante a implementação, sempre disponível para consulta no GitHub da disciplina.

Inicializando um Projeto Java com Spring Boot

Nesta seção, vamos iniciar um projeto Java com Spring Boot. Para isso, utilizaremos o site start.spring.io, que é a página oficial para começar um

projeto do zero com Spring Framework. Primeiramente, verifique se o Java e o Maven estão instalados na sua máquina. Recomendo usar a versão 17 do JDK, disponível no site da Oracle, e a última versão do Maven.

Acesse o site start.spring.io, onde você encontrará a tela de inicialização do projeto. Escolha o Maven como gerenciador de dependências e Java como a linguagem de programação. Mantenha a versão estável mais recente do Spring Boot selecionada. No campo Project Metadata, insira o nome do seu grupo e artefato. No nosso caso, o grupo será com.descomplica e o artefato será frameblog, que é o nome da aplicação.

Adicione as dependências necessárias: Spring Web, Spring Data JPA e Spring Security. Essas dependências são essenciais para a implementação do nosso projeto. O Spring Web permitirá a criação de endpoints, o Spring Data JPA facilitará a persistência de dados com Hibernate, e o Spring Security implementará a segurança da aplicação.

Após configurar as dependências, gere o arquivo zip do projeto e faça o download. Extraia o arquivo e importe o projeto na sua IDE como um projeto Maven. A estrutura básica do projeto incluirá uma classe principal para iniciar a aplicação, arquivos de configuração e as dependências necessárias.

Criação de Endpoints

Vamos criar os endpoints do nosso projeto. Um endpoint é uma URL que permite o acesso a um serviço específico pela aplicação cliente. Primeiramente, importe o projeto na sua IDE e configure a porta 8080 no arquivo application.properties.

Crie os pacotes necessários para a estrutura do projeto: models, enums, controllers e services. No pacote models, crie as classes das entidades: User, Tag, Post e Comment. Utilize as anotações @Entity e @Table para definir as entidades e suas respectivas tabelas no banco de dados. Para

cada entidade, defina os atributos e os relacionamentos, como o relacionamento many-to-one entre Post e User.

No pacote controllers, crie as classes dos controladores: UserController, TagController, PostController e CommentController. Utilize a anotação @RestController para mapear as classes como controladores REST e @RequestMapping para definir os caminhos dos endpoints. Por exemplo, no UserController, defina o caminho "/users" e crie os métodos para as operações básicas do CRUD (Create, Read, Update, Delete).

No pacote services, crie as interfaces dos serviços: UserService, TagService, PostService e CommentService. Essas interfaces definirão os contratos para as operações das entidades. Implemente as interfaces no pacote de implementação (imp), utilizando as anotações @Service e @Autowired para injetar as dependências dos repositórios.

Criação do CRUD do Projeto

Agora, vamos continuar a implementação dos endpoints, focando nas operações básicas do CRUD. CRUD é um acrônimo para Create, Read, Update e Delete, que são as operações fundamentais para a manipulação de dados em uma aplicação.

No UserController, criamos o método save para salvar um novo usuário. Utilizamos a anotação @PostMapping para mapear o endpoint "/save" e a anotação @RequestBody para receber os dados do usuário na requisição. No método save, verificamos se o usuário já existe no banco de dados. Se existir, retornamos um erro. Caso contrário, criamos um novo usuário e salvamos no banco de dados utilizando o repositório.

Para as operações de leitura, criamos os métodos getAll e getById. Utilizamos as anotações @GetMapping e @PathVariable para mapear os endpoints e receber os parâmetros da requisição. No método getAll,

retornamos uma lista de todos os usuários do banco de dados. No método getById, retornamos um usuário específico baseado no ID.

Para as operações de atualização, criamos o método update. Utilizamos a anotação @PostMapping e recebemos os dados do usuário na requisição. Verificamos se o usuário existe e, em caso afirmativo, atualizamos os dados no banco de dados.

Para a operação de exclusão, criamos o método delete. Utilizamos a anotação @DeleteMapping e recebemos o ID do usuário na requisição. Verificamos se o usuário existe e, em caso afirmativo, removemos do banco de dados.

A estrutura do CRUD deve ser replicada para as outras entidades (Tag, Post e Comment), utilizando os mesmos princípios de mapeamento de endpoints, injeção de dependências e validação de dados. Isso garantirá uma aplicação robusta e consistente.

Ao final, você terá uma aplicação funcional com as operações básicas de CRUD implementadas, pronta para ser expandida e aprimorada conforme necessário.

GitHub da Disciplina

Você pode acessar o repositório da disciplina no GitHub a partir do seguinte link:

https://github.com/FaculdadeDescomplica/Framework. Esse espaço é o seu portal para mergulhar fundo no universo da aprendizagem interativa. Nele, você encontrará todos os códigos, além dos links para os arquivos e dados.

Conteúdo Bônus

Para um conteúdo bônus gratuito e valioso para alunos de graduação sobre Framework e Criação de uma API com Java e Spring Boot, recomendo o curso “Spring Boot Tutorial for Beginners” disponível no YouTube pelo canal freeCodeCamp.org. Este curso oferece uma introdução abrangente ao Spring Boot, abordando desde a configuração inicial até a criação de uma API completa, com exemplos práticos e explicações detalhadas.

Título: Spring Boot Tutorial for Beginners (Java Framework)

Canal: freeCodeCamp.org

Plataforma: Youtube

Esse curso é ideal para complementar os conhecimentos adquiridos em aula, proporcionando uma visão prática e detalhada do desenvolvimento de APIs com Java e Spring Boot.

Referência Bibliográfica

CARDOSO, L. da C. **Design de aplicativos**. Intersaberes: 2022

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7.ed. Pearson: 2018.

JOÃO, B. do N. **Usabilidade e interface homem-máquina**. Pearson: 2017

LEAL, G. C. L. **Linguagem, programação e banco de dados**: guia prático de aprendizagem. Intersaberes: 2015.

MEDEIROS, L. F. de. **Banco de dados**: princípios e prática. Intersaberes: 2013.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados**: implementação em SQL, PL/SQL e Oracle 11g. Pearson: 2013.

SETZER, V. W.; SILVA, F. S. C. **Bancos de dados**. Blucher: 2005.

VICCI, C. (Org.). **Banco de dados**. Pearson: 2014.

Ir para exercício