Java JPA: persistência de objetos Java para tabelas de forma automática.b

Springboot já tem embutido o Tomcat, que é o servidor da aplicação (pra rodar a aplicação, que é o container).

Como modificar a porta do localhost, caso a porta 8080 esteja já sendo utilizada?

Finalizar aplicação – src/main/resources – application.properties – escrever: server.port=${port:8081} salvar e rodar de novo a aplicação.

Inicialmente já tem toda configuração como dependências (pom.xml).

1 commit

Controladores REST: são os recursos (resources)

Padrão REST: colocar os verbos corretamente sobre as requests. **(estudar endpoint REST)**

hashCode e equals: para que sejam verificados se os objetos são iguais pelos conteúdos e não pelos ponteiros de memória

Classe Serializable: declaração na classe para que ela implemente a interface Serializable objetos podem ser convertidos para sequência de bytes para poderem serem gravados em arquivos ou usados na web.

Quando testamos no Postman os dados já saem em JSON, o próprio Springboot faz a conversão de dados para JSON com os mesmos nomes das Classes.]

OBS: Interfaces não podem ser instanciadas, portanto escolhemos uma classe que implementa a interface.

Ex: List<Categoria> lista = new ArrayList<>( );

2 commit

Dependências a instalar: Banco de dados H2 (relacional) ele é em memória e open source.

JPA padrão Java...

Devtools: utilizar para testar o banco de dados

Por padrão, no pom.xml temos as dependências padrão do Springboot: starter web (escolhemos no início do projeto)

3 commit