

CETEJ35 - Java Web - JAVA_XXX (2024_01)

[Meus cursos](#) / [CETEJ35 - Web \(2024_01\)](#) / [Semana 07: 14/10 a 20/10](#) / [API Reativa](#)

API Reativa

✓ **Feito:** Ver

A fazer: Gastar pelo menos 20 minutos na atividade


A fazer: Passar pela atividade até o fim

Fecha: segunda-feira, 2 dez. 2024, 00:00

Nesta aula vamos construir uma nova aplicação que funciona como uma API, recebendo dados usando a arquitetura REST. Para fazer isso, vamos usar uma nova tecnologia do Spring - o Spring WebFlux.

ENTIDADE

Vamos usar uma abordagem *bottom-up* para desenvolver esse projeto, começando pela entidade persistente. Para isso, criamos o arquivo **Todo.java** no pacote **demo**. Esse arquivo representa uma *tarefa* no nosso gerenciador de tarefas.

Em vez de uma classe, criamos um **record**. Um java **record** é uma entidade **imutável** para a transferência de dados entre objetos. **Record** é uma adição da [versão 16 do Java](#). Um detalhe relevante é que um **record** é imutável. Dessa forma, uma vez atribuídos valores, eles não podem ser alterados. Outras características do **record** tornam ele [inviável](#) para u persistente com JPA. Contudo, ele funciona bem com MongoDB.

Se Java **record** ainda é algo novo pra você, você pode [conhecer mais sobre o assunto nesse link](#).

Diferente de uma classe, um **record** define os atributos logo na sua declaração. Nesse projeto, vamos considerar que uma *tarefa* tem os seguintes atributos:

- *id* que identifica a tarefa como única em uma lista de tarefas;
- *titulo* que define o nome da tarefa;
- *descricao* que descreve detalhes da tarefa; e,
- *feito* que marca a tarefa como concluída ou não.

Para garantir que toda tarefa tem um título minimamente descritivo, vamos adicionar uma validação. Colocamos a validação no construtor do **record**. Desse modo, uma exceção será levantada caso o título seja nulo ou menor que três caracteres. O código na figura abaixo mostra o resultado final do **Todo.java**.

```
1 package com.example.demo;
2
3 public record Todo(String id, String titulo, String descricao, boolean feito) {
4
5     public Todo {
6         if (titulo == null || titulo.length() < 3) {
7             throw new IllegalArgumentException("Um título maior que 3 caracteres é necessário.");
8         }
9     }
10
11 }
```

A linha 3 mostra a definição do **record**, seus atributos, e o tipo de cada atributo. As linhas 5 a 9 definem o construtor. A validação entre as linhas 6 e 8 garante um título minimamente válido para a *arefa*.

Notou a ausência de qualquer anotação identificando essa classe como entidade persistente? Não, não está errado. O Spring Data reconhece o uso da entidade mesmo sem qualquer anotação. **Mas atenção:** isso não funciona para entidades JPA!

O código desenvolvido nesta Seção está disponível no [Github](#) na branch **semana06-20-entidade**.


Retroceder

Avançar

◀ Verificação de aprendizado - Integração

Seguir para...

Atividade II WebConf ▶

✉ Contate o suporte do site 

Você acessou como RAFAEL ROCHA DA SILVA PROENCA (Sair)
CETEJ35 - Web (2024_01)



- Tema
- Adaptable
- Boost
- Clássico
- Campus
- Apucarana
- Campo Mourão
- Cornélio Procópio
- Curitiba
- Dois Vizinhos
- Francisco Beltrão
- Guarapuava
- Londrina
- Medianeira
- Pato Branco
- Ponta Grossa
- Reitoria
- Santa Helena
- Toledo
- UTFPR
- Ajuda
- Chat UTFPR
- Calendário Acadêmico
- Biblioteca
- e-Mail
- Nuvem (OwnCloud)
- Produção Acadêmica
- Secretaria Acadêmica
- Sistemas Corporativos
- Sistema Eletrônico de Informação - SEI
- Suporte ao usuário
- Criação de curso
- Comunidade
- Português - Brasil (pt_br)
- Deutsch (de)
- English (en)



Português - Brasil (pt_br)

Resumo de retenção de dados

Baixar o aplicativo móvel.

 Dê um feedback sobre este software 

Universidade Tecnológica Federal do Paraná - UTFPR

Suporte ao usuário

