

CETEJ35 - Java Web - JAVA_XXX (2024_01)

[Meus cursos](#) / [CETEJ35 - Web \(2024_01\)](#) / [Semana 04: 23/09 a 29/09](#) / [Criando, Alterando e Excluindo](#)

Criando, Alterando e Excluindo

✓ **Feito:** Ver

✓ **Feito:** Gastar pelo menos 20 minutos na atividade

✓ **Feito:** Passar pela atividade até o fim

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

Nesta aula nós finalizamos a implementação das quatro operações CRUD. Isso significa que nosso usuário será capaz de criar, alterar, excluir e listar as cidades em uma base de dados. Observe que ainda estamos usando uma base local, baseada em uma lista em memória. Nós vamos evoluir esse projeto até integrarmos essa base com um banco de dados.

CRIANDO

Para criar uma nova cidade, vamos começar alterando nossa página Web (*View*) para que ela envie dados da cidade (*Model*) para o método `criar()`, na classe `CidadeController` (*Controller*).

Abra o arquivo `crud.ftl` e adicione as seguintes propriedades conforme indicado na Figura abaixo.



```
20 | <form action="/criar" method="POST">
```

A propriedade `action` indica para onde os dados devem ser enviados, enquanto a propriedade `method` indica o método HTTP que deve ser usado para o envio de dados. POST costuma ser o método mais adequado para envio de dados de formulários.

O próximo passo é adicionar a propriedade `name` nas tags `input`, que são usadas para inserir o nome da cidade e do estado. A propriedade `name` mapeia os valores digitados na tag `input` com os atributos na classe `Cidade` (*Model*). Para que esse mapeamento seja feito automaticamente, é importante que os valores usados na propriedade `name` seja igual ao nome dos atributos na classe `Cidade`. Veja na Figura a seguir como ficou o código do formulário após as alterações.

```
src > main > resources > templates > crud.ftl
20 | <form action="/criar" method="POST">
21 |   <div class="form-group">
22 |     <label for="nome">Cidade:</label>
23 |     <input name="nome" type="text"
24 |       class="form-control" placeholder="Informe o nome
25 |       da cidade" id="nome">
26 |   </div>
27 |   <div class="form-group">
28 |     <label for="estado">Estado:</label>
29 |     <input name="estado" type="text"
30 |       class="form-control" placeholder="Informe o
31 |       estado ao qual a cidade pertence"
32 |       id="estado">
33 |   </div>
34 |   <button type="submit" class="btn btn-primary">CRIAR</
35 |   button>
36 | </form>
37 | <table class="table table-striped table-hover mt-5">
38 |   <thead class="thead-dark">
```

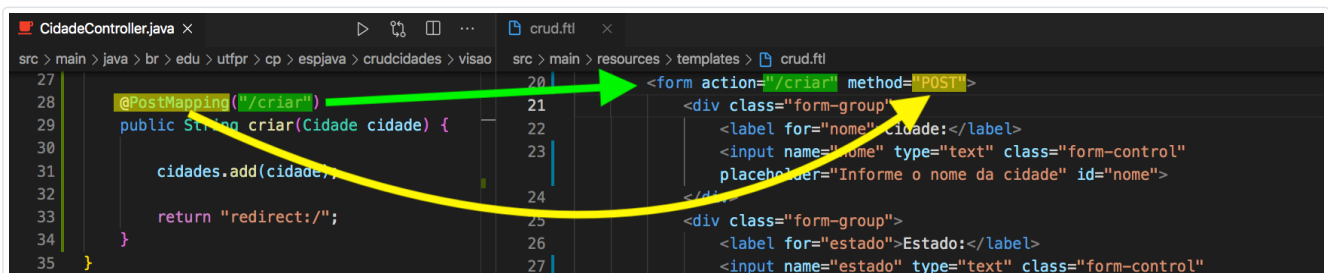
```
src > main > java > br > edu > utfpr > cp > espjava > crudcidades > visao > Cidade.java > ...
1 | package br.edu.utfpr.cp.espjava.crudcidades.visao;
2 |
3 | public final class Cidade {
4 |     private final String nome;
5 |     private final String estado;
6 |
7 |     public Cidade(final String nome, final String estado) {
8 |         this.nome = nome;
9 |         this.estado = estado;
10 |     }
11 |
12 |     public String getEstado() {
13 |         return estado;
14 |     }
15 |
16 |     public String getNome() {
17 |         return nome;
18 |     }
19 | }
```

Agora, vamos adicionar o método responsável por receber os dados do formulário e adicionar a nova cidade na lista. Para isso, abra a classe **CidadeController** e crie um novo método, que vamos chamar de **criar()**.

Esse método deve receber um objeto do tipo **Cidade** como parâmetro. Assim como o método **listar**, **criar** retorna uma **String**, representando a página que deve ser carregada. Nesse caso, o que queremos é que o usuário seja enviado para a página **crud.ftlh**. Mas também queremos que a lista de cidades seja carregada e mostrada na tabela.

Por isso, em vez de retornar o nome da página, vamos retornar um **redirect:/**, que representa um redirecionamento para o método **listar()**. O método **listar()**, por sua vez, carrega a lista de cidades e chama a página **crud.ftlh**.

Por fim, precisamos adicionar a anotação que vai mapear o endereço no formulário com o método **criar()**. Diferentemente do método **listar()**, o método **criar()** vai receber uma anotação **org.springframework.web.bind.annotation.PostMapping("/criar")**. A URL na anotação deve ser a mesma usada na propriedade **action**, do formulário na página **crud.ftlh**. Também observe que estamos usando um **@PostMapping** em vez de **@GetMapping**. Isso porque o formulário envia os dados por meio do método **POST**, do protocolo HTTP.



```
CidadeController.java
27
28 @PostMapping("/criar")
29 public String criar(Cidade cidade) {
30     cidades.add(cidade);
31     return "redirect:/";
32 }
33
34
35

crud.ftlh
20 <form action="/criar" method="POST">
21 <div class="form-group">
22 <label for="nome">Cidade:</label>
23 <input name="nome" type="text" class="form-control"
24     placeholder="Informe o nome da cidade" id="nome">
25 </div>
26 <div class="form-group">
27 <label for="estado">Estado:</label>
28 <input name="estado" type="text" class="form-control"
```

Veja como ficou o código completo do *Controller* na Figura abaixo.



```

11  @Controller
12  public class CidadeController {
13
14      private Set<Cidade> cidades;
15
16      public CidadeController() {
17          cidades = new HashSet<>();
18      }
19
20      @GetMapping("/")
21      public String listar(Model memoria) {
22
23          memoria.addAttribute("listaCidades", cidades);
24
25          return "/crud";
26      }
27
28      @PostMapping("/criar")
29      public String criar(Cidade cidade) {
30
31          cidades.add(cidade);
32
33          return "redirect:/";
34      }
35  }

```



Neste ponto, você já pode executar o projeto e tentar inserir uma nova cidade.

Não esqueça que estamos usando uma lista em memória para guardar os valores. Por isso, quando o aplicativo é reiniciado, os dados são perdidos. Muito em breve vamos adicionar persistência no código e armazenar os dados em banco.

O código desenvolvido nesta Seção está disponível no [Github](#), na branch `semana03-20-crud-criar`.


Retroceder

Avançar

◀ Atividade I WebConf

Seguir para...

Verificação de aprendizado - Criando, Alterando e Excluindo ▶

✉ 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 Procopio

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

