

PROCESSO DE DESENVOLVIMENTO		
DOCUMENTAÇÃO DE PROJETO		
Nome do curso: Java Front/Back	Aula 3 - Servlets	Responsável: Raphael Nascimento

Vídeo 3.1: criando uma classe usando Servlet

Olá, seja bem-vindo(a)!

Vamos a mais uma videoaula do curso de desenvolvimento Java web! Nesse vídeo, você irá conhecer a Servlet, que é uma importante estrutura para o desenvolvimento de aplicações web em Java. No vídeo passado, você conheceu o Tomcat, que é um exemplo de aplicação Java que implementa a tecnologia de Servlet.

Inclusive, essa tecnologia foi pioneira do desenvolvimento em Java web e possibilitou o desenvolvimento de páginas dinâmicas. Assim, a Servlet foi evoluindo durante os anos, principalmente pela grande aceitação da comunidade, e isso fez com que essa ferramenta seja até hoje utilizada no desenvolvimento de software web. Portanto, Servlets são classes Java que atuam como pequenos servidores, recebendo requisição, processando e retornando uma resposta. A partir do surgimento dessa tecnologia, se tornou possível montar uma resposta a uma requisição, utilizando parâmetros provenientes da própria requisição.

Como visto na aula passada, as respostas às requisições eram estáticas, já montadas. Com essa tecnologia, nós vamos conseguir dinamizar como sua aplicação irá responder a uma requisição. Além disso, poderá utilizar diferentes tipos de resposta, como arquivos do tipo HTML, JSON, ou até mesmo imagem, através do

processamento da classe Servlet. Essas ações proporcionam uma maior interação entre o usuário e a aplicação que você desenvolveu, por exemplo.

Agora, você precisa ir evoluindo em sua aplicação. Pelo que foi estudado na aula 2, você já tem o seu projeto e ambiente configurados, prontos para serem customizados. Já conseguiu acessar uma página criada em seu sistema, através da página de boas-vindas. Neste momento, você precisa dar forma para seu projeto CatalogoDeFilmes. Então, suponha que sua loja virtual conte com alguns filmes já cadastrados e que precisam ser exibidos. Sefaz necessário que você crie mais uma vez um arquivo HTML chamado listadefilmes.html no mesmo diretório que o arquivo de boas-vindas.html, ou seja, dentro da pasta WebContent.

Você deve seguir o mesmo passo de criação realizado em listadefilmes.html visto em videoaulas anteriores. Lembre-se de escolher a opção New, e depois HTML File, pois o html já vem previamente formatado. No arquivo, listadefilmes.html, edite a tag title, inserindo a informação Lista de Filmes. Na tag body, você deve montar a sua lista de filmes contendo o nome do filme, gênero e ano de publicação. Para isso, realize as seguintes instruções no arquivo listadefilmes.html:

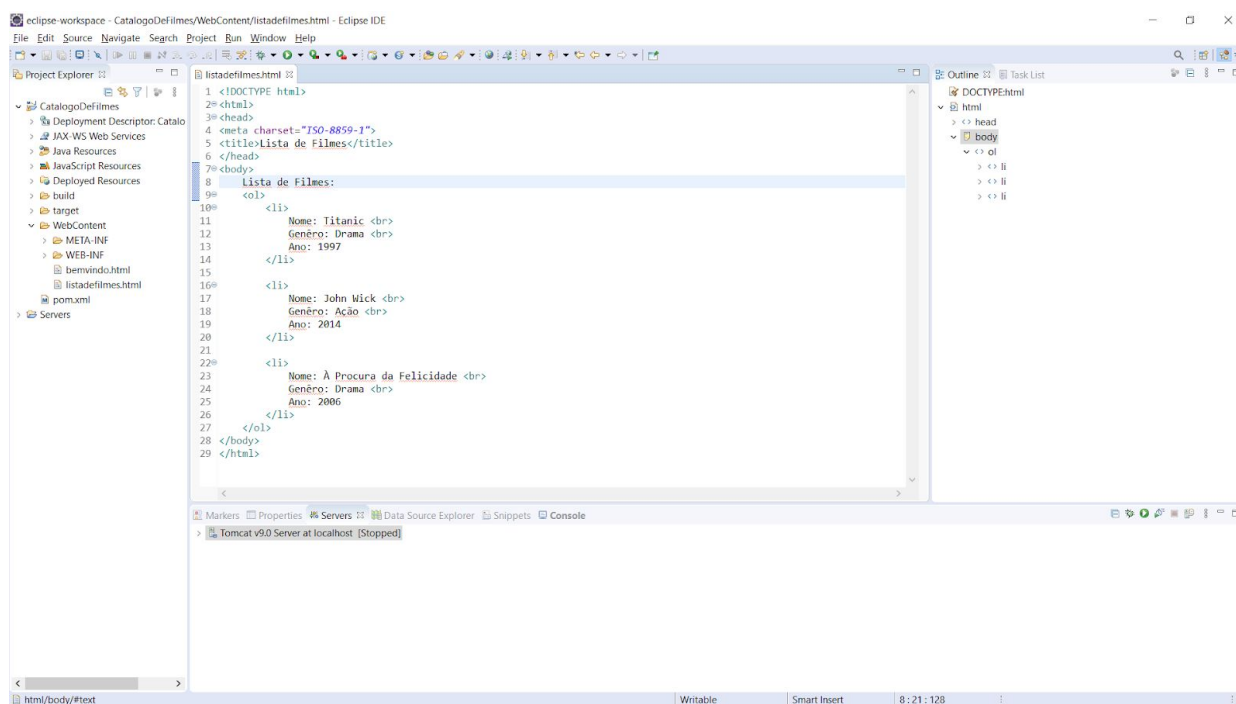
Passo 1 – Adicione um informativo à página que será exibida, escrevendo Lista de Filmes;

Passo 2 – Crie a tag , que é uma tag de criação de listas em HTML;

Passo 3 – Adicione os itens da sua lista. Para isso, use a tag ; adicione a tag ; e escreva Nome: Titanic, por exemplo. Agora, dê uma quebra de linha usando a tag
. Na linha abaixo, adicione outra informação importante a sua lista de filmes, o gênero, que tal? Portanto, escreva Gênero: Drama; dê outra quebra de linha com a tag
; e, na linha debaixo, informe o ano de publicação do filme, Ano: 1997. Em seguida, feche o primeiro item da sua lista, usando a tag . Repita esse processo,

inserindo mais dois filmes, John Wick, do gênero ação e ano 2014; e À Procura da Felicidade, do gênero drama e ano 2006;

Passo 4 –Com os itens da nossa lista criada, feche a tag de criação de listas, usando .



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Lista de Filmes</title>
6 </head>
7 <body>
8 <div>
9 <ol>
10 <li>
11 <div>
12 <div>Nome: Titanic <br>
13 <div>Gênero: Drama <br>
14 <div>Ano: 1997
15 </div>
16 </div>
17 </li>
18 <li>
19 <div>
20 <div>Nome: John Wick <br>
21 <div>Gênero: Ação <br>
22 <div>Ano: 2014
23 </div>
24 </div>
25 </li>
26 <li>
27 <div>
28 <div>Nome: À Procura da Felicidade <br>
29 <div>Gênero: Drama <br>
30 <div>Ano: 2006
31 </div>
32 </div>
33 </li>
34 </ol>
35 </div>
36 </body>
37 </html>
```

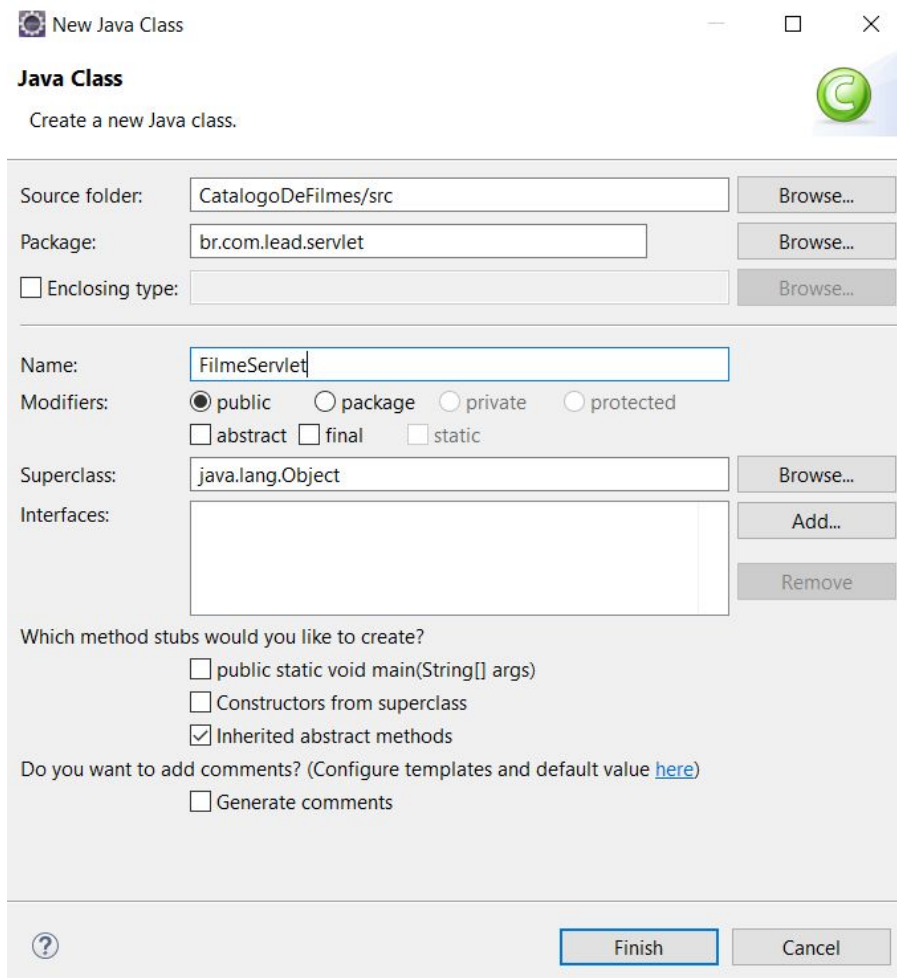
Agora que você criou e editou o arquivo listadefilmes.html, você deve executar a aplicação para avaliar o resultado. Feito isso, vá novamente em Servers e acione o start. Para acessar a página que você acabou de criar, vá até o seu navegador e insira o endereço <http://localhost:8080/CatalogoDeFilmes/listadefilmes.html>. Acessando a página, você deve encontrar a lista de filmes que acabou de montar.

← → ↻ 🏠 ⓘ localhost:8080/CatalogoDeFilmes/listadefilmes.html

Lista de Filmes:

1. Nome: Titanic
Genêro: Drama
Ano: 1997
2. Nome: John Wick
Genêro: Ação
Ano: 2014
3. Nome: À Procura da Felicidade
Genêro: Drama
Ano: 2006

Ao acessar sua página, você pode perceber que o conteúdo é apresentado conforme o esperado. Contudo, essa é uma página estática, e o que você precisa é transformá-la em uma página dinâmica. Para isso, você irá se utilizar das classes Servlets. Nesse sentido, você deve criar uma nova classe chamada FilmeServlet e adicioná-la ao pacote `br.com.lead.servlet`. Essa ação requer o uso do Eclipse. Assim, acesse o Eclipse, navegue até o seu projeto e, acessando o menu de opções, através do teclado pressionando a tecla Aplicação, navegue até a opção New e, a seguir, até Class.



New Java Class

Java Class

Create a new Java class.

Source folder: CatalogoDeFilmes/src Browse...

Package: br.com.lead.servlet Browse...

☐ Enclosing type: Browse...

Name: FilmeServlet

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

Com essa classe FilmeServlet criada, você deve editá-la. Nesse momento, a edição irá ocorrer da seguinte maneira: a classe FilmeServlet deve estender HttpServlet, pois HttpServlet é a classe que irá definir o comportamento da classe FilmeServlet. Dessa maneira, a sua implementação irá trabalhar em cima do protocolo HTTP. É importante lembrar que toda classe Servlet deve ser mapeada para um padrão de URL. Por exemplo, na sua aplicação, você deve utilizar o padrão /filme. Assim, as requisições do tipo <http://localhost:8080/CatalogoDeFilmes/filme> serão enviadas para a sua Servlet. Para realizar esse mapeamento, utilize a anotação

@WebServlet("/filme") que fica logo acima da especificação da sua classe, ou seja, acima da declaração do nome da sua classe.

```
@WebServlet("/filme")
public class FilmeServlet extends HttpServlet {
```

Ah, uma curiosidade é que, em versões antigas do Java e da tecnologia de Servlets, realizar esse mapeamento em um arquivo de configuração chamado web.xml demandava mais esforço e era mais propenso a erros, pois o mapeamento era feito em um arquivo à parte, que centralizava todos os mapeamentos e, caso sua aplicação tivesse várias classes Servlets, esse arquivo ficaria com diversas configurações. Contudo, atualmente, esse passo se tornou mais prático através do uso da anotação @WebServlet.

Feito isso, você deve sobrescrever o método service. Este método é responsável por receber uma requisição e tratá-la. Além disso, o método service recebe, por parâmetro, dois objetos do tipo HttpServletRequest e HttpServletResponse.

```
1 package br.com.lead.servlet;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletException;
6 import javax.servlet.annotation.WebServlet;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 @WebServlet("/filme")
12 public class LivroServlet extends HttpServlet {
13
14     @Override
15     protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
16
17     }
18 }
19
```

O objetivo de implementar esse método em sua aplicação é torná-la mais dinâmica. Para enviar sua lista de filmes para o usuário, quando for requisitado através da sua

aplicação, você deve utilizar o conceito de orientação a objetos. Para isso, crie a classe Filme, que é a responsável por representar os seus filmes, utilizando o eclipse. Crie a classe Filme, mas a deixe em outro pacote, isto é, adicione ao pacote br.com.lead.modelo. Essa ação é necessária a fim de deixar sua aplicação mais organizada.

Com a classe criada, adicione os atributos, que são nome do tipo String, gênero do tipo String e ano do tipo Integer. Agora, você irá criar um construtor que receberá os dados referentes ao nome, gênero e ano. A seguir, você deverá atribuir a esses dados recebidos os seus respectivos atributos. Feito isso, você deve criar os métodos Get para cada atributo. Assim, será possível capturar o valor de cada atributo.

Pronto, sua classe Filme foi criada!

```
1 package br.com.lead.modelo;
2
3 public class Filme {
4
5     private String nome;
6     private String genero;
7     private Integer ano;
8
9     public Filme(String nome, String genero, Integer ano) {
10         this.nome = nome;
11         this.genero = genero;
12         this.ano = ano;
13     }
14
15     public String getNome() {
16         return nome;
17     }
18
19     public String getGenero() {
20         return genero;
21     }
22
23     public Integer getAno() {
24         return ano;
25     }
26 }
27
```

No próximo vídeo, você irá aprender a como utilizar essa estrutura criada, para poder exibir os dados dos filmes para o usuário que realizar uma requisição a sua aplicação.

Até lá! Bons estudos!