

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

Vídeo 3.2: Exibindo objetos criados para o usuário

Olá, tudo bem? Seja bem-vindo(a)!

Nesta videoaula, você irá aprender a criar objetos que farão parte de uma lista e exibir esses dados para o usuário.

Dando continuidade à aula anterior, vamos até a classe FilmeServlet. Nela, você irá criar três objetos do tipo Filme, que farão parte da sua lista de filmes. Os objetos vão representar os filmes Coringa, do gênero Drama e do ano 2019; Matrix, do gênero Ação e do ano 1999; e Forrest Gump, do gênero Drama e do ano 1994. Para isso, utilize os conceitos estudados na videoaula anterior. Em caso de dúvida, reveja a videoaula 3.1.

Com os objetos criados, você irá criar uma lista, do tipo ArrayList do Java, e adicionar os objetos de filme a essa lista.

```
@Override
protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    Filme coringa = new Filme("Coringa", "Drama", 2019);
    Filme matrix = new Filme("Matrix", "Ação", 1999);
    Filme forrestGump = new Filme("Forrest Gump", "Drama", 1994);

    ArrayList<Filme> filmes = new ArrayList<Filme>();

    filmes.add(coringa);
    filmes.add(matrix);
    filmes.add(forrestGump);
}
```

Com a lista preenchida, você irá criar uma página HTML para adicionar esses filmes como uma lista a ser exibida. Você irá utilizar nesse processo os objetos recebidos como parâmetro do método service, que são req e resp, dos tipos HttpServletRequest e HttpServletResponse, respectivamente. Siga os passos descritos a seguir.

Inicialmente, você precisa informar o tipo de resposta que a lista irá retornar. Para isso, adicione a seguinte instrução `resp.setContentType("text/HTML")`. Agora, você irá criar um objeto chamado out, do tipo PrintWriter e irá atribuir a este objeto o conteúdo proveniente de `resp.getWriter()`, em que resp é o objeto recebido por parâmetro do método service. Com esse objeto criado, você irá conseguir montar o seu HTML de resposta.

Para escrever uma nova linha nesse HTML, utilize o método `out.println`. Para adicionar uma descrição, utilize a seguinte instrução `out.println("<h2>Lista de Filmes, utilizando Servlets:</h2>")`. Em seguida, adicione uma tag utilizada para criar listas denominada `out.println("")`. Com a lista aberta, será necessário adicionar os itens à lista. Esses itens correspondem aos seus filmes. Agora, você deve criar uma instrução do tipo "for" para percorrer essa lista. Para cada item da lista, utilize o método `out.println` três vezes, isto é, um para informar o nome do filme; outro para informar o gênero; e, por fim, a informação do ano. No início do método for, você deve adicionar a tag ``, que indica os itens da lista. Ao final, o método for

deve ser fechado com a tag ``. Após a instrução `for`, você deve fechar a tag `` e fechar o objeto `out`, utilizando o método `out.close()`.

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

    @Override
    protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        Filme coringa = new Filme("Coringa", "Drama", 2019);
        Filme matrix = new Filme("Matrix", "Ação", 1999);
        Filme forrestGump = new Filme("Forrest Gump", "Drama", 1994);

        ArrayList<Filme> filmes = new ArrayList<Filme>();

        filmes.add(coringa);
        filmes.add(matrix);
        filmes.add(forrestGump);

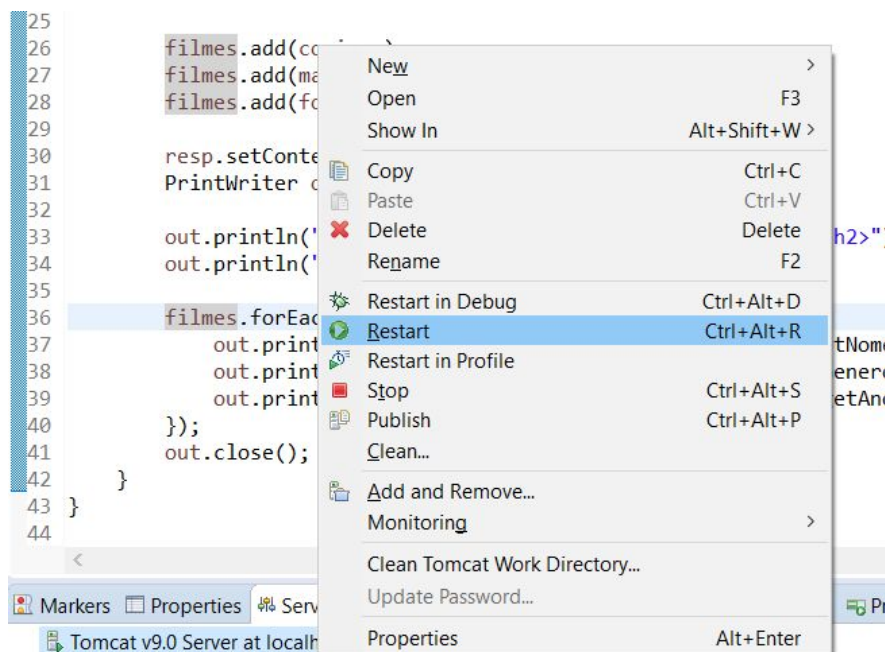
        resp.setContentType("text/HTML");
        PrintWriter out = resp.getWriter();

        out.println("<h2>Lista de Filmes utilizando Servlets:</h2>");
        out.println("<ol>");

        filmes.forEach(filme -> {
            out.println(String.format("<li> Nome: %s", filme.getNome()));
            out.println(String.format(" Genero: %s", filme.getGenero()));
            out.println(String.format(" Ano: %s </li>", filme.getAno()));
        });
        out.close();
    }
}
```

Pronto! Você acabou de montar a sua primeira página HTML, utilizando Servlet!

Para conferir o resultado, você deve iniciar o tomcat. Localize, novamente, a aba Servers do Eclipse, onde tem o servidor Tomcat configurado. Com o seu servidor selecionado, acesse o menu de opções, pode ser pelo teclado, utilizando a tecla Aplicação, e escolha a opção Restart.



Isso é necessário sempre que você criar uma nova estrutura no seu projeto, e como foi criado uma Servlet nova, então, se faz necessário.

Com o Tomcat reiniciado, você deve acessar o endereço <http://localhost:8080/CatalogoDeFilmes/filme> através do navegador web. Esse é o endereço que você mapeou na sua Servlet. Nesse momento, a página web que deve ser exibida é a página que você acabou de montar dentro da sua Servlet, contendo sua lista de filmes.



Lista de Filmes utilizando Servlets:

1. Nome: Coringa Genero: Drama Ano: 2019
2. Nome: Matrix Genero: Ação Ano: 1999
3. Nome: Forrest Gump Genero: Drama Ano: 1994

Até agora, você criou páginas que exibem todos os filmes inseridos na sua lista. Avançando em seu processo de aprendizagem, suponha que você não queira visualizar todos os filmes, ou seja, que você deseje filtrá-los pelo gênero drama, por exemplo. Assim, você precisa informar, para sua requisição, essa informação. Isso ocorre através do denominado passar parâmetro na URL. Esse parâmetro corresponde à drama e ao seu conteúdo.

Para passar um parâmetro em uma URL, você deve utilizar o caractere ? seguido do nome do parâmetro e do seu valor, deve ser feita da seguinte maneira: ?genero=drama. Feito isso, a URL ficará <http://localhost:8080/CatalogoDeFilmes/filme?genero=drama>. Entretanto, ao acessar essa URL, agora, você perceberá que a mesma lista é exibida, nada foi alterado. Isso decorre porque não foi adicionado a lógica para filtrar os filmes de acordo com o parâmetro recebido na sua Servlet, ou seja, você passou os parâmetros, mas não o utilizou para realizar o filtro dos filmes. Então, adicione essa regra para filtrar os filmes de acordo com o parâmetro utilizado.

Em primeiro lugar, você deve resgatar o valor do parâmetro que veio na URL. Para isso, utilize o método `getParameter` de `req`, informando o nome do parâmetro que você deseja resgatar, entre parênteses e aspas duplas. O retorno desse método é uma `String` e deverá ficar da seguinte maneira: `String genero = req.getParameter("genero")`. Com o valor do parâmetro em mãos, você irá filtrar a sua lista por esse parâmetro. Aqui, você tem algumas possibilidades, utilize o `Stream` do Java por ser mais otimizado e poder deixar o filtro em apenas uma linha, mas você pode realizar o filtro como preferir, lembrando-se de padronizar os valores para `Upper Case`, para evitar problemas de comparação entre letras maiúsculas e minúsculas.

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

    @Override
    protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        Filme coringa = new Filme("Coringa", "Drama", 2019);
        Filme matrix = new Filme("Matrix", "Ação", 1999);
        Filme forrestGump = new Filme("Forrest Gump", "Drama", 1994);

        ArrayList<Filme> filmes = new ArrayList<Filme>();

        filmes.add(coringa);
        filmes.add(matrix);
        filmes.add(forrestGump);

        String genero = req.getParameter("genero");

        resp.setContentType("text/HTML");
        PrintWriter out = resp.getWriter();

        out.println("<h2>Lista de Filmes utilizando Servlets:</h2>");
        out.println("<ol>");

        filmes.stream().filter(filme -> filme.getGenero().toUpperCase().equals(genero.toUpperCase())).forEach(filme -> {
            out.println(String.format("<li> Nome: %s", filme.getNome()));
            out.println(String.format(" Genero: %s", filme.getGenero()));
            out.println(String.format(" Ano: %s </li>", filme.getAno()));
        });
        out.close();
    }
}
```

Feito isso, você implementou a regra. Agora, acesse o endereço novamente <http://localhost:8080/CatalogoDeFilmes/filme?genero=drama> e Perceba que ele exibirá uma página apenas com os filmes do gênero drama.

← → ↻ 🏠 ⓘ localhost:8080/CatalogoDeFilmes/filme?genero=drama

Lista de Filmes utilizando Servlets:

1. Nome: Coringa Genero: Drama Ano: 2019
2. Nome: Forrest Gump Genero: Drama Ano: 1994

O que você acabou de realizar agora foi uma requisição do tipo GET, passando parâmetros. No protocolo HTTP, você pode passar parâmetros de outra maneira, sem que o parâmetro fique visível na URL, mas, nesse caso, você deve utilizar a requisição do tipo POST. Nesse método, os dados enviados como parâmetros vão no corpo da requisição e não ficam visíveis na URL. Como as requisições que você está fazendo agora não necessitam de passagem de parâmetros no corpo da

requisição, então, você está utilizando apenas o modelo do tipo GET. Contudo, existem outros métodos no protocolo HTTP, mas essa abordagem você estudará mais à frente no curso.

Na sua classe Servlet, o método `service` consegue trabalhar tanto com GET quanto com POST, porém você deve trabalhar com requisições do tipo GET e POST de forma separada. Para isso, basta que você sobrescreva os métodos `doGet` e `doPost` para trabalhar com GET e POST de maneira independente um do outro. Lembrando que esses métodos recebem os mesmos parâmetros que o método `service`, certo?

Portanto, nesta videoaula, você conheceu a tecnologia Servlet, uma das pioneiras e bastante utilizada para o desenvolvimento de páginas dinâmicas; aprendeu a implementar uma classe Servlet de exemplo; conheceu uma maneira de passar parâmetros para o seu servidor, através da URL; e viu um pouco da diferença de uma requisição GET e POST.

Nos próximos vídeos, você irá evoluir mais ainda com o seu projeto, através do uso de outras tecnologias. Estas irão te proporcionar uma maior flexibilidade no desenvolvimento da sua aplicação. Como exemplo, tem-se o JSP. Com ele, não será mais necessário montar o HTML dentro da sua Servlet, mas isso você conhecerá no próximo vídeo.

Até a próxima videoaula!