



ROTEIRO DE LABORATÓRIO - JSP

Java Server Pages (JSP) é uma tecnologia que permite a criação de aplicações web dinâmicas. Esta tecnologia tem acesso a toda família de APIs do Java, incluindo a API JDBC para acesso a bases de dados.

O ciclo de vida de um JSP

O ciclo de vida de um JSP pode ser definido como um processo completo desde sua criação até sua destruição, o que é bastante similar ao ciclo de vida de um servlet. Os seguintes passos são seguidos por um JSP.

- Compilação
- Inicialização: Quando o container carrega um JSP, este invoca o método `jspInit()` antes de atender a qualquer requisição. Se for necessário uma implementação específica do método `jspInit()`, este método deverá ser reescrito.

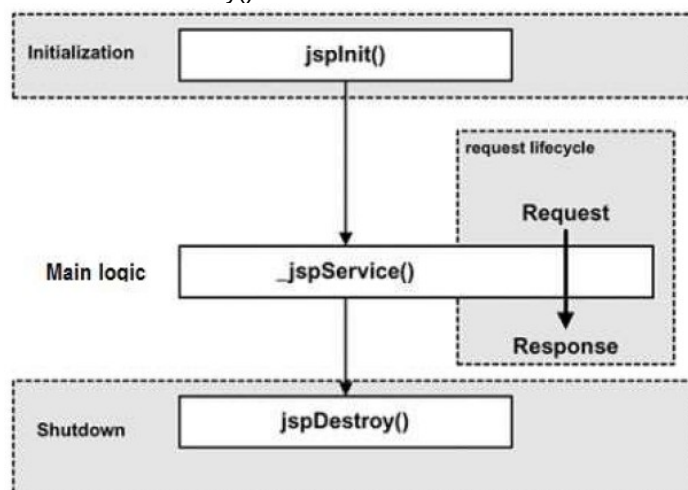
```
public void jspInit(){  
    // Initialization code...  
}
```

- Execução: Esta fase representa todas as interações de requisição-resposta até o JSP ser destruído. Quando um browser solicita uma página JSP e esta foi corretamente carregada e inicializada, o JSP engine invoca o método `_jspService()`. Este método recebe um `HttpServletRequest` e um `HttpServletResponse` como parâmetro, conforme mostrado a seguir:


```
void _jspService(HttpServletRequest request,  
                  HttpServletResponse response)  
{  
    // Service handling code...  
}
```

Observação: Este método não pode ser reescrito.

- Cleanup: Semelhante ao método `destroy()` de um servlet

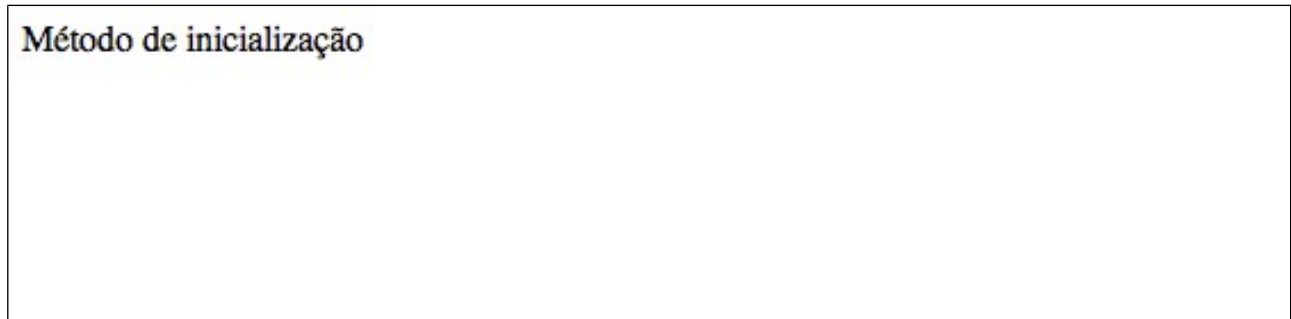


Uma rápida prática

Usando o eclipse, crie um novo  **Dynamic Web Project**. Neste, crie uma nova página JSP. Nosso objetivo neste momento é testar os métodos que compõem o ciclo de vida de um JSP. Agora, no arquivo recém criado, digite o código a seguir:

```
<body>
<%!
    String frase;
    public void jspInit(){
        frase = "Executando método de inicialização";}
%>
<% out.println(frase);%>
</body>
```

Agora, execute nossa aplicação e visualize o resultado. Sua tela deve estar semelhante a tela a seguir.



Sintaxe JSP

Um scriptlet pode conter qualquer sentença Java, variáveis ou declaração de métodos, ou até mesmo expressões que sejam válidas. A seguir é apresentada a sintaxe do Scriptlet:

```
<% code fragment %>
```

Agora, vamos testar nosso “segundo” scriptlet. Já usamos scriptlets na prática anterior. Na página anteriormente criada, digite o código seguinte:

```
<body>
Hello World!<br/>
<% out.println("Seu endereço IP é: " + request.getRemoteAddr()); %>
</body>
```

Quando você executar sua aplicação você deverá visualizar o endereço IP da máquina que originou a requisição.

Declarações JSP

Declarações são utilizadas para declarar uma ou mais variáveis ou métodos que podem ser utilizadas em seu código Java posteriormente. Você deve declarar todas as variáveis

e métodos antes que estes possam ser utilizadas no seu arquivo JSP. A seguir é apresentada a sintaxe de declaração utilizada por arquivos JSP.

```
<%! declaration; [ declaration; ]+ ... %>
```

Expressões em JSP

Um elemento de expressão em JSP contém uma expressão que deve ser avaliada, convertida em String e por fim inserida no local onde aparece sua marcação no arquivo JSP. A seguir é apresentada a sintaxe de expressões em JSP.

```
<%= expression %>
```

Agora, um exercício proposto.

Crie uma página JSP que escreva em tela o nome do dia da semana. Esta escrita deve ser dinâmica, ou seja, se hoje for sexta-feira, o resultado deve ser o seguinte:

Hoje é sexta

Caso hoje seja sábado, o resultado deve ser o seguinte:

Hoje é sábado

Sugestão: Escreva sua aplicação em Java para desktop e adapte-a para JSP.