

Get e Post Formulários Web

Objetivo da Aula

1. Criar um formulário.
2. Obter os dados de um formulário no JSP.
3. Entender quando utilizar o Get e Post

Introdução

Hoje a maior parte dos sites trabalham muito com dados, desde um cadastro de cliente até um feedback dado pelo cliente. Nesta aula aprenderemos a como criar páginas para obtenção de dados para a nossa Aplicação.

Formulário

Um formulário contém todas as informações que queremos que sejam processadas no nosso Servidor Web. Esses dados são enviados através de uma Submissão “*submit*”.

O HTML possui a tag “*<form>*” para montarmos um formulário.

Criando o formulário

```
<body>
  <form action="texto.jsp">
    Nome:<br/>
    <input name="nome" /><br/>

    Sobrenome:
    <br/><input name="sobrenome" /><br/>

    Email:<br/>
    <input name="email" /><br/><br/>

    <input type="submit" value="Gravar" />
  </form>
</body>
```

1. Utilizamos a tag “*<form>*” para enviar dados para o servidor. São apenas enviados para o servidor os dados dentro desta tag.
2. O “*action*” na tag “*<form>*” indica para onde estamos mandando o formulário, no nosso caso, estamos enviando para uma página JSP do servidor.
3. Dentro das tags “*<input>*” estamos utilizando utilizando o atributo “*name*”. Utilizamos este atributo para dar um nome ao campo da nossa página. Todo campo com dados precisa ter um nome para que o JSP identifique ele.

Obtendo os dados de um formulário no JSP

```
<body>
    <%
        String nome;
        String sobrenome;
        String email;

        nome = request.getParameter("nome");
        sobrenome = request.getParameter("sobrenome");
        email = request.getParameter("email");
    %>

    Nome: <%=nome%><br/>
    Sobrenome: <%=sobrenome%><br/>
    Email: <%=email%><br/>
</body>
```

1. Todo formulário chega como uma requisição no Servidor Java.
2. Toda Página JSP possui uma variável chamada “request” que controla as requisições.
3. Obtemos um dado do formulário através do método “getParameter”, passando como parâmetro o nome do campo do formulário.

Lembre-se

O parâmetro passado para o método “getParameter” do “request” tem que ser exatamente igual ao nome do campo no HTML.

```
<input name= "email" /><br/><br/>

sobrenome = request.getParameter( "email" );
```

O “getParameter” retorna o valor sempre como “String”, logo quando precisamos do valor como um valor numérico precisaremos utilizar um conversor

Obtendo valor numérico

Sabemos que o “getParameter” retorna os valores de um campo do formulário como “String”, mas muitas vezes iremos precisar utilizar o valor como “int”. Para isso precisaremos converter o dado.

Segue o exemplo do formulário:

```

<body>
  <form action="numero.jsp">
    Num:<br/>
    <input name="num1" /><br/>
    <input type="submit" value="Enviar" />
  </form>
</body>

```

A página JSP:

```

<body>
  <%
    int num;

    num = Integer.parseInt(request.getParameter("num1"));

  %>
  Num:<%=num%><br/>
</body>

```

1. Não podemos fazer “num = request.getParameter(“num1”);” pelas diferenças dos tipos.
2. O “Integer” é um conversor de dados. Tanto de “String” para “int” e vice versa.

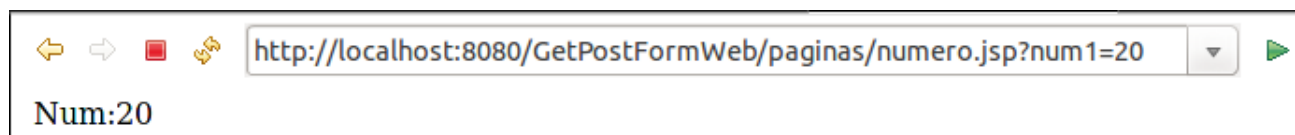
Exercícios

1. Monte um formulário HTML onde preencheremos informações sobre um Computador. O formulário deve ter no mínimo 5 dados. Este formulário deve ser mandado para uma página JSP que irá obter os dados e mostrá-los na página.
2. Monte uma página em HTML que tenha 2 campos dentro de um formulário. Cada campo recebe um número. O formulário deve passar esses números para uma página JSP, esta página deve obter os dados e mostrá-los junto com a soma dos 2.
3. Monte um formulário HTML que tenha 2 campos numéricos, esses dados devem ser passados para uma página JSP que indicará quais dos dois números é o maior.
4. Monte uma calculadora web utilizando HTML e JSP. A calculadora deve dar a opção para o usuário escolher qual operação irá realizar. A calculadora deve apresentar 4 operações no mínimo.

Métodos GET e POST

Quando montamos um formulário podemos utilizar o método “GET” ou o “POST” para enviar os dados.

A primeira vista a diferença entre os 2 é que quando utilizamos o método “GET” os valores presentes no formulário estarão na URL.



1. Nota-se que temos na nossa URL “?num1=20” que são valores digitados no formulário.
2. Agora imagine se fosse um cadastro de usuário, onde ele coloca a senha, seria estranho a senha aparecer na url.

Existem outras diferenças entre “**GET**” e “**POST**”, a maioria delas no protocolo HTTP. A pergunta é quando utilizamos uma ou outra.

GET

Basicamente utilizamos o “**GET**” para obter dados ou fazer uma pesquisa, pois aí não existe o problema de uma “identificação” aparecer na tela. A questão é, você não está fazendo nenhuma mudança no servidor.

POST

Utilizamos o “**POST**” para enviar dados para serem processados, como por exemplo adicionar um usuário. Sendo assim, você estará mudando algum dado no servidor.

Fontes

Livros : Use a Cabeça! Java.
Use a Cabeça! Servlets & JSP.

Sites: www.guj.com.br
<http://www.if.ufrgs.br/~betz/jaulas/aula2.htm>
<http://www.caelum.com.br/apostila-java-web/jaserver-pages/>