

# Aula 4 Servlet

Disciplina: Tecnologia da Web

Professor: Flavio Barbieri Gonzaga



## A classe HttpServlet

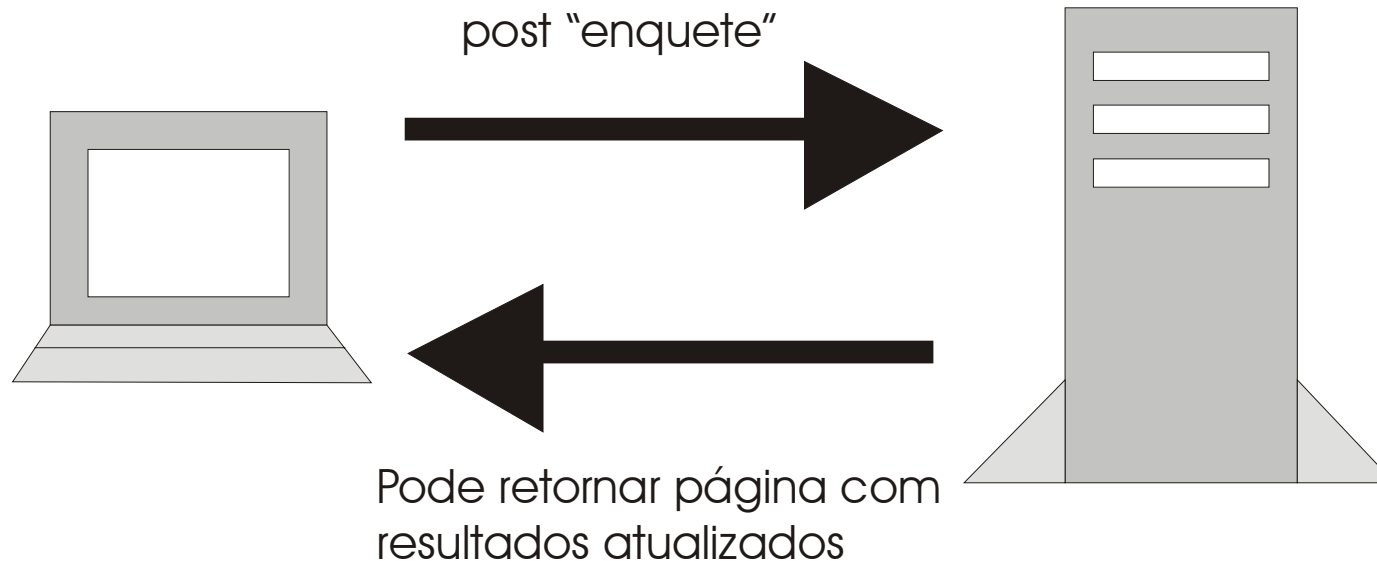
- Como já vimos, a classe HttpServlet estende a classe GenericServlet.
- Dos vários métodos existentes na mesma, os mais comuns e utilizados são os métodos doGet e doPost.

# A classe HttpServlet – método doGet



- Método doGet geralmente é utilizado quando uma página é requisitada.
- Utilizando o método doGet, o navegador geralmente busca pela página na pasta temporário. Caso a mesma esteja presente, é carregada.

# A classe HttpServlet – método doPost



- Método doPost geralmente é utilizado para enviar dados ao Servidor. Ex: Escolha de uma opção em uma enquete.
- Utilizando o método doPost, o navegador **não** busca pela página na pasta temporário. Pois a página resultante muda constantemente.

# Código utilizando HttpServlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class GetPost extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<HTML>");
        out.println("<HEAD>");
        out.println("<TITLE>Metodo Get</TITLE>");
        out.println("</HEAD>");
        out.println("<BODY>");
        out.println("<FORM method=post>");
        out.println("<BR> Primeiro Nome: <INPUT type=text name=primeiroNome>");
        out.println("<BR> Ultimo Nome: <INPUT type=text name=ultimoNome>");
        out.println("<BR><INPUT type=submit value=Enviar!>");
        out.println("</FORM>");
        out.println("</BODY>");
        out.println("</HTML>");
        out.close();
    }
}
```

# Código utilizando HttpServlet

```
public void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String Nome = request.getParameter("primeiroNome");

    out.println("<HTML>");
    out.println("<HEAD>");
    out.println("<TITLE>Metodo Post</TITLE>");
    out.println("</HEAD>");
    out.println("<BODY>");
    out.println("<h1>Post recebido!</h1>");
    out.println("Primeiro nome: " + Nome + "<BR>");
    Nome = request.getParameter("ultimoNome");
    out.println("Ultimo nome: " + Nome);
    out.println("<HR>");
    out.println("</BODY>");
    out.println("</HTML>");
    out.close();
}
}
```

# Entendendo o Código

```
import javax.servlet.*;  
import javax.servlet.http.*;  
import java.io.*;
```

- Importando o pacote `javax.servlet` estamos importando a classe `ServletException`.
- Importando o pacote `javax.servlet.http` estamos importando a classe `HttpServlet`
- Importando o pacote `java.io` estamos importando a classe `IOException`

# Entendendo o Código

```
public class GetPost extends HttpServlet {
```

- Declarando a classe GetPost como sendo extendida da classe HttpServlet

# Entendendo o Código

```
public void doGet(HttpServletRequest request,  
                  HttpServletResponse response)
```

- Sobrecarga do método doGet.
- O método possui os parâmetros do tipo HttpServletRequest e HttpServletResponse.
- Os parâmetros carregam dados da requisição enviada pelo cliente e resposta do servidor respectivamente.

# Entendendo o Código

```
throws ServletException, IOException {
```

- Podemos tratar exceções ocorridas caso o servlet não consiga atender a requisição pedida (classe ServletException), ou exceções ocorridas na leitura/gravação de dados no cliente (classe IOException).

# Entendendo o Código

```
response.setContentType("text/html");
```

- Nessa linha dizemos que a resposta gerada será do tipo documento html.

# Entendendo o Código

```
PrintWriter out = response.getWriter();
```

- Criamos o objeto out do tipo PrintWriter. Esse objeto estará gerando a saída de código em html para o cliente.

# Entendendo o Código

```
out.println("<HTML>");  
out.println("<HEAD>");  
out.println("<TITLE>Metodo Get</TITLE>");  
out.println("</HEAD>");  
out.println("<BODY>");
```

- Início do código html.

# Entendendo o Código

```
out.println("<FORM method=post>");
```

- Definimos que a página html gerada terá um form que chamará o método post quando acionado.

# Entendendo o Código

```
out.println("<BR> Primeiro Nome: <INPUT type=text name=primeiroNome>");  
out.println("<BR> Ultimo Nome: <INPUT type=text name=ultimoNome>");
```

- O form terá dois campos do tipo text. Sendo que os campos serão passados como parâmetros com os nomes *primeiroNome* e *ultimoNome*.

# Entendendo o Código

```
out.println("<BR><INPUT type=submit value=Enviar!>");
```

- Definindo um botão com o rótulo Enviar!

# Entendendo o Código

```
out.println("</FORM>");  
out.println("</BODY>");  
out.println("</HTML>");  
out.close();
```

- Finalizando o código html e fechando o fluxo para completar a página.

# Entendendo o Código

```
out.println("</FORM>");  
out.println("</BODY>");  
out.println("</HTML>");  
out.close();
```

- Finalizando o código html e fechando o fluxo para completar a página.

# Entendendo o Código

```
String Nome = request.getParameter("primeiroNome");

out.println("<HTML>");
out.println("<HEAD>");
out.println("<TITLE>Metodo Post</TITLE>");
out.println("</HEAD>");
out.println("<BODY>");
out.println("<h1>Post recebido!</h1>");
out.println("Primeiro nome: " + Nome + "<BR>");
Nome = request.getParameter("ultimoNome");
out.println("Ultimo nome: " + Nome);
out.println("<HR>");
out.println("</BODY>");
out.println("</HTML>");
out.close();
```

- O método post funciona de maneira semelhante, com o acréscimo da String Nome que é a responsável por receber os parâmetros passados nos campos *primeiroNome* e *ultimoNome* no método get.

## Exercício

- Crie um servlet que utilizando dos métodos get e post simule uma página de login.
  - Caso o usuário informe os dados corretos, o mesmo poderá ter acesso à uma página;
  - Caso contrário, uma mensagem de erro deve ser exibida, com a opção para o usuário tentar novamente.
  - Dica: utilize o método `response.sendRedirect()`;

## Bibliografia

- Java para a Web com Servlets, JSP e EJB – 1ª Edição. Budi Kurniawan – Editora Ciência Moderna
- Java como Programar – 6ª Edição. H. M. Deitel, P. J. Deitel – Editora Pearson Prentice Hall
- [www.argonavis.com.br](http://www.argonavis.com.br) (Helder da Rocha)