

## Lab 04 - Servlet Básico

Neste laboratório vamos exercer características básicas do Servlet

### Exercícios

**Exercício 1:** Construir e executar a aplicação **SaudacaoServlet**.

**Exercício 2:** Criar uma aplicação que exibe informações do objeto **HttpRequest**, o que reflete a solicitação **HTTP** do cliente.

### Exercício 1 - Construir e executar a aplicação SaudacaoServlet

#### 1.1. Aplicação SaudacaoServlet e RespostaServlet

1. Crie um novo **Dynamic Project Web** chamado Hello.
2. Crie um **Servlet** com o nome de **SaudacaoServlet.java** e implemente o método **doGet()**, escreva o código abaixo na implementação do método:

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    // then write the data of the response
    out.println("<html>" + "<head><title>Hello</title></head>");

    // then write the data of the response
    out.println("<body bgcolor=\"#ffffff\">" +
        "<h2>Olá, meu nome é Servlet. Qual é o seu nome?</h2>" +
        "<form method=\"get\">" +
        "<input type=\"text\" name=\"nome\" size=\"25\">" + "<p></p>" +
        "<input type=\"submit\" value=\"Enviar\">" +
        "<input type=\"reset\" value=\"Resetar\">" + "</form>");

    String nome = request.getParameter("nome");
    if ((nome != null) && (nome.length() > 0)) {
        out.println("<h2>Olá, " + nome + "!");
    }
    out.println("</body></html>");
    out.close();
}
```

3. Execute a aplicação no servidor invocando a **URL** do servlet que acabou de ser criado, **SaudacaoServlet.java**, e veja o resultado.



## 1.2. Modificar o servlet para exibir o hobby

1. Dê um duplo clique sobre **SaudacaoServlet.java** dentro de **JavaResources:src -> default package -> SaudacaoServlet** para abri-lo no editor.
2. Modifique o método **doGet()** para que apareça mais um campo para o usuário informar o seu hobby e o mesmo seja mostrado junto com a resposta conforme imagem abaixo:



## Exercício 2 - Criar uma aplicação com que exibe informações do objeto *HttpRequest*, o que reflete a solicitação *HTTP* do cliente.

1. Crie um novo **Servlet** chamado **CapturaInformacoes.java** que implemente o método **doGet()** e **doPost()** conforme o código abaixo. Em seguida, execute o Servlet para ver o resultado. Perceba que o método **doPost()** foi implementado para chamar o método **doGet()**, ou seja, qualquer tipo de requisição para este servlet será executado o método **doGet()**. Veja na imagem abaixo o resultado que deve ser mostrado, dados retirados do objeto **HttpServletRequest** que é criado na requisição do usuário com o servidor.

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Informação de Requerimento</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h3>Informação de Requerimento</h3>");
    out.println("Method: " + request.getMethod() + "<br>");
    out.println("Request URI: " + request.getRequestURI() + "<br>");
    out.println("Protocol: " + request.getProtocol() + "<br>");
    out.println("PathInfo: " + request.getPathInfo() + "<br>");
    out.println("Remote Address: " + request.getRemoteAddr() + "<br>");
    out.println("</body>");
    out.println("</html>");
}

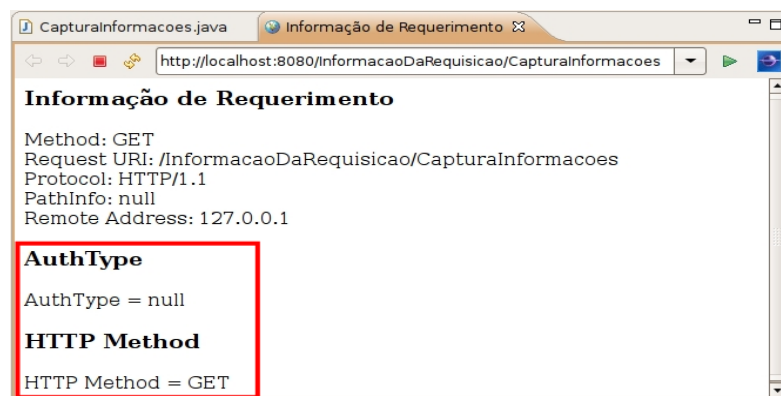
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
    doGet(request, response);
}
```



3. Modifique o servlet **CapturaInformacoes.java** inserindo o código abaixo logo após a linha que mostra o **Remote Address**.

```
// Exibir tipo de autenticação
out.println("<h3> AuthType </h3>");
out.println("AuthType = " + request.getAuthType() + "<br>");
// Exibir método HTTP
out.println("<h3> HTTP Method </h3>");
out.println("HTTP Method = " + request.getMethod() + "<br>");
```

4. Execute o servlet e veja o resultado apresentado.



5. Altere a primeira linha do método **doGet()**, onde é atribuído o valor de **ContentType** conforme abaixo e veja o que vai acontecer quando o usuário tentar acessar o servlet.

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

6. Execute a aplicação e tente acessar a página do servlet novamente. Deverá aparecer algo parecido com a imagem abaixo, ou seja, a página não será mostrada pelo navegador, mas aparecerá como um download a ser feito. Isso ocorre porque o tipo **application** não é um tipo conhecido pelo navegador a ser traduzido. Se você salvar o arquivo e abri-lo no bloco de notas irá ver todo o código **HTML** normal.

