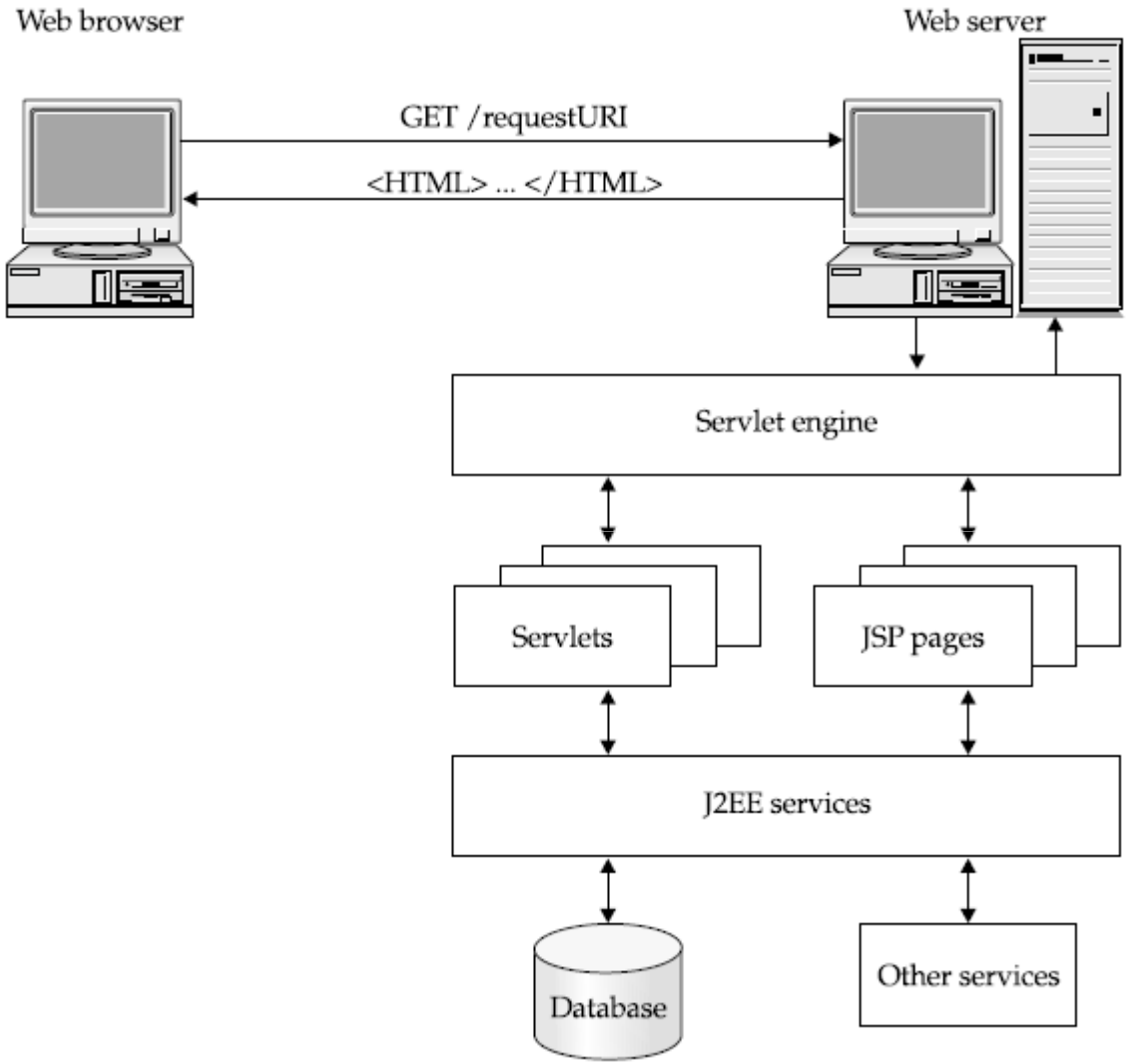
Programação para Web I
Aula 11

Profa. Silvia Bertagnolli

Servlets

Servlets e Páginas JSP



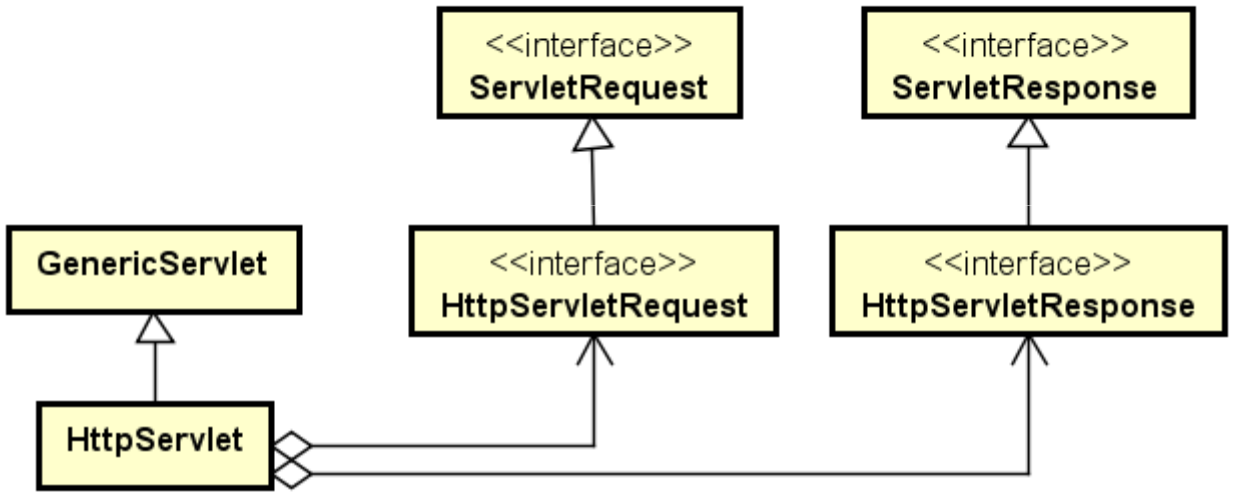
Servlets: principais classes

HttServletRequest - responsável pelo processamento das requisições

HttServletResponse - responsável pelo envio das respostas

HttpSession - classe que representa uma sessão

API: Servlet HTTP



Servlet como Controller

Os servlets podem controlar todo o fluxo das requisições e as chamadas das páginas

Para isso é usada a classe RequestDispatcher e os métodos forward e include

Página index.html

O que acontece quando o usuário clica no botão Entrar?

```
<form action="Controlador" method="post">
    <label ><strong>E-mail:</strong>
    </label> <input type="text" name="email" value="" />
    <br />
    <label><strong>Senha:</strong>
    </label> <input type="password" name="senha"  value="" />
    <input type="hidden" name="logica" value="LoginLogout" />
    <br />
    <input  name="entrar" type="submit"  value="Entrar" />
</form>
```

Servlet Controlador

Servlet Controlador at /Servlets_Aula2

Página index1.html

```
<form action="Controlador1" method="post">
    <label ><strong>E-mail:</strong>
    </label> <input type="text" name="email" value="" />
    <br />
    <label><strong>Senha:</strong>
    </label> <input type="password" name="senha"  value="" />
    <input type="hidden" name="logica" value="LoginLogout" />
    <br />
    <input  name="entrar" type="submit"  value="Entrar" />
</form>
```

Servlet Controlador1

```
String email = request.getParameter("email");
String senha = request.getParameter("senha");
if(email!=null && senha!=null){
       request.getRequestDispatcher("cadPessoa.html").
                                    forward(request, response);
}
else
     request.getRequestDispatcher("loginInvalido.html").
                                    forward(request, response);
```

Exercícios

1) O que acontece quando você está na página CadPessoa e clica no botão Enviar?

2) Por que isso ocorre?

Página index2.html

```
<form action="Controlador2?acao=Login" method="post">
    <label ><strong>E-mail:</strong>
    </label> <input type="text" name="email" value="" />
    <br />
    <label><strong>Senha:</strong>
    </label> <input type="password" name="senha"  value="" />
    <input type="hidden" name="logica" value="LoginLogout" />
    <br />
    <input  name="entrar" type="submit"  value="Entrar" />
</form>
```

Servlet Controlador2

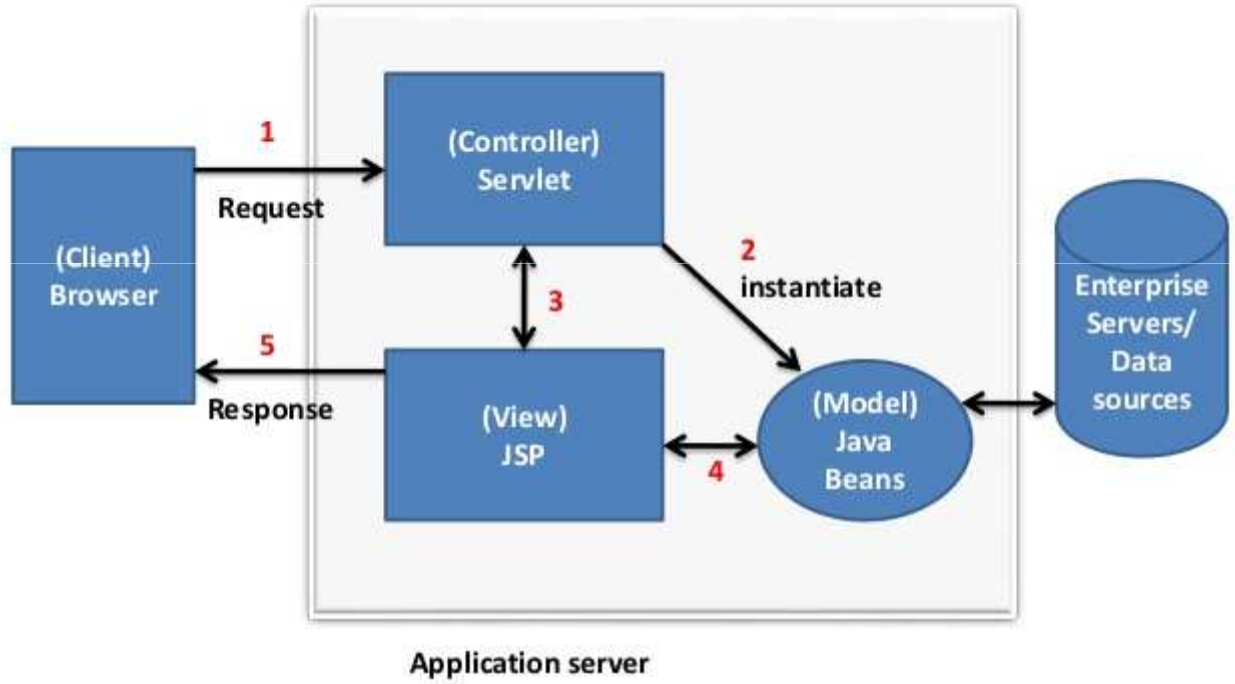
```
String acao = request.getParameter("acao");
if(acao.equals("Login")){
   String email = request.getParameter("email");
   String senha = request.getParameter("senha");
   if(email!=null && senha!=null){
      request.getRequestDispatcher("cadPessoa2.html").
            forward(request, response);
   }
   else
       request.getRequestDispatcher("loginInvalido.html").
                                        forward(request, response);
  }
```

Servlet Controlador2

```
String acao = request.getParameter("acao");
...
if(acao.equals("cadPessoa")){
   String nome = request.getParameter("nome");
   String endereco = request.getParameter("endereco");
   if(nome!=null && endereco!=null){
     request.getRequestDispatcher("cadastroSucesso.html").
                                        forward(request, response);
}
else
     request.getRequestDispatcher("cadastroErro.html").
                                      forward(request, response);
}
```

Quais seriam os próximos passos?

MVC com Servlets



Passos com servlets?

Passo 1 - Ler os dados dos formulários

Passo 2 - Montar os objetos

Passo 3 - Chamar os objetos DAO correspondentes

Passo 4 - Interligar todas as páginas entre si

Servlet Controlador3

```
String acao = request.getParameter("acao");
if(acao.equals("Login")){
   String email = request.getParameter("email");
   String senha = request.getParameter("senha");
   Usuario user = new Usuario(email, senha);
   if(user.autentica()){
      request.getRequestDispatcher("cadPessoa2.html").
                                    forward(request, response);
   }
   else
      request.getRequestDispatcher("loginInvalido.html").
                                    forward(request, response);
            }
```

Classe Usuário

```
public class Usuario{

    //...

    public boolean autentica(){

        UsuarioDAO user = new UsuarioDAO();

        return user.autentica(this);

    }
```

Classe UsuarioDAO

```
public class UsuarioDAO implements GenericDAO<Usuario>{
  //...
  public boolean autentica(Usuario user){
     boolean retorno = false;
     try(Connection connection = new
                 ConnectionFactory().getConnection();
         PreparedStatement stmt =
                    connection.prepareStatement(
                          SQLs.AUTENTICA_USUARIO.getSql())){
         ResultSet rs = stmt.executeQuery();
         rs.last();
         if(rs.getRow()>=1) return true;
     }catch(SQLException e){
   ...
```

Problemas com esse tipo de solução?

Problemas

Os dados dos objetos são manipulados pelo servlet para depois serem associados aos objetos

O servlet fica extenso e manipula várias informações simultaneamente

O código HTML fica misturado às instruções Java

Solução: JSP

JSP

O QUE É JSP?

Tecnologia Java usada para o desenvolvimento de aplicações Web

Utiliza o conceito de geração de páginas dinâmicas

Permite escrever dentro do HTML comandos da linguagem de programação Java usando: expressões, diretivas, scriptlets, etc

Exemplo 1

```
<html>
  <head>
     <title>Exemplo 1</title>
  </head>
  <body>
     <%
       out.println(“Mensagem!“);
     %>
  </body>
</html>
```

Exemplo 2

```
<html>
  <head>
     <title>Exemplo 2</title>
  </head>
  <body>
    <%   out.println("Parâmetro nome:"+
           request.getParameter("nome")); %>
  </body>
</html>
```

Exemplo 3

```
<html>
  <head> <title>Exemplo 3</title></head>
  <body>
...
      <%
         String msg = request.getParameter("msg");
       %>
      <b>Mensagem por parâmetro: <%=msg%> </b>
  </body>
</html>
```

Estrutura da Página

Uma página JSP pode ser dividida em:

Cabeçalho: inclui diretivas que informam alguns atributos da página bem como algumas configurações (imports, contentType, página de erro, entro outras)

Declarações: inclui declarações de métodos, atributos, constantes, etc.

Corpo da página: inclui o código que irá gerar o HTML dinâmico

Estrutura da Página

O corpo da página pode conter:

HTML estático

Código JAVA puro, ou scriptlets: **<%...%>**

Diretivas: **<%@.....%>**

Expressões: **<%=.......%>**)

Ações: <jsp:include../>

Qualquer outro tipo de conteúdo estático: (JavaScript, SVG, XML, etc...).

Scriptlets

Bloco de código JAVA que será executado durante a requisição

```
<% int anoAtual = 2017;
    int anoNascimento = 2000;
    int idade = anoAtual-anoNascimento;
    out.println(idade);
%>
```

Scriptlets

```
<% for (int i=0; i<10; i++)

        out.println("<b> i= "+i+"</b><br />");

%>
```

Declarações

Bloco de código JAVA que define variáveis, constantes e métodos referentes a página. É definido entre os símbolos **<%!** e **%>**

```
<% imprimir(response.getWriter(), 20);%>
```

```
<%!
   private void imprimir(PrintWriter out, int num){
      for (int i=0; i<num; i++)
           out.println("<b> i= "+i+"</b><br />");
   }
%>
```

Expressões

```
<% int anoAtual = 2014;
   int anoNascimento = 2000;
   int idade = anoAtual-anoNascimento;
%>
<html>
  <head> </head>
  <body>
  <h1>Calculando a idade</h1>
        <h2>   <%=idade%>   </h2>
    </body>
</html>
```

Diretiva page

```
<%@page contentType="text/html"
                              pageEncoding="UTF-8"%>
```

Define que o conteúdo será html e no formato UTF-8

Vamos analisar a página Exemplo6.jsp?

Diretiva page

```
<%@page import="java.io.PrintWriter"%>
```

Usada para importar classes Java ou desenvolvidas pelo programador

Diretiva página de Erro

```
<%@page errorPage="PaginaErro.jsp" %>
```

Direciona a página atual para a página de erro caso alguma exceção ocorra

errorPage: especifica uma URL para uma página de tratamento de erros

Agora ver Exemplo4 e PaginaErro.jsp

Diretiva página de Erro

```
<%@page errorPage="PaginaErro.jsp" %>
```

isErrorPage: informa se a página é de tratamento de erro ou não

isErrorPage – define qual é a página de erro

```
<%@page isErrorPage="true" %>
<!DOCTYPE html>
<html>
  <head> <title>JSP Page</title>  </head>
  <body>
     <h2>Descri&ccedil;&atilde;o do erro: <%=exception%></h2>
     <br/>
     <% exception.printStackTrace(new java.io.PrintWriter(out));%>
    </body>
</html>
```

Manipulando Formulários

Usando JSP em formulários

```
<form method="POST" action="Exemplo9Leitura.jsp">
            Usuário
            <input type="text" name="usuario"/>
            <br />
            Email:
            <input type="text" name="email"/>
            <br />
            <input type="submit" value="OK"/>
            <input type="reset" value="Limpar"/>
</form>
```

Usando JSP em formulários

```
<%@page contentType="text/html"pageEncoding="UTF-8"%>
<%@page errorPage="PaginaErro.jsp" %>
<%
    String nome = request.getParameter("usuario");
    String email = request.getParameter("email");
%>
<html>
    <head> <title>JSP Page</title> </head>
    <body>
        <h1> Você informou:</h1>
        <h2> Nome: <%= nome %> </h2>
        <h2> Email: <%= email %> </h2>
    </body>
</html>
```

Manipulando Coleções

Manipulando Coleções

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="classes.Colecao"%>
<%@page import="java.util.*" %>
<html>
    <head><title>JSP Page</title> </head>
    <body>
        <h1>Dados da coleção:</h1>
        <% List<String> lista = Colecao.carregaColecao();
            for (String valor : lista) {
        %>
        <h2><%= valor%> </h2>
        <%}%>
 </body>
</html>
```

Classe Colecao

```
public class Colecao {

    public static List<String> carregaColecao() {
        List<String> minhaLista = new ArrayList<>();
        for (int i = 0; i < 10; i++) {
            minhaLista.add("valor " + i);
        }
        return minhaLista;
    }
}
```