

Java Enterprise Edition - JEE

08. MVC - Model View Controller

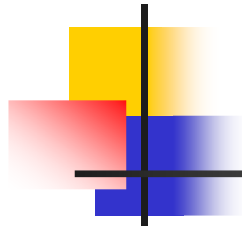


Esp. Márcio Palheta
gtalk: marcio.palheta@gmail.com



Agenda

- Servlet ou JSP?
- Arquitetura MVC;
- A camada de controle;
- Request dispatchers;
- Exercício 01 – Implementação MVC



Servlet ou JSP?

- Colocar todo HTML dentro de uma Servlet realmente não parece a melhor idéia.
- O que acontece quando precisamos mudar o design da página?
- Também não parece agradável utilizar código JAVA em páginas HTML;
- Vamos utilizar o melhor dos dois mundos;



JSP ou Servlet?

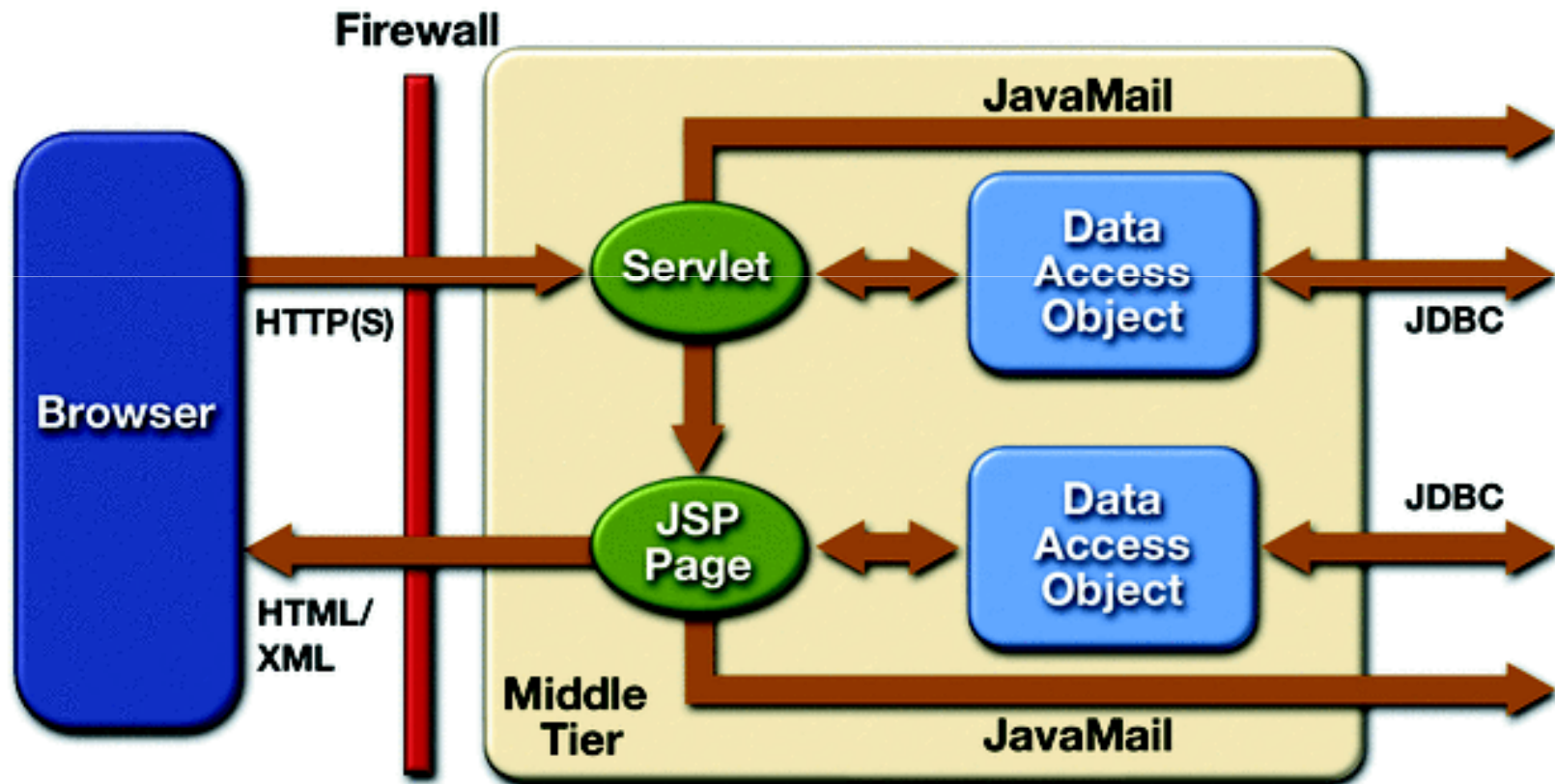
- A JSP foi criada apenas para apresentar o resultado, não devendo realizar acessos a bancos de dados.
- O acesso a banco e redirecionamento das telas deve ficar a cargo da Servlet.
- A Servlet deve realizar o controle, e a JSP deve ser preocupar apenas em exibir o resultado.
- A Servlet possui a lógica de negócios e a JSP tem a lógica de apresentação.



Arquitetura MVC

- Model – camada responsável pela persistência dos dados da aplicação;
- View – camada responsável pela exibição das telas e interação com o usuário;
- Controller – camada responsável por controlar o fluxo de páginas da aplicação;

Arquitetura MVC





A camada de controle

- Tem como principal objetivo o **controle do fluxo navegacional** da aplicação;
- Serve de **interface** entre a camada **View** e a camada **Model**;
- Recebe requisições de uma tela(página JSP ou HTML);
- Realiza um determinado procedimento, como chamada de consultas;
- Redireciona para a próxima tela;



Request dispatchers

- Para separarmos a lógica de negócio da lógica de apresentação, precisamos criar um mecanismo capaz de:
 - Receber uma requisição;
 - Executar um processamento;
 - E redirecionar para o próximo **recurso web**;



Request dispatchers

- A API da servlet nos permite fazer tal **redirecionamento**.
- É necessário que conheçamos a URL que queremos acessar;
- Utilizamos um objeto **RequestDispatcher** para acessar outro recurso web;
- Um recurso web pode ser uma página jsp ou uma servlet

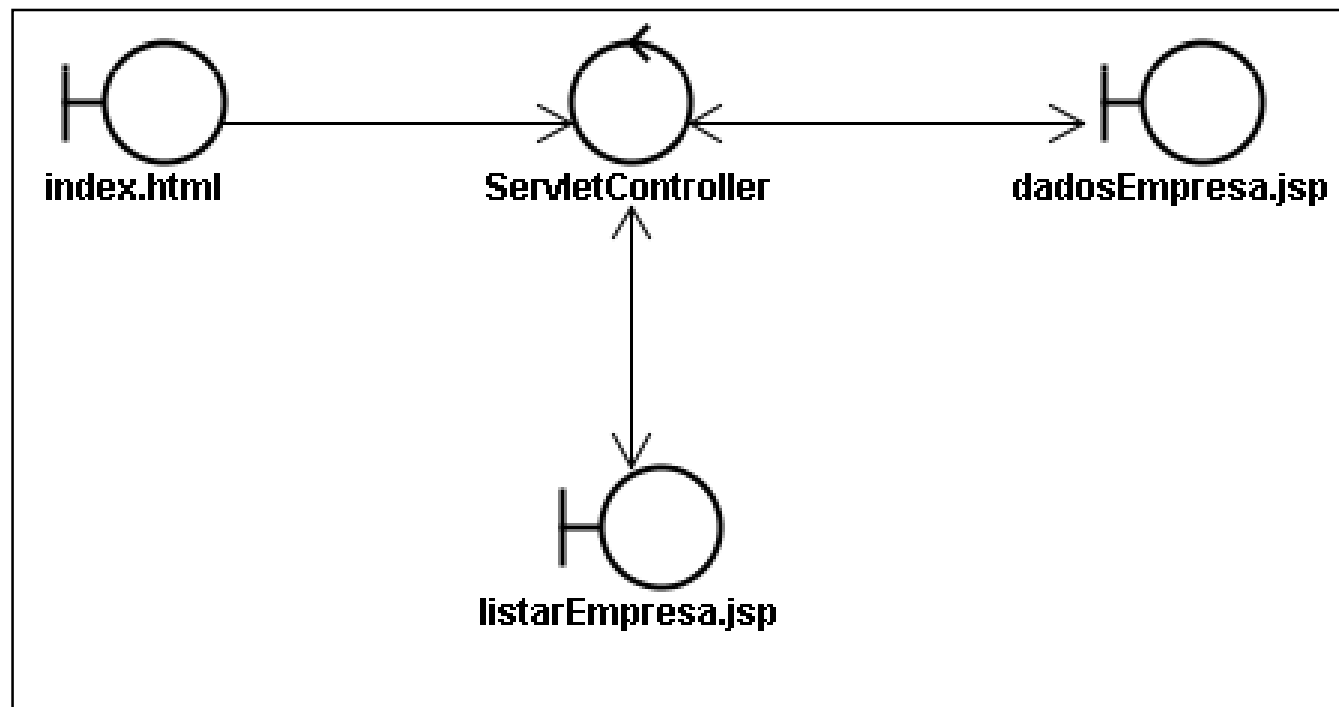


Código Servlet para redireccionamiento de páginas

- //Prepara o redireccionamento
- `RequestDispatcher rd =
request.getRequestDispatcher("/proxima.jsp");`
- //Redirecciona para a url `"/proxima.jsp"`
- `rd.forward(request,response);`

Exercício 01

- Implementar a o diagrama navegacional a seguir:





Exercícios - MVC

- MVC com JSPs, Servlets e DAOs;
- Tarefas das páginas:
 - Exibir dados e enviar requisições para Servlet;
- Tarefas da Servlet:
 - Receber requisição, invocar serviços DAO e;
 - Encaminhar para a próxima página;
- Tarefa das classes DAO:
 - Persistir os dados solicitados

Página index.html

```
index.html x
1 <html>
2 <head><title>Sistema empresa</title></head>
3 <body>
4     <h1>Sistema de empresas</h1>
5     1 <a href="controller?metodo=listar">Listar</a>
6       <br><br>
7     2 <a href="controller?metodo=cadastrar">Cadastrar</a>
8 </body>
9 </html>
```



listagemEmpresa.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head><title>Listagem de empresas</title></head>
<body>
<table border="2">
    <tr><th>CNPJ</th><th colspan="3">RAZÃO SOCIAL</th></tr>
    <c:forEach var="empresa" items="${listaDeEmpresas}">
        <tr>
            <td>${empresa.cnpj}</td>
            <td>${empresa.razaoSocial}</td>
            <td><a href="controller?metodo=alterar&id=${empresa.id}">Alterar</a></td>
            <td><a href="controller?metodo=excluir&id=${empresa.id}">Excluir</a></td>
        </tr>
    </c:forEach>
</table><br>
<button onclick="history.go(-1)">Voltar</button>&nbsp;&nbsp;&nbsp;
<button onclick="location.href='controller?metodo=cadastrar'">Cadastrar</button>
</body>
</html>
```

dadosEmpresa.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<html>
<head><title>Dados da empresa</title></head>
<body>
  <h1>
    <c:choose>
      <c:when test="${empresa.id==null}">Cadastrar empresas</c:when>
      <c:otherwise>Alterar dados da empresa</c:otherwise>
    </c:choose>
  </h1>
  <form action="controller?metodo=salvar" method="post">
  <table>
    <c:if test="${empresa.id!=null}">
      <tr>
        <td>ID: </td>
        <td><input type="text" name="edId" value="${empresa.id}" readonly/></td>
      </tr>
    </c:if>
    <tr>
      <td>CNPJ: </td>
      <td><input type="text" name="edCNPJ" value="${empresa.cnpj}" /></td>
    </tr>
    <!-- Continua... -->
```

dadosEmpresa.jsp – cont...



```
<!-- Continua... -->
<tr>
  <td>RAZÃO SOCIAL: </td>
  <td><input type="text" name="edRazaoSocial"
              value="${empresa.razaoSocial}"/></td>
</tr>
<tr>
  <td>ENDEREÇO: </td>
  <td><input type="text" name="edEndereco"
              value="${empresa.endereco}"/></td>
</tr>
<tr>
  <td>TELEFONE: </td>
  <td><input type="text" name="edTelefone"
              value="${empresa.telefone}"/></td>
</tr>
<tr>
  <td>SITE: </td>
  <td><input type="text" name="edSite" value="${empresa.site}"/></td>
</tr>
<tr>
  <td>EMAIL: </td>
  <td><input type="text" name="edEmail" value="${empresa.email}"/></td>
</tr>
<!-- Continua... -->
```


dadosEmpresa.jsp – final

```
<!-- Continua... -->
<tr>
  <td>Data de Criação: </td>
  <td><input type="text" name="edData"
    value="<fmt:formatDate value="${empresa.dataCriacao.time}"
    pattern="dd/MM/yyyy" />
  </td>
</tr>
<tr>
  <td><input type="submit" value="Enviar dados"/></td>
  <td><input type="button" value="Cancelar" onclick="history.go(-1)"/></td>
</tr>
</table>
</form>
</body>
</html>
```



Página mensagem.html

```
<html>
<head>
<title>Mensagem</title>
</head>
<body>
  <h1>Operação realizada com sucesso</h1>
  <a href="index.html">Página inicial</a><br>
  <a href="controller?metodo=Listar">Listagem</a>
</body>
</html>
```

ServletController.java

Parte inicial

```
package br.fucapi.curso.jee.control; 1

import java.io.IOException;


//Lista de operacoes possiveis, usando ENUM
enum Metodo{ALTERAR, CADASTRAR, EXCLUIR, SALVAR, LISTAR}; 2

@WebServlet("/controller")
public class ServletController extends HttpServlet { 3
    private static final long serialVersionUID = 1L;

    //Objeto DAO utilizado para invocar servicos da camada de modelo
    private EmpresaDAO dao = new EmpresaDAO();


    //continua...
```

ServletController – cont...



```
protected void service(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    String pagina = "";
    //Metodo invocado pela camada de visao
    String metodoSTR = request.getParameter("metodo");
    Metodo metodoEnum = Metodo.valueOf(metodoSTR.toUpperCase());
    //Seleciona o procedimento, de acordo com a requisicao da view
    switch (metodoEnum) {
        case CADAstrar:
            pagina = cadastrar(request); break;
        case ALTERAR:
            pagina = alterar(request); break;
        case EXcluir:
            pagina = excluir(request); break;
        case SALVAR:
            pagina = salvar(request); break;
        case LISTAR:
            pagina = listar(request); break;
    }
    //Definicao do objeto de redirecionamento
    RequestDispatcher dispatcher =
        request.getRequestDispatcher("/") + pagina);
    //Redirecionando para a proxima pagina
    dispatcher.forward(request, response);
}
```

ServletController – cont...



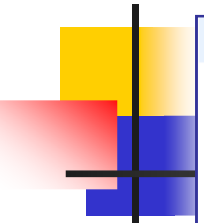
```
private String cadastrar(HttpServletRequest request) {
    request.setAttribute("empresa", new Empresa());
    return "dadosEmpresa.jsp";
}

private String alterar(HttpServletRequest request) {
    String id = request.getParameter("id");
    Empresa empresa = dao.consultar(new Long(id));
    request.setAttribute("empresa", empresa);
    return "dadosEmpresa.jsp";
}

private String excluir(HttpServletRequest request) {
    String id = request.getParameter("id");
    dao.excluir(new Long(id));
    return "mensagem.html";
}

private String listar(HttpServletRequest request) {
    List<Empresa> lista = dao.listar();
    request.setAttribute("listaDeEmpresas", lista);
    return "listagemEmpresa.jsp";
}
```

ServletController – final...



```
private String salvar(HttpServletRequest request) {
    Empresa empresa = new Empresa();
    String id = request.getParameter("edId");
    empresa.setCnpj(request.getParameter("edCNPJ"));
    empresa.setRazaoSocial(request.getParameter("edRazaoSocial"));
    empresa.setEndereco(request.getParameter("edEndereco"));
    empresa.setTelefone(request.getParameter("edTelefone"));
    empresa.setSite(request.getParameter("edSite"));
    empresa.setEmail(request.getParameter("edEmail"));
    String dataEmTexto = request.getParameter("edData");
    try {
        Date date =
            new SimpleDateFormat("dd/MM/yyyy").parse(dataEmTexto);
        empresa.setDataCriacao(Calendar.getInstance());
        empresa.getDataCriacao().setTime(date);
    } catch (ParseException e) { e.printStackTrace(); }
    if(id == null || id.equals("")){
        dao.cadastrar(empresa);
    }else{
        empresa.setId(new Long(id));
        dao.alterar(empresa);
    }
    return "mensagem.html";
}
```



Atividades do projeto final:

- Definição de padrão de telas;
- Implementação de protótipo;
- Teste de navegabilidade;
- Validação do protótipo;



O que vem a seguir?

- Revisão de MVC;
- Camada de controle com struts;
- Persistência com Hibernate;
- Arquitetura de software com:
 - View: JSP, Servlets e Filtros;
 - Controller: Struts;
 - Model: Dao e Hibernate;



Referências

- Hall, Marty, “Core Servlets and Java Server Pages”, Janeiro 2002, Sun Microsystems Press;
- <http://java.sun.com/>
- <http://java.sun.com/j2ee/1.6/docs/tutorial/doc/index.html>
- <http://java.sun.com/products/jndi/docs.html>
- <http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>

Java Enterprise Edition - JEE

08. MVC - Mode View Controller



Esp. Márcio Palheta
gtalk: marcio.palheta@gmail.com