

Spring MVC - Continuação

- **Exercício 1.** No projeto "aulaspring", implemente uma listagem dos alunos que deve ser apresentada atualizada após as operações de inserção, alteração e remoção. Cada linha da tabela deve ter um link para "Alterar" e um link para "Remover". Essa página deve conter também um link para a página de inserção de novos registros que já fizemos na última aula.

INSTRUÇÕES:

- **Primeiro Passo:** para implementar a listagem dos alunos, você deve modificar a classe `AlunoController.java`, acrescentando o seguinte método:

```
@RequestMapping("listaAlunos")
public String lista(Model model) {
    AlunoDAO dao = new AlunoDAO();
    model.addAttribute("alunos", dao.getLista());
    return "aluno/lista";
}
```

Agora, crie o arquivo `lista.jsp` com o seguinte conteúdo:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="f"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Listagem dos Alunos</title>
</head>
<body>
    <table border=1>
        <tr>
            <td><b>Id</b></td>
            <td><b>Nome</b></td>
            <td><b>E-mail</b></td>
            <td><b>Endereço</b></td>
        </tr>
        <c:forEach items="${alunos}" var="aluno">
            <tr>
                <td>${aluno.id}</td>
                <td>${aluno.nome}</td>
                <td>${aluno.email}</td>
                <td>${aluno.endereco}</td>
            </tr>
        </c:forEach>
    </table>
</body>
</html>
```

- **Segundo Passo:** para redirecionar uma requisição de volta para a lista de alunos, você deve modificar a classe `AlunoController.java`, modificando os métodos de cada operação como neste exemplo:

```
@RequestMapping("insere")
public String insere(Aluno aluno) {
    AlunoDAO dao = new AlunoDAO();
    dao.insere(aluno);
}
```

```

        return "redirect:listaAlunos";
    }

```

- **Terceiro Passo:** para criar links de "Remover" na listagem, além de implementar a operação na classe `AlunoController.java`, você precisa do seguinte código:

```

<td><a href="remove?id=${aluno.id}">Remover</a></td>

```

- **Quarto Passo:** para criar links de "Alterar" na listagem, além de implementar a operação na classe `AlunoController.java`, você precisa do seguinte código:

```

<td><a href="exibeAluno?id=${aluno.id}">Alterar</a></td>

```

Crie também uma ação para a URL `exibeAluno` na classe `AlunoController.java`. Essa ação deve obter os dados do registro de acordo com o id fornecido e colocar os dados em campos de uma página jsp para que o usuário possa modificar:

```

@RequestMapping("exibeAluno")
public String exibe(Long id, Model model) {
    AlunoDAO dao = new AlunoDAO();
    model.addAttribute("aluno", dao.buscaPorId(id));
    return "aluno/exibe";
}

```

A página `exibe.jsp` contém:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<html>
  <head>
    <title>Alteração de Aluno</title>
  </head>
  <body>
    <form action="alteraAluno" method="POST">
      <input type="hidden" id="id" name="id" value="${aluno.id }"/><br />
      Nome: <input type="text" id="nome" name="nome" value="${aluno.nome }"/><br />
      Email: <input type="text" id="email" name="email" value="${aluno.email }"/><br />
      Endereço: <input type="text" id="endereco" name="endereco" value="${aluno.endereco }" /><br />
      <input type="submit" value="Alterar"/>
    </form>
  </body>
</html>

```