

## Enviando dados para a JSP

Nosso próximo passo agora é listar as contas que foram criadas. O processo é sempre o mesmo, precisamos escrever uma ação em um Controller e depois uma View para mostrar esses dados.

Vamos começar pelo Controller. Precisamos pedir a lista de Contas para o DAO, e mandá-la pra View "conta/lista":

```
@RequestMapping("/listaContas")
public String lista() {
    ContaDAO dao = new ContaDAO();
    List<Conta> contas = dao.lista();
    return "conta/lista";
}
```

Mas a pergunta é: como mandar a variável "contas" (que tem a lista de Contas), para a JSP? Para isso, precisamos usar a classe `ModelAndView` do Spring. Essa classe é como se fosse uma ponte entre o controller e a view; podemos inserir "elementos" nela, que estarão disponíveis na JSP. Basta criarmos essa classe, e depois devolvermos ela no método (ao invés de String). Como não devolveremos mais a String com o caminho do JSP, o passaremos também para esse `ModelAndView` :

```
@RequestMapping("listaContas")
public ModelAndView lista() {
    ContaDAO dao = new ContaDAO();
    List<Conta> contas = dao.lista();

    ModelAndView mv = new ModelAndView("conta/lista");
    mv.addObject("contas", contas);
    return mv;
}
```

Veja o método `addObject()` : ele recebe uma String e um Objeto. A String é o nome/apelido que o Objeto receberá na View. Nesse caso, a variável `contas` estará disponível com o nome "contas" (o nome bateu, mas não precisa ser necessariamente assim).

Na JSP, podemos capturá-la usando EL ou JSTL. Por exemplo, `${contas}` .

Precisamos agora escrever a JSP. Vamos lá:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<body>
    <table style="height: 10px; width: 775px; border="1">
        <tr>
            <th>Código</th>
            <th>Descrição</th>
            <th>Valor</th>
            <th>Tipo</th>
        </tr>
    </table>

```

```
<th>Paga?</th>
<th>Data de pagamento</th>
</tr>

<c:forEach items="${contas}" var="conta">
<tr>
  <td>${conta.id}</td>
  <td>${conta.descricao}</td>
  <td>${conta.valor}</td>
  <td>${conta.tipo}</td>
  <td id="conta_${conta.id}">
    <c:if test="${conta.paga eq false}">
      Não paga
    </c:if>
    <c:if test="${conta.paga eq true }">
      Paga!
    </c:if>
  </td>
  <td><fmt:formatDate value="${conta.dataPagamento.time}" pattern="dd/MM/yyyy"/></td>
</tr>
</c:forEach>
</table>

</body>
</html>
```

Repare que fizemos uso do `foreach` da JSTL para iterar sobre os elementos da Conta.