



**SOU Java**

sociedade de usuários java

**Apache Struts 2**  
**Turbinando o**  
**desenvolvimento Web**

**Claudio Miranda**



*fisl8.0*

**Struts<sup>2</sup>**

# Agenda

- Funcionalidades
- Binding
- Apresentação
- Interceptadores
- Validação



# Agenda

- Funcionalidades
- Binding
- Apresentação
- Interceptadores
- Validação



# Funcionalidades

- Baseado em POJOs (Plain Old Java Object)
- Independência de Servlets
- Anotações ou XML
- Suporte a temas
- Validações detalhadas e customizáveis
- EL avançado
  - Baseado em OGNL

# Funcionalidades

- Suporte a diferentes tecnologias de apresentação
  - Velocity, freemarker, JSP
- Modo de desenvolvimento
- Suporte nativo para AJAX
  - DOJO e DWR
- Integração nativa com Spring
  - Spring beans podem ser actions

# Funcionalidades

- Encadeamento de Actions
- Actions configuradas por pacotes
- Arquitetura voltada em plugins
  - Tiles, JasperReports, JFreeChart, JSF, Sitemesh,
- Baseado no projeto WebWork 2

# Funcionalidades

Struts 1	Struts 2
Action	Action
ActionForm	Action or POJO
ActionForward	Result
struts-config.xml	struts.xml
ActionServlet	FilterDispatcher
RequestProcessor	Interceptors
validation.xml	<i>Action-validation.xml</i>

# Agenda

- Funcionalidades
- **Binding**
- Apresentação
- Interceptadores
- Validação

# Struts<sup>2</sup>

# Binding

Binding é baseado na tecnologia “ValueStack”, onde os dados do request contrói o POJO e é injetado na Action

- Basta declara o seu POJO no Action e usá-lo no JSP/Velocity/Freemarker

# Binding

## Formulário (Freemarker)

```
<@s.form action="login_save" >
  <@s.textfield label="Nome de Usuário"
name="login.name" validate="true"/>
  <@s.password label="Senha"
name="login.password" validate="true"/>
  <@s.textfield label="e-mail"
name="email" validate="true"/>
  <@s.submit value="Salvar"/>
</@s.form>
```

# Binding

Implementação da Action

```
public class LoginAction extends  
ActionSupport {
```

```
private Login login;  
private String email;
```

# Binding

- Pode ser usado OGNL em expressões mais complexas

```
<s:form action="submitAddressesInfo"
namespace="/conversion">
  <s:iterator value="%{new int[3]}"
status="stat">
  <s:textfield label="%{'Address '#stat.index}"
name="%{'addresses('\ 'id'+#stat.index+' \ ' ).adre
ss'}" />
  </s:iterator>
  <s:submit />
</s:form>
```

# Agenda

- Funcionalidades
- Binding
- **Apresentação**
- Interceptadores
- Validação

# Struts<sup>2</sup>

# Apresentação

- Podem ser usadas 3 tecnologias para apresentação
  - Freemarker (padrão)
  - JSP 2
  - Velocity
- Configurável no arquivo struts.xml
- OGNL pode ser usado em todas
- Suporta objetos no contexto

# Apresentação

- Freemarker
  - Exibe os erros de maneira melhorada
  - Sintaxe similar a do Velocity
  - Suporte a JSP Taglibs

# Apresentação

- Freemarker
  - Suporte a JSP Taglibs

```
<servlet-name>JspSupportServlet</servlet-name>  
<servlet-class>  
org.apache.struts2.views.JspSupportServlet  
</servlet-class>  
<load-on-startup>1</load-on-startup>
```

```
<#assign mytag=JspTaglibs ["/WEB-  
INF/mytag.tld"]>  
<@mytag.tagx attribute1="some ${value}"/>
```

# Agenda

- Funcionalidades
- Binding
- Apresentação
- **Interceptadores**
- Validação

# Struts<sup>2</sup>

# Interceptadores

- Permitem interceptar a invocação antes e após a Action ser invocada
- Interferir com o ciclo de vida da Action, de maneira não intrusiva
- Muitos interceptadores prontos e úteis

# Interceptadores

- Debugging
- Execute and Wait
- File Upload
- Logger
- ModelDriven
- Non-IoC version of OpenSessionInView
- Profiling
- Roles
- Scope
- Timer
- Token

# Agenda

- Funcionalidades
- Binding
- Apresentação
- Interceptadores
- Validação

# Struts<sup>2</sup>

# Validação

- Javascript
- AJAX
- Server side
- Mensagens de erros localizadas
- Anotações
- Validações customizáveis
- Uso de expressões regulares

# Validação

- Anotações

## @Validation()

```
public interface AnnotationDataAware {  
    void setBarObj(Bar b);  
    Bar getBarObj();  
    @RequiredFieldValidator(message = "You must  
enter a value for data.")  
    @RequiredStringValidator(message = "You must  
enter a value for data.")  
    void setData(String data);  
    String getData();  
}
```

# Validação

- XML

```
<validators>
  <field name="login.name">
    <field-validator type="requiredstring">
      <message><![CDATA[É necessario informar um nome de
usuario de 4 a 20 caracteres]]></message>
    </field-validator>
    <field-validator type="stringlength">
      <param name="minLength">4</param>
      <param name="maxLength">20</param>
      <message><![CDATA[É necessario informar um nome de
usuario de 4 a 20 caracteres]]></message>
    </field-validator>
  </field>
```

# Validação

- Expressão regular

```
<field name="myStrangePostcode">  
  <field-validator type="regex">  
    <param name="expression">  
<![CDATA([aAbBcCdD][123][eEfFgG][456])]>  
    </param>  
  </field-validator>  
</field>
```

# Agenda

- Funcionalidades
- Binding
- Apresentação
- Interceptadores
- Validação



# Struts 2

- Modo de desenvolvimento
  - `devMode=true; profiling=yes; debug=console`
- Facilmente testável
  - Desacoplado de servlets
- Aplicações Struts1 podem funcionar
- Migração não é complexa, mas não é trivial
- Grande quantidade de tags de apresentação
- Produtividade no desenvolvimento

# Struts 2

- Comunidade vibrante
- Extensa documentação
- Licença Apache
- Inovação tecnológica

# Struts 2

Tutoriais, Guias, Dicas, Artigos

<http://struts.apache.org>

<http://cwiki.apache.org/S2WIKI/home.html>

[claudio@claudius.com.br](mailto:claudio@claudius.com.br)