

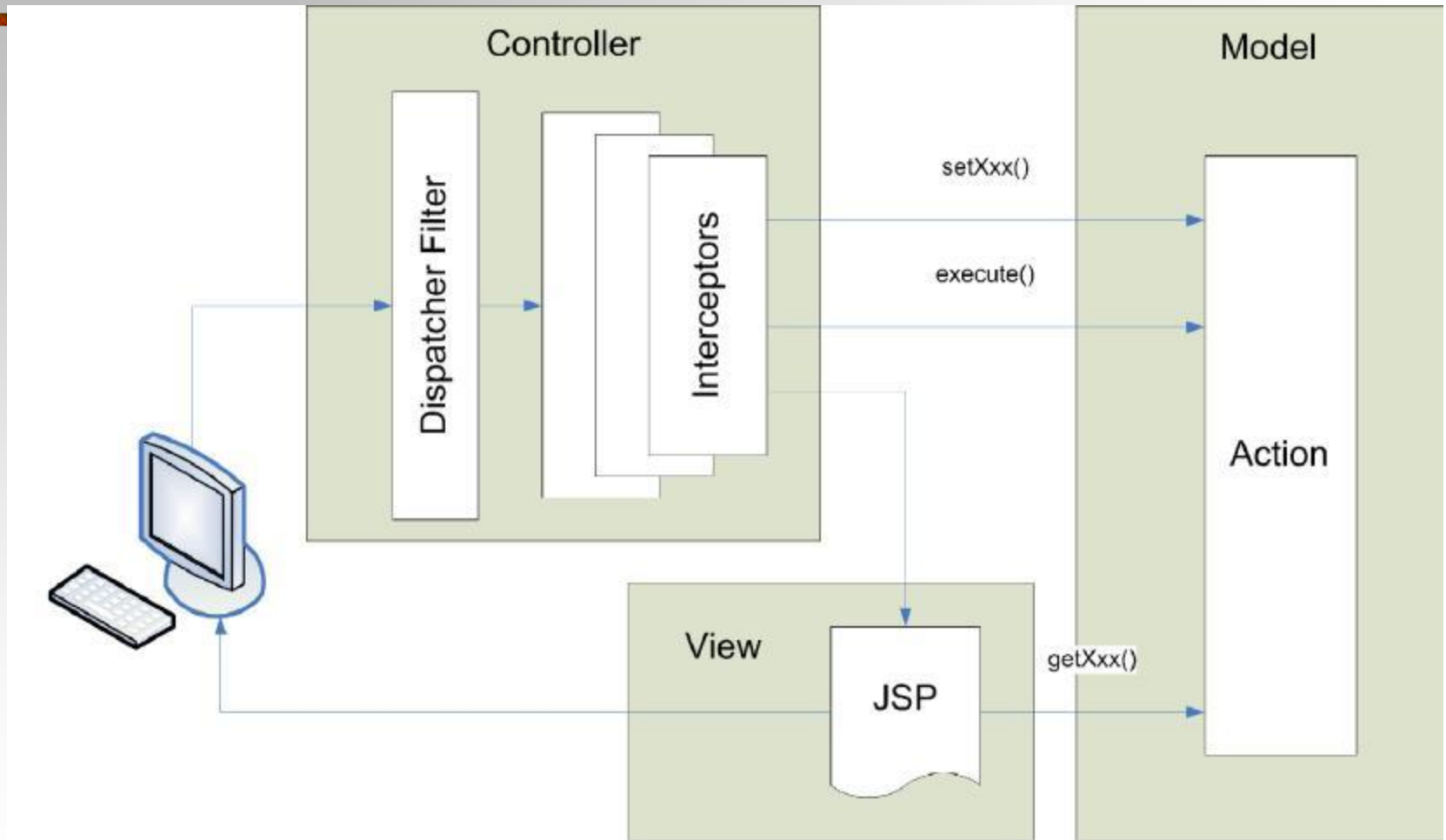
Struts 2

Veerle Ongenae

Struts

- Implementatie MVC/Model 2
 - Eigenlijk MVC2 of pull-MVC
 - Acties maken deel uit van het model
 - Struts 2: samengaan
 - Struts
 - WebWork
-

Principe Struts 2



Principe Struts 2

■ Controller

- Filter
- Interceptors
 - Instellen eigenschappen actieklassen
 - Oproepen actie

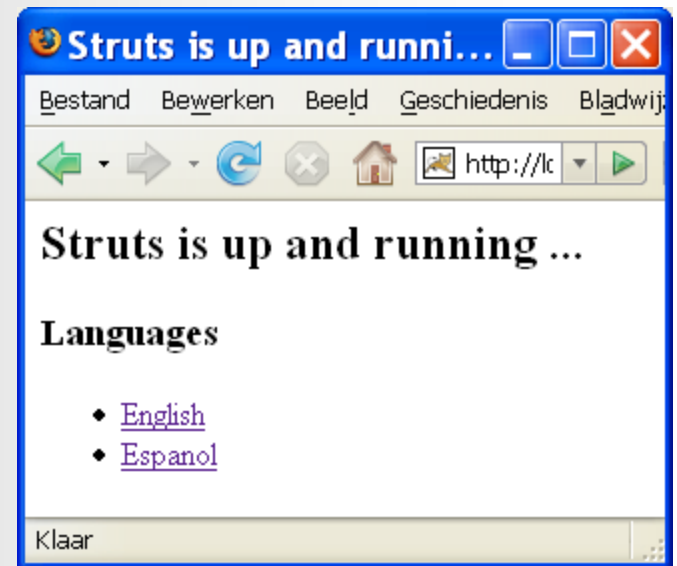
■ Model

- Actieklassen
 - Uitvoeren actie
 - Data: instellen en opvragen

■ View

- Tonen model
 - Bv. JSP
-

Voorbeeld



- <http://localhost:8084/struts/HelloWorld.action>
- Actie
 - example.HelloWorld.java
- View
 - HelloWorld.jsp

Controller

■ Filter

- Verschil met servlets
 - Genereren GEEN antwoord
- Request bekijken en manipuleren
 - Headers, parameters, attributen
- Response: headers en body aanpassen
- Ketting van filters

■ Configuratie in web.xml

Configuratie Controller

<filter>

 <filter-name>struts2</filter-name>

 <filter-class>

 org.apache.struts2.dispatcher.FilterDispatcher

 </filter-class>

</filter>

<filter-mapping>

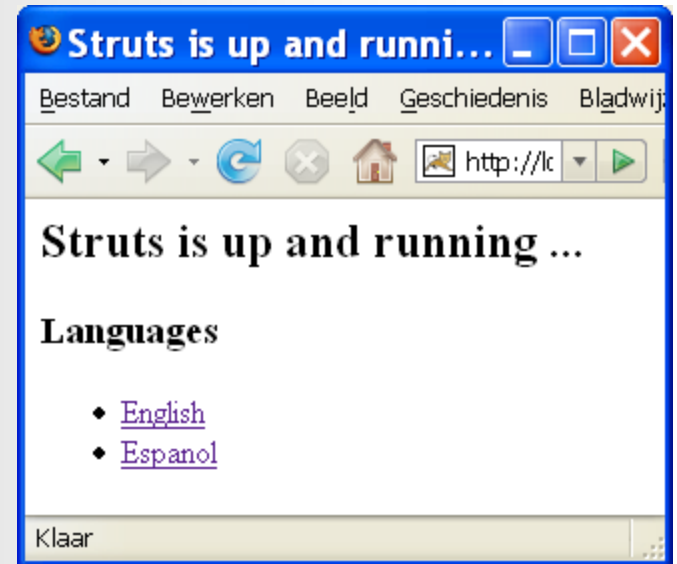
 <filter-name>struts2</filter-name>

 <url-pattern>/*</url-pattern>

</filter-mapping>

Actie

- HelloWorld.java
- Java-klasse
 - Afgeleid van
com.opensymphony.xwork2.ActionSupport
 - Methode execute
 - Eigenlijke actie
 - Resultaat: string → type resultaat actie
 - Bv. SUCCESS
 - Eigenschappen
 - Data



Actie: ingebouwde methodes, ...

■ getText()

- Opvragen informatie uit properties-bestand (zelfde map als klasse)
 - package.properties
 - package_es.properties

■ SUCCESS

- Resultaat uitvoeren actie
- Beschikbare resultaatcodes (via ActionSupport)

String SUCCESS = "success";

String NONE = "none";

String ERROR = "error";

String INPUT = "input";

String LOGIN = "login";

Samenvoegen

- URL verbinden met actie
 - Resultaat actie koppelen aan view
 - struts.xml
 - In WEB-INF/classes
 - Opnemen in basis van Java-map
-

struts.xml

■ Configuratie voor Struts

- Overschrijven standaardconfiguratie
- Beschrijving acties
- Kan opgesplitst worden in verschillende xml-bestanden

<struts>

<constant name="struts.enable.DynamicMethodInvocation"
value="false" />

<constant name="struts.devMode" value="false" />

<include file="example.xml"/>

</struts>

struts.xml

■ example.xml

```
<struts>
```

```
  <package name="example" extends="struts-default">
```

```
    <action name="HelloWorld" class="example.HelloWorld">
```

```
      <result>/HelloWorld.jsp</result>
```

```
    </action>
```

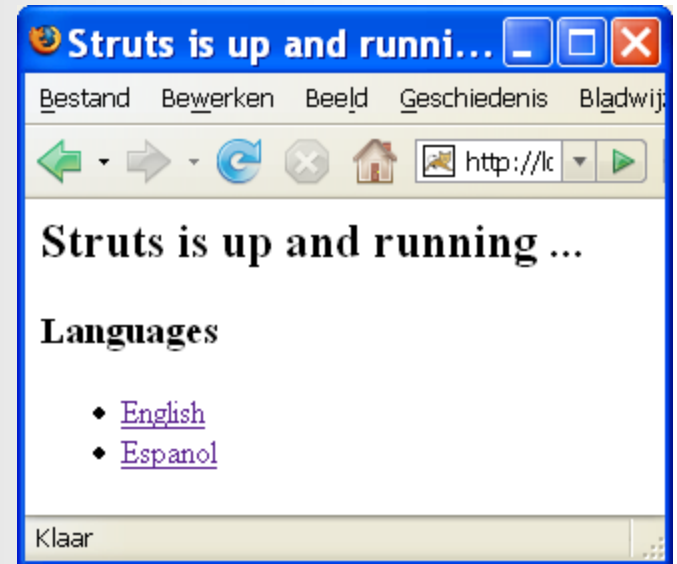
```
  </package>
```

```
</struts>
```

Beschrijving actie

View

- HelloWorld.jsp
- Gebruik taglib struts-tags
 - property-tag
 - Eigenschap van de bijhorende actieklasse tonen
 - url-tag
 - Maakt URL op basis van action-attribuut
 - Toevoegen parameters met param-tag
 - Gebruikt in a-tag
 - a-tag
 - Genereert een link
 - text-tag
 - Haalt tekst uit properties-bestand



View

- Toegang tot data via OGNL
 - Object Graph Navigation Language (OGNL)
 - Maakt gebruik van een context map (Map-object)
 - value-stack
 - Een stack met objecten
 - Tijdelijke objecten (bv. van een foreach, ...)
 - Action-object
 - Named objects (~scope): #application, #session, #request, #parameters, #attr (page, request, session, application)
 - %{ ... }
 - Te evalueren uitdrukking
 - value-attribuut: OGNL-uitdrukking
-

OGNL value-stack

■ Objecten op value-stack

■ Eigenschappen onmiddellijk opvragen

- Zonder verwijzing naar object

■ Bv. 2 objecten op value-stack

- Type Stad
- Type Foto

■ naam

■ omschrijving

■ url

Foto
•String getOmschrijving() •String getUrl()
Stad
•String getNaam()

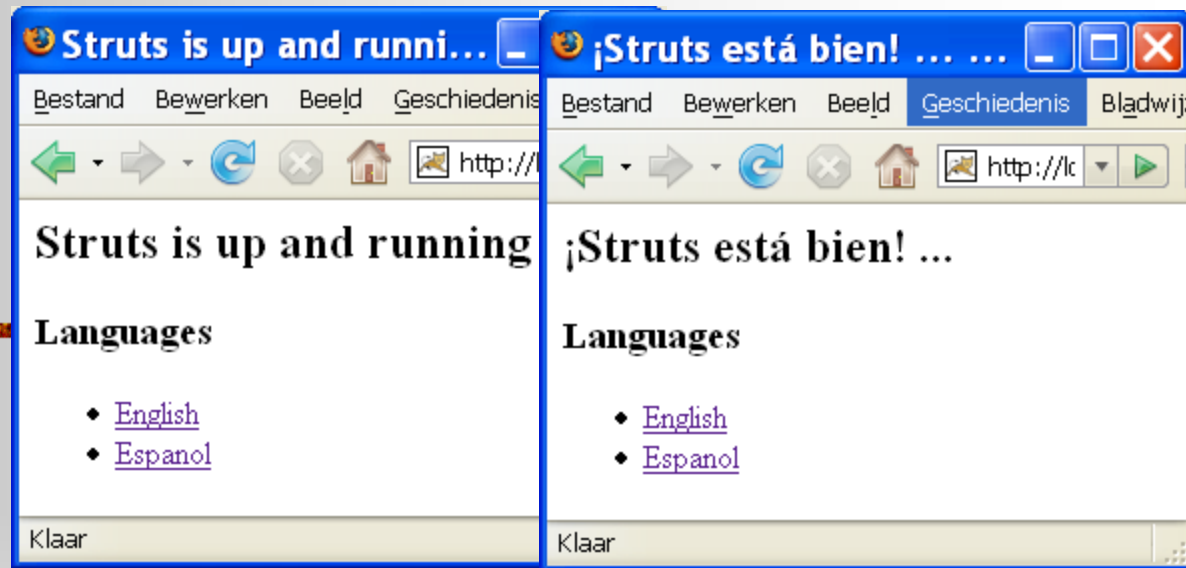
OGNL named objects

- `#parameters['thema']` of `#parameters.thema` (`~getParameter`)
 - `#request['view']` of `# request.view` (`~getAttribute`)
 - `#session['naam']` of `#session.naam`
 - `#application['catalogus']` of `#application.catalogus`
 - `#attr['view']` of `#attr.view`
-

Achter de schermen

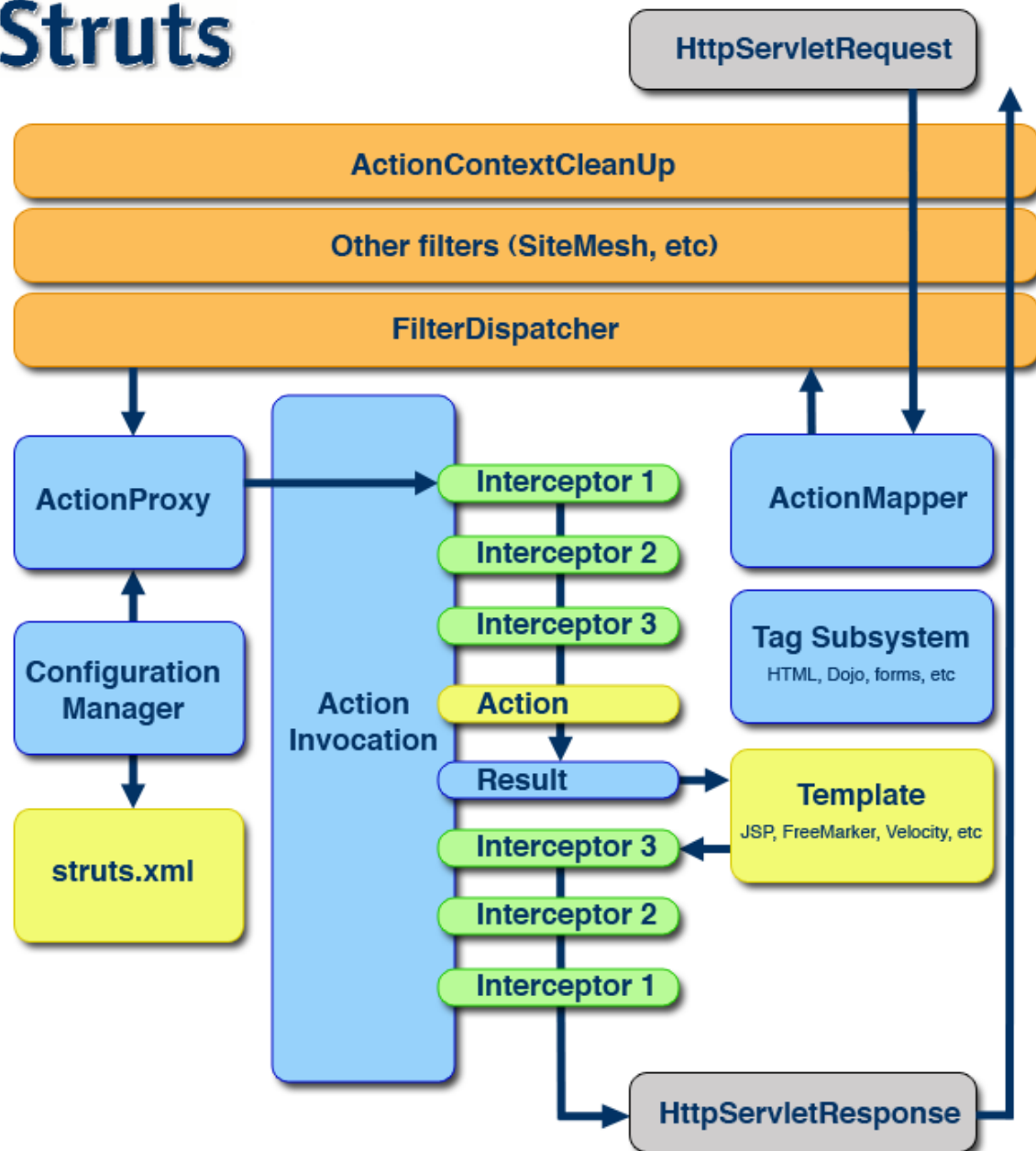
- HTTP-request voor HelloWorld.action
 - web.xml → doorspelen aan FilterDispatcher (ingang applicatie)
 - struts.xml → bijhorende Actie-object bepalen
 - Aanmaken Actie-object; oproepen execute-methode
 - Resultaat execute → view bepalen (JSP)
 - JSP parsen
 - property-tag → oproepen methode actie-object
 - HTTP-antwoord naar client
-

Localization



- Pagina's taalonafhankelijk maken
- Labels per taal in properties-bestanden
 - package.properties (voor het hele package)
 - NaamClass.properties (per klasse)
- Taal instellen via parameter
 - request_locale
 - Opgevangen door een interceptor (I18n Interceptor)
 - Plaatst locale op ActionContext

Struts



Key:

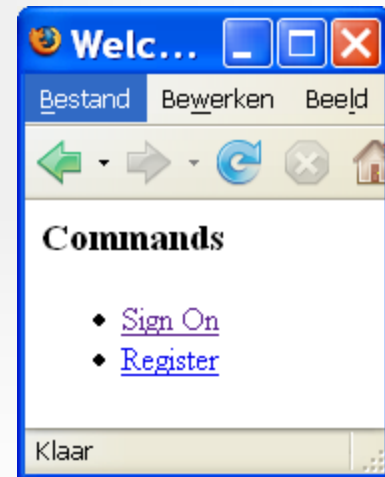
Servlet Filters Struts Core Interceptors User created

Interceptors

- Localization
 - Opvullen eigenschappen actie-object
 - Oproepen actie
 - Validatie
-

Voorbeeld 2

- <http://localhost:8084/struts/Welcome.jsp>
- <http://localhost:8084/struts/Welcome.action>



url samengevat

■ Link naar bron

```
<link href="<s:url value="/css/examplecss"/>" .... />
```

■ Link naar actie

```
<a href="<s:url action="Login_input"/>">Sign On</a>
```

■ Link met parameter

```
<s:url id="url" action="HelloWorld">
```

```
    <s:param name="request_locale">en</s:param>
```

```
</s:url>
```

```
<s:a href="%{url}">English</s:a>
```

Koppeling naar struts

■ Welcome.action en Welcome.jsp

- Geen actie
 - Geen class-attribuut
 - Default
 - ActionSupport
- Wel struts-tags

■ struts.xml (example.xml)

```
<action name="Welcome" >  
  <result>/Welcome.jsp  
  </result>  
</action>
```

■ Alternatieven

```
<action name="*" >  
  <result>/tutorial/{1}.jsp</result>  
</action>
```

■ of

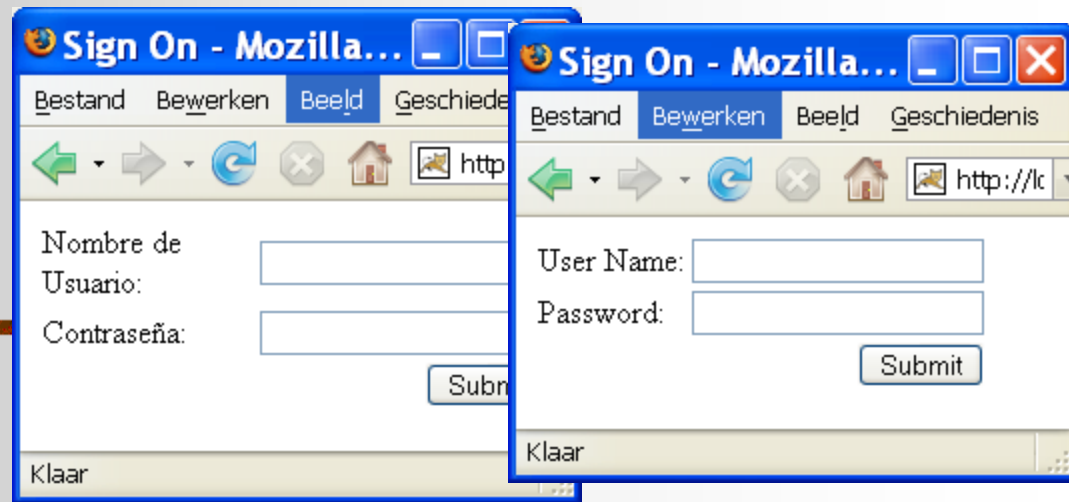
```
<action name="*"    
  class="example.ExampleSupport">  
  <result>/{1}.jsp</result>  
</action>
```

Lege actieklasse

struts.xml

- Kan meerdere action-elementen bevatten
 - Het eerste element dat toegepast kan worden, wordt toegepast
 - `<action name="*" ...>` steeds als laatste
-

Inlogformulier



- http://localhost:8084/blank_struts/example/Login_input.action
- Maakt gebruik van de struts-tags s:form, s:textfield, s:password en s:submit
- Actie formulier: Login.action
- key-attribuut (s:textfield en s:password)
 - Sleutel om data bijhorend label uit properties-bestand
 - Naam parameter van bijhorende input-tag

Verwerken loginformulier

- Actie: Login.java
 - Eigenschappen actie-klassen worden automatisch ingevuld met formulierparameters
 - Actie
 - Methode execute
 - Resultaat
 - INPUT → opnieuw naar login-formulier
 - SUCCESS → verder naar applicatie
-

Meerdere resultaten mogelijk

- In struts.xml
- Meerdere result-tags in de action-tag
 - name-attribuut = resultaat actie
 - Standaard: success
 - type-attribuut = type resultaat
 - Doorsturen naar JSP, naar andere actie, ...

```
<action name="Login" class="example.Login">  
  <result name="input">/Login.jsp</result>  
  <result type="redirect-action">Menu</result>  
</action>
```

Validatie

- Controle invoer gebruiker
 - XML-document
 - Naamactie-validation.xml
 - Bv. Login-validation.xml
 - Naam te valideren veld, type validatie, foutboodschap (kan uit .properties-bestand → key)
 - Eerste keer tonen loginformulier
 - Geen validatie
-

Validatie

■ Eerste keer geen validatie

■ Speciale methode van ActionSupport

- input → resultaat = input

■ Configuratie in struts.xml (example.xml)

```
<action name="Login_*" method="{1}" class="example.Login">  
  <result name="input">/Login.jsp</result>  
  <result type="redirect-action">Menu</result>  
</action>
```

■ Actie: Login_input

- Validatie + execute worden NIET uitgevoerd

■ Anders

■ Validatie + execute

- Bij fout → input
-

Beschikbare validatie

- required validator
 - Niet null?
- requiredstring validator
 - Niet null? Niet allemaal spaties?
- int validator
 - Int? In interval?
- date validator
 - Int? In interval?
- expression validator
 - Logische OGNL-uitdrukking
- fieldexpression validator
 - Logische OGNL-uitdrukking, gekoppeld aan een veld
- email validator
 - Email?
- url validator
 - url?
- conversion validator
 - Conversie gelukt?
- stringlength validator
 - Lengte in interval?
- regex validator
 - Voldoet aan reguliere uitdrukking?

Voorbeelden validatie

```
<validators>
```

```
<!-- Field Validators for email field -->
```

```
<field name="email">
```

```
<field-validator type="required">
```

```
<message>...</message>
```

```
</field-validator>
```

```
<field-validator type="email">
```

```
<message>...</message>
```

```
</field-validator>
```

```
</field> ...
```

```
<field name="bar">
```

```
<field-validator type="required">
```

```
<message>...</message>
```

```
</field-validator>
```

```
<field-validator type="int">
```

```
<param name="min">6</param>
```

```
<param name="max">10</param>
```

```
<message>bar must be between  
${min} and ${max}, current value is  
${bar}.</message>
```

```
</field-validator>
```

```
</field> ...
```

Voorbeelden validatie

```
<field name="bar2">
  <field-validator type="regex">
    <param name="expression">[0-9],[0-9]</param>
    <message>...</message>
  </field-validator>
</field>

<field name="date">
  <field-validator type="date">
    <param name="min">12/22/2002</param> <param
    name="max">12/25/2002</param> <message>...</message>
  </field-validator>
</field>
```

ActionContext

- Context waarin een actie wordt uitgevoerd

```
import com.opensymphony.xwork2.ActionContext;  
ActionContext context = ActionContext.getContext();
```

- Toegang tot

- Sessie
 - Parameters
 - ...
-

Referenties

- Starting with Struts 2, Ian Roughley
(<http://www.infoq.com/minibooks/starting-struts2>)
 - Struts tutorial
(<http://struts.apache.org/2.1.8/docs/bootstrap.html>)
 - Struts guide
(<http://struts.apache.org/2.1.8/docs/guides.html>)
-

Referenties

- Struts API
(<http://struts.apache.org/2.x/struts2-core/apidocs/index.html>)
 - Struts tags
(<http://struts.apache.org/2.1.8/docs/tag-reference.html>)
-