HANDS ON STRUTS2

Ian Roughley, Consultant

S295739

Java

Sun microsystems

Learn how to combine Struts2
with complimentary technologies
to provide a complete web application stack

GOAL

What I Know

My Goal

How Do I Find Them?

How Do I Order Them?

How Do I Eat Them?

Things To Know (Agenda)

- How to start a new project
- What application features are available
- How to access business services
- What page template options are there
- What Ajax options are available

Things To Know (Agenda)

- How to start a new project
- What application features are available
- How to access business services
- What page template options are there
- What Ajax options are available

Starting a New Project

- Manually
  - Configure `web.xml`
  - Write configuration and code
- Using Maven2
  - Generate project
  - Working baseline
  - Integrated servlet engine

Basic Elements

- Configuration files
  - `web.xml`
  - `struts.xml`
- Application elements
  - Action class
  - JSP template
- A way to run the application

web.xml

```
<web-app>

  <filter>
    <filter-name>action2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.FilterDispatcher
    </filter-class>
  </filter>

  <filter-mapping>
    <filter-name>action2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

</web-app>
```

struts.xml

```
<struts>

  <package name="myPackage" namespace="/"
      extends="struts-default">

    <action name="index"
        class="com.fdar.s2.IndexAction">
      <result name="success">/jsp/index.jsp</result>
    </action>

  </package>

</struts>
```

Actions

```
public class IndexAction {

  private String name;

  public String getName() { return name; }

  public void setName(Stringe newName) {
    name = newName;
  }

  public String execute() throws Exception {
    …
    return "success";
  }
}
```

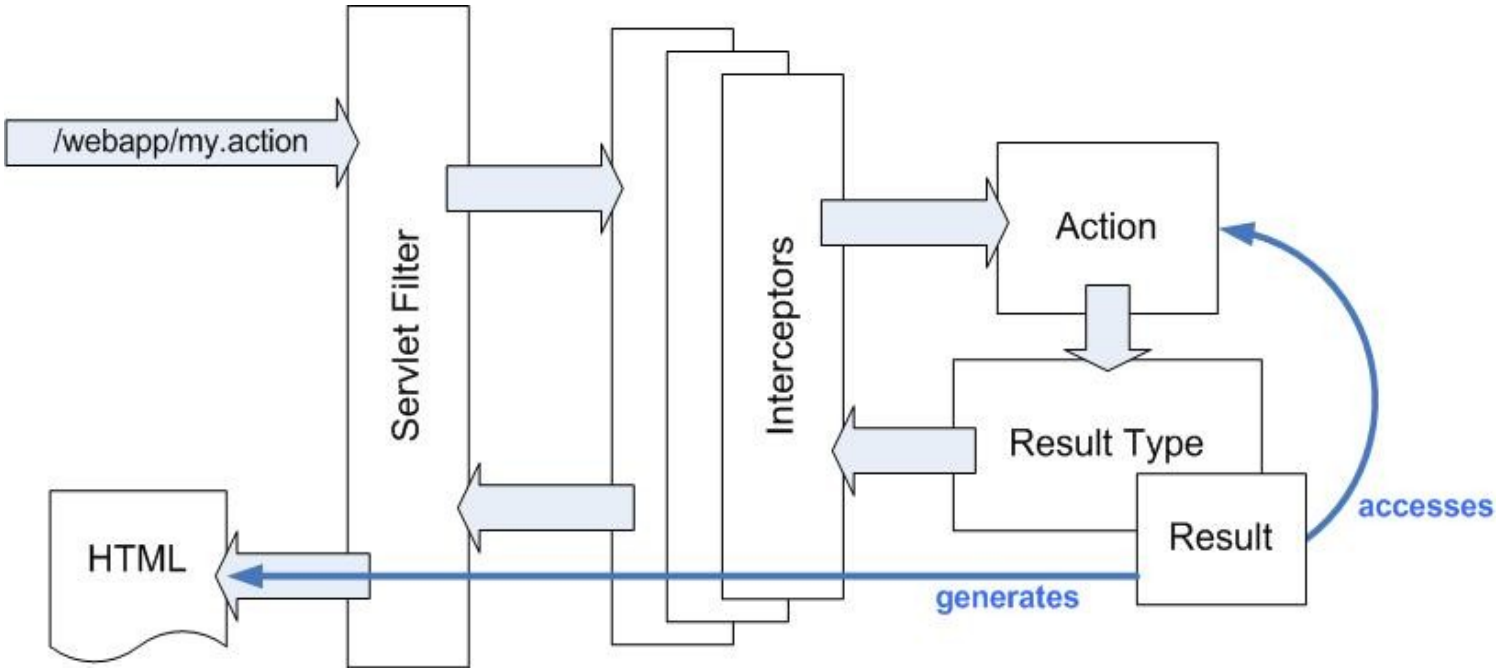
JavaServer Pages™ (JSP™) technology

```
<%@taglib prefix="s" uri="/struts-tags" %>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Index</title>
</head>

<body>
  <s:form namespace="/" action="index">
       <s:textfield label="What is your name?" name="name" />
       <s:submit />
  </s:form>
</body>
</html>
```

So What's Changed?

Feature	Struts1	Struts2
Configuration	`struts-config.xml`	`struts.xml`
Actions	Action	POJO
Data	HTTP Objects / `ActionForm`	Action / Model
Logic Method	`execute()`	Any Method
Multiple Logic	`DispatchAction`	Any Method
Threading Model	Thread-safe	POJO

Putting It Together



Available Interceptors

- `exception`
- `modelDriven`
- `params`
- `prepare`
- `validation`
- `workflow`
- `fileUpload`
- `checkbox`
- `profiling`
- `roles`
- `servletConfig`
- `token`

Configuring Interceptors

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

    <interceptor-stack name="validationWorkflowStack">
      <interceptor-ref name="basicStack"/>
      <interceptor-ref name="validation"/>
      <interceptor-ref name="workflow"/>
    </interceptor-stack>

    <default-interceptor-ref name="defaultStack"/>

  </package>
</struts>
```

Interceptors per Action

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

    <action name="test"
        class="com.fdar.actions.MyAction">
      <result name="success">/result.jsp</result>
      <interceptor-ref name="defaultStack"/>
    </action>

  </package>
</struts>
```

Available Result Types

- `dispatcher`
- `freemarker`
- `velocity`
- `xslt`
- `plainText`
- `chain`
- `httpheader`
- `redirect`
- `redirectAction`
- `stream`

Plug-ins

- Adds new functionality
  - Interceptors, Interceptor Stacks & Result Types
- Structure is a web application
  - **`struts-plugin.xml`** for configuration
  - Use **`JAR`** for deployment
- Place in **`WEB-INF/lib`** directory

Things To Know (Agenda)

- How to start a new project
- What application features are available
- How to access business services
- What page template options are there
- What Ajax options are available

Data Conversion

- HTML Forms ←→ Java™ technology objects/type

```
<s:textfield name="price" />

public void setPrice(float price)
public float getPrice()
```

Types Converted

- Built in types
  - **`boolean`**, **`Boolean`**, **`char`**, **`Character`**, **`int`**, **`Integer`**, **`float`**, **`Float`**, **`long`**, **`Long`**, **`double`**, **`Double`**, and **`Date`**
- Implement custom converters

Master-Detail Action

```
public class MyAction {

  private List<Item> items;

  public List<Item> getItems() {
    return items;
  }

  public void setItems(List<Item> items) {
    this.items = items;
  }
  …
}
```

Master-Detail Forms

```
<input type="text" name="items[0].name" />
<input type="text" name="items[0].description" />

<input type="text" name="items[1].name" />
<input type="text" name="items[1].description" />
```

Viewing and Updating

```
<s:iterator value="items" status="stat">

    <s:hidden name="items[%{#stat.index}].id" />

    Item #<s:property value="#stat.index+1" /> <br/>

    <s:textfield label="Name"
        name="items[%{#stat.index}].name" />

    <s:textfield label="Description"
        name="items[%{#stat.index}].description" />

</s:iterator>
```

Validation

- Annotations for
  - Required Fields, Ranges, Email, Expressions, Regular Expressions,
    URLs, Visitor
- Can be configured via XML
- **`validate()`** method in **`Validateable`**
- Can create custom validators

Action Configuration

```
@Validation
public class MyAction {

  private String name;

  public String getName() {
    return name;
  }

  @RequiredStringValidator(
      message="Please enter a name", trim=true)
  public void setName(String name) {
    this.name = name;
  }
  …
}
```

Another Action Configuration

```
@Validation
public class MyAction {

  private String name;
  public String getName() { return name; }
  public void setName(String name) { this.name = name; }

  @Validations( expressions = {
        @ExpressionValidator(message="Name cannot be Bob",
            expression="name!='bob'" )
   })
   public String execute() throws Exception {
        …
   }
}
```

Validation Result

- No changes to JSP
  - **`errorMessage`** and **`errorLabel`** CSS classes
- **`validation`** and **`workflow`** interceptors
- Requires a configured **`INPUT`** result

Things To Know (Agenda)

- How to start a new project
- What application features are available
- How to access business services
- What page template options are there
- What Ajax options are available

web.xml

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        classpath*:applicationContext*.xml
    </param-value>
</context-param>

<listener>
    <listener-class>
        org.springframework.web.context.ContextLoaderListener
    </listener-class>
</listener>
```

applicationContext.xml

```
<beans
    xmlns="http://www.springframework.org/schema/beans"
    … >

    <bean id="userService"
        class="com.fdar.s2.services.UserServiceImpl" />

</beans>


public class UserServiceImpl implements UserService {
    …
}
```

The Action

```
public class MyAction extends ActionSupport {

  private UserService service;

  public void setUserService( UserService userService) {
    this.service = userService;
  }

  public String execute() {
    …
    return SUCCESS;
  }
}
```

Injection Mechanism

- Options
  - `name` (default)
  - `type`
  - `constructor`
  - `auto`

```
<constant value="type"
  name="struts.objectFactory.spring.autoWire" />
```

Things To Know (Agenda)

- How to start a new project
- What application features are available
- How to access business services
- What page template options are there
- What Ajax options are available

SiteMesh

- Configured externally
- Decorates HTML via URLs
  - Can use **`META`** tags to specify template
- Configuration options include
  - Configuration File
  - Agent
  - Parameter
  - Printable

Update `web.xml`

```
<filter>
    <filter-name>sitemesh</filter-name>
    <filter-class>
        com.opensymphony.module.sitemesh.filter.PageFilter
    </filter-class>
</filter>

<filter-mapping>
    <filter-name>sitemesh</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

decorators.xml

```
<decorators defaultdir="/WEB-INF/decorators">

    <excludes>
        <pattern>/styles/*</pattern>
        <pattern>/scripts/*</pattern>
        <pattern>/api/*</pattern>
        <pattern>/*.html</pattern>
    </excludes>

    <decorator name="main" page="main.jsp">
        <pattern>/*</pattern>
    </decorator>

</decorators>
```

main.jsp

```
<%@taglib prefix="decorator"
  uri="http://www.opensymphony.com/sitemesh/decorator" %>

<html>
<head>
    <title><decorator:title default="Struts Starter"/></title>
    <link href="<s:url value='/styles/main.css'/>"
          rel="stylesheet" type="text/css" media="all"/>
    <decorator:head/>
</head>
<body id="page-home"
      onload="<decorator:getProperty property="body.onload"/>">

    <decorator:body/>

</body>
</html>
```

Apache Tiles

- Part of the application configuration
- Developer specifies template(s)
- Plug-in provided functionality
  - Uses Result Type

Update web.xml

```
<listener>
    <listener-class>
        org.apache.struts2.tiles.StrutsTilesListener
    </listener-class>
</listener>
```

Configure `struts.xml`

```
<struts>
  <package name="myPackage" extends="tiles-default">
    <action name="index"
        class="com.fdar.s2.IndexAction">
      <result name="success" type="tiles">indexTile</result>
    </action>
  </package>
</struts>
```

tiles.xml

```
<tiles-definitions>

  <definition name="indexTile"
      template="/tiles/layout.jsp">
    <put name="title" value="Tiles Showcase"/>
    <put name="header" value="/tiles/header.jsp"/>
    <put name="body" value="/tiles/indexBody.jsp"/>
  </definition>

</tiles-definitions>
```

layout.jsp

```
<%@ taglib prefix="tiles"
        uri="http://tiles.apache.org/tags-tiles" %>

<html>
<head>
  <title><tiles:getAsString name="title"/></title>
</head>

<body>
    <tiles:insertAttribute name="header"/>

    <p id="body">
        <tiles:insertAttribute name="body"/>
    </p>
</body>
</html>
```

Things To Know

- How to start a new project
- What application features are available
- How to access business services
- What page template options are there
- What Ajax options are available

Using Ajax Tag Libraries

- Use the **`ajax`** theme
  - **`div`**, **`a`**, **`submit`**, **`tabbedpanel`**, **`tree`**
- Utilizes the Dojo Toolkit
- Will change between 2.0.x and 2.1.x

Form Example

```
<div>
  <s:form namespace="/search" action="search" method="POST">
    <s:textfield label="Search for" name="titlePartial" />
    <s:submit label="Go"/>
  </s:form>
</div>

<div id="main">
  …
</div>
```

Ajax Form Example

```
<s:head theme="ajax" />

<div>
  <s:form namespace="/search" action="search" method="POST">
    <s:textfield label="Search for" name="titlePartial" />
    <s:submit label="Go" theme="ajax" targets="main" />
  </s:form>
</div>

<div id="main">
  …
</div>
```

Using the `xslt` Result Type

```
<struts>
  <package name="services"
        namespace="/api" extends="struts-default">

    <action name="index"
        class="com.fdar.s2.IndexAction">
      <result type="xslt">
        <param name="exposedValue">items</param>
      </result>
    </action>

  </package>
</struts>
```

XML Result

```
<result>
    <item>
        <id>1</id>
        <name>Book</name>
    </item>
    <item>
        <id>2</id>
        <name>Scroll</name>
    </item>
</result>
```

Using the `json` Result Type

```
<struts>
  <package name="services"
        namespace="/api" extends="json-default">

    <default-interceptor-ref name="json" />

    <action name="index"
        class="com.fdar.s2.IndexAction">
      <result type="json">
        <param name="root">items</param>
      </result>
    </action>

  </package>
</struts>
```

JSON Result

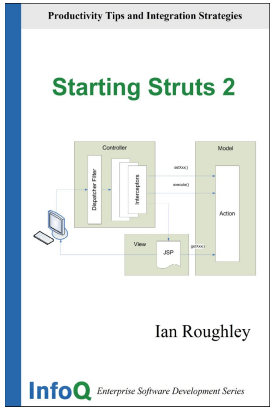
```
[
  {
    "id": 1,
    "name": "Book"
  },
  {
        "id": 2,
        "name": "Scroll"
  }
]
```

Summary

- Simple programming model
- Everything you need is available
- Plays well with others
- Looks good in the latest fashions

Goal Achieved

Books



http://www.apress.com/book/view/978159059903

http://www.infoq.com/minibooks/starting-struts2

For More Information

- Practical Apache Struts 2 Web 2.0 Projects
  - `http://www.apress.com/book/view/1590599039`
- Apache Struts2 Project
  - `http://struts.apache.org`
- Annotations
  - `http://struts.apache.org/2.x/docs/annotations.html`
- Ajax Tags
  - `http://struts.apache.org/2.x/docs/ajax-tags.html`
- Available Plug-ins
  - `http://cwiki.apache.org/S2PLUGINS/home.html`

Questions

Ian Roughley

`http://www.fdar.com`

`ian@fdar.com`

THANK YOU

Ian Roughley, Consultant

S295739