

Struts 2

Syllabus Topics

Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC, pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results and Result Types, Value Stack/OGNL

Introduction

The struts 2 framework is used to develop MVC-based web application. The Struts 2 framework is used to develop MVC (Model View Controller) based web applications.

Syllabus Topic : Basic MVC Architecture

6.1 Basic MVC Architecture

- Q. Explain Basic MVC Architecture in detail.
- Q. Discuss the different stages of basic MVC architecture.

- Model View Controller or MVC is a software design pattern for developing web applications. A Model View Controller pattern is consisting of three parts –
 - o **Model** – It is the lower most level of the pattern which is responsible for maintaining data.
 - o **View** – view is responsible for displaying all or a part of the data to the user.
 - o **Controller** – It is a software Code that manages the interactions between the Model and View.
- MVC is most popular because it separates the application logic from the user interface layer. Basically the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View.

The View then uses the data prepared by the Controller to generate a final presentable response as shown in Fig. 6.1.1

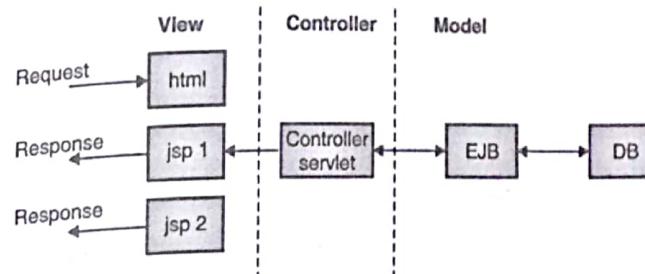


Fig. 6.1.1 : Basic MVC Architecture

1. The Model

The model represents enterprise data and business rules that govern access to and updates to this data. It manages the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

2. The View

The view signifies the user interface, which is nothing but presentation of data in a particular format, triggered by a controller's decision to present the data. They are mainly script-based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

3. The Controller

The controller translates view interactions into actions to be performed by the model. It is responsible for responding to the user and perform appropriate interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

Syllabus Topic : Struts 2 Framework Features

6.2 Struts 2 Framework Features

- Q. Explain struts 2 framework in detail.

- Struts is a java based framework which separate the application logic that interact with the database from an HTML page that form the response.

☞ Struts framework architecture

- A framework is a group of services that provide developers with common set of functionality to be reused amongst multiple applications.
- Struts is a java based framework which separate the application logic that interact with the database from an HTML page that form the response.
- Strut is not a technology, it is a framework that can be used along with other java based technologies.
- Strut makes ease of enterprise application development with flexible and extensible application architecture and custom tags.
- Strut 2 is a web application framework based on the OpenSymphony WebWork framework that implements the MVC design pattern.

☞ Components of Struts 2 framework

- The Struts2 framework has five core components :

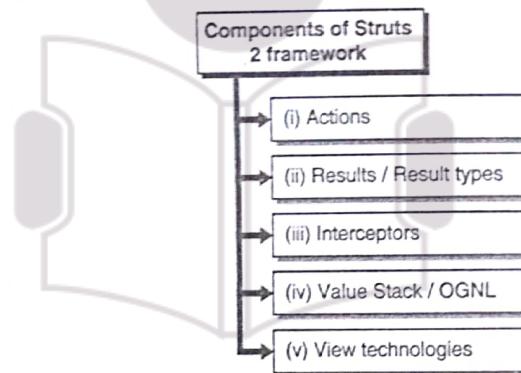


Fig. C6.1 : Components of Struts 2 framework

→ (i) Actions

Model is implemented in struts using actions. Actions include the business logic and interact with the persistence storage to store, retrieve and manipulate data.

→ (ii) Results / Result types

Results are the UI that represent the user requirements from the application onto the client browser.

→ (iii) Interceptors

Interceptors are used to implement the controller in strut applications. It handles the request from the user and selects the appropriate view in return. In Struts ActionServlet is used as controller.

→ (iv) Value Stack / OGNL

It provides common thread, linking and enabling integration between the other components.

→ (v) View technologies

View can be a combination of result types and results. In Struts the view is implemented using JSP (Java Server Pages).

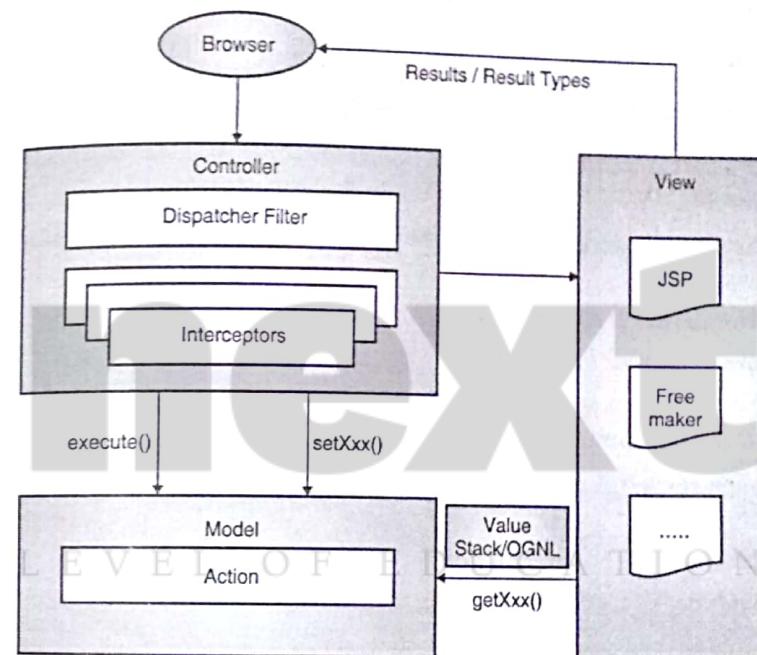


Fig. 6.2.1 : Struts 2 life cycle scenario

- (i) User sends a request to the server for requesting the resource.
- (ii) FilterDispatcher accepts the request and determines the appropriate Action. The ActionServlet receives the request and perform :
 - It stores all the request values into a JavaBean class.
 - Determines which action class to invoke for processing the request.
- (iii) Configured interceptors perform the functionalities like validation, file upload etc.
- (iv) Selected action is executed to perform the operation.
 - Model component helps the action class to processes the request. The model interacts with the database and process the request.

- After completing the request processing the Action class returns an ActionForward to the controller.
- (v) Configured interceptors are applied to the post-processing if required.
- (vi) Based on the ActionForward the controller will invoke the appropriate view.
 - Result is prepared by the view and returns the result to the user.
 - The HTTP response is rendered back to the user by the view component.

6.2.1 Struts 2 Features

Q. Explain the advantages or features of struts 2.0.

The features of struts 2 framework are :

1. Configurable MVC (Model View and Controller) components

- Struts 2 framework provides all the components view and action information in struts.xml file.
- It Promote less configuration with the help of default values for various settings.
- Provide method to re-factor the existing code specifications into the plug-in framework.

2. POJO (Plain Old Java Object) based actions

It is Action based framework. Any java class with execute() method can act an Action class.

3. AJAX support

- It supports Ajax technology used to make asynchronous request i.e. it doesn't block the user.
- It sends only the required field data to the server not all. So it makes the performance fast.
- It supports the Ajax features as :
 - o Ajax based client side validation.
 - o Rich pub-sub event model.
 - o Interactive auto complete tag.
 - o Remote form submission support.
 - o Advanced DIV template that provide dynamic reloading of partial HTML.
 - o Advanced template with ability to load and evaluate JavaScript remotely.

4. Integration support

Struts 2 applications can be integrated with other frameworks such as Hibernate, Web Work and Spring.

5. Various Result Types

- It supports JSP, Freemarker, Velocity etc. technologies as result types.
- Tag mark-ups in Struts2 can be modified using Freemarker templates. It does not require JSP or java knowledge. Basic Knowledge in HTML, XML and CSS is only required to modify the tags.

6. Various Tag support

- It provides various types of tags such as UI tags, Data tags, control tags etc to ease the development of enterprise application.
- Reduces the XML configuration by adding plug-in and improves REST support. Annotation based configuration are used to eliminate XML configuration.

7. Theme and Template support

- Style sheet based mark-up allow creating consistent pages with less code specifications.
- It provides three types of theme support : xhtml, simple and css_xhtml.
- Support for generating views using templates.
- Manual testing time could be reduced with the help of built-in debugging tools

8. Object Graph Navigation Language (OGNL)

Object Graph Navigation Language (OGNL) is an expression language that traverses the entire framework.

6.2.2 Struts 2 Disadvantages

Q. Explain the disadvantages of struts 2.

1. Bigger learning curve : To use MVC with Struts, programmer should be familiar with the standard JSP, Servlet APIs and a large and elaborate framework.
2. Poor documentation : Compared to the standard servlet and JSP APIs, Struts has fewer documentations.
3. Less transparent : With Struts Java-based Web applications makes difficult to understand the framework.
- A framework is a group of services that provide developers with common set of functionality to be reused amongst multiple applications.
- Struts is a java based framework which separate the application logic that interact with the database from an HTML page that form the response.
- Strut is not a technology, it is a framework that can be used along with other java based technologies.
- Strut makes ease of enterprise application development with flexible and extensible application architecture and custom tags.

- Struts 2 is a web application framework based on the OpenSymphony WebWork framework that implements the MVC design pattern.

Syllabus Topic : Struts 2 MVC Pattern

6.3 Struts 2 MVC Design Pattern

- Q. Explain the application flow in MVC with respect to struts.
- Q. Explain struts MVC design pattern in detail.
- Q. Discuss the different components of struts design pattern

- Struts 2 follows the Model-View-Controller(MVC) design patterns as :

(i) Action - Model

Model is implemented in struts using actions. Actions include the business logic and interact with the persistence storage to store, retrieve and manipulate data.

(ii) Result - View

View can be a combination of result types and results. In Struts the view is implemented using JSP (Java Server Pages)

(iii) FilterDispatcher - Controller

FilterDispatcher accepts the request and determines the appropriate Action. The ActionServlet which act as FilterDispatcher receives the request and perform :

- It stores all the request values into a JavaBean class.
- Decides which action class to invoke for request processing.

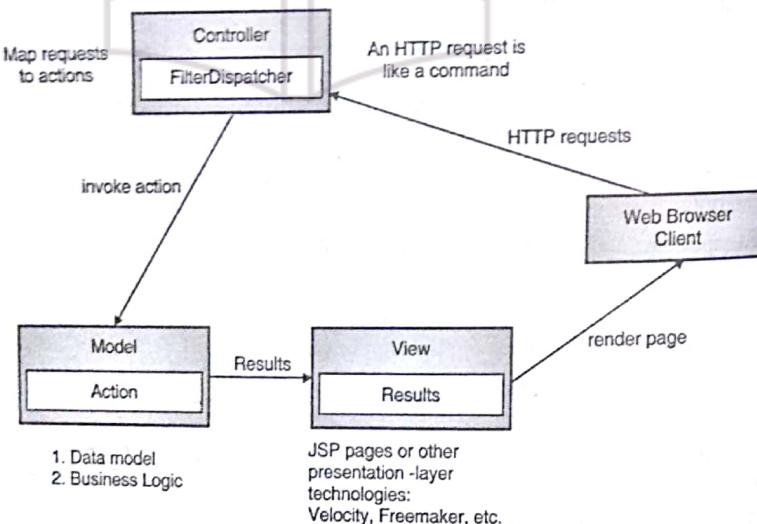


Fig. 6.3.1 : MVC module interactions in Struts

- Each action is defined and implemented by the framework defined execute() method. Action gets the information from the request and performs the business logic and passes the result to JavaBean or EJB. Action is an encapsulation of requests and is placed in ValueStack.

- Model component consist of the data storage and business logic. Model returns a result to the controller that decides which output page to be sent as the response.

- View is the presentation component of MVC pattern. JSP files with other techniques such as tiles, velocity, freemarker, etc. can be combined to provide a better presentation layer. Controller map, incoming HTTP requests to the appropriate actions.

- Mappings are defined by XML-based configuration (struts.xml) or Java annotations.

Syllabus Topic : Request Life Cycle

6.4 Request Life Cycle

- Q. Explain the request life cycle of struts in detail.

Refer Fig. 6.3.1 we can understand the work flow through user's request life cycle in Struts 2 as follows :

1. Initially User sends a request to the server for requesting for some resource (for example- pages).
2. By looking at the request The Filter Dispatcher determines the appropriate Action.
3. Validation, file upload etc. functionalities are applied by Configured interceptor .For example- Selected action is performed based on the requested operation.
4. Again, configured interceptors are applied to do any post-processing if required.
5. Finally, the result is prepared by the view and returns the result to the user.

Syllabus Topic : Examples

6.5 Examples

Here we will see two different example

1. Struts example without IDE
2. Struts example with Myeclipse IDE

6.5.1 Struts Example without IDE

Steps of creation of application Struts 2

Step 1 : Create the directory structure



Step 2 : Create input page (.jsp)



Step 3 : Provide the entry of Controller in (web.xml) file



Step 4 : Create the action class (.java)



Step 5 : Map the request with the action in (struts.xml) file and define the view components



Step 6 : Create view components (.jsp)



Step 7 : Load the jar files



Step 8 : Start server and deploy the project

Step 1 : Create the directory structure

Here, struts.xml file must be located in the classes folder. The directory structure of struts 2 is same as servlet/JSP as shown in Fig. 6.5.1.

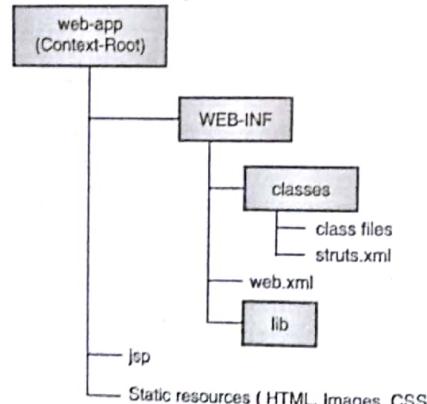


Fig. 6.5.1 : Directory structure

Step 2 : Create input page (First.jsp)

- Here will creates a JSP page form using struts User interface (UI) tags. We need to specify uri /struts-tags prefix="s" %>
- Here, we will use s:form to create a form, s:textfield to create a **text field**, s:submit to create a **submit button**.

First.jsp

```
<%@ taglib uri="/struts-tags" prefix="s" %>
<s:form action="Painting">
  <s:textfield name="pid" label="Painting Id"></s:textfield>
  <s:textfield name="pname" label="Painting Name"></s:textfield>
  <s:textfield name="price" label="Painting Price"></s:textfield>
  <s:submit value="save"></s:submit>
</s:form>
```

Step 3 : Provide the entry of Controller in (web.xml) file

Here , StrutsPrepareAndExecuteFilter class works as the controller. Struts 2 uses filter for the controller. It is implicitly provided by the struts framework.

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

Step 4 : Create the action class (Painting.java)

This will be a simple bean class. Here in struts 2, action is POJO (Plain Old Java Object). It has one extra method **execute** which is invoked by struts framework by default.

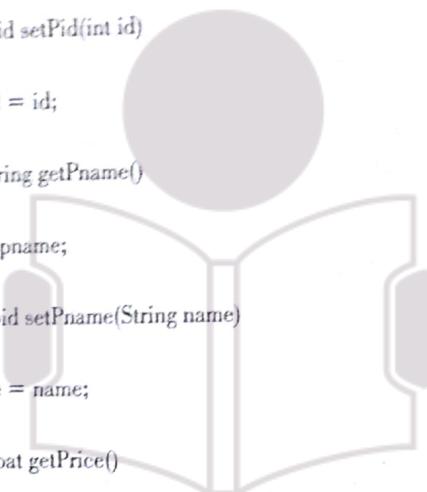
Painting.java

```

package myPack;

public class Painting
{
    private int pid;
    private String pname;
    private float price;
    public int getPid()
    {
        return pid;
    }
    public void setPid(int id)
    {
        pid = id;
    }
    public String getPname()
    {
        return pname;
    }
    public void setPname(String name)
    {
        pname = name;
    }
    public float getPrice()
    {
        return price;
    }
    public void setPrice(float price)
    {
        this.price = price;
    }
    public String execute()
    {
        return "success";
    }
}

```



Step 5 : Map the request in (struts.xml) file and define the view components

- struts.xml file plays an important role. It provides information about the action to struts framework and let it decides which result to be invoked. It has many elements such as struts, package, action and result.
- Struts : It represents an application. Struts element is the root element.
- Package : It generally extends the struts-default package where many interceptors and result types are defined. Here package represents a module of the application. Package element is the sub element of struts.
- Action : It is the sub element of package. Package element contains name, class and method attributes. If we don't specify name attribute by default execute() method will be invoked for the specified action class. Action represents an action to be invoked for the incoming request.
- Result : It represents a view (result) that will be invoked. It is the sub element of action. Struts framework checks the string returned by the action class, if it returns success, result page for the action is invoked whose name is success or has no name.
 - o Result element has **name** and **type** attributes. Both are optional.
 - o If we don't specify the **result name**, by default success is assumed as the result name. If you don't specify the **type** attribute, by default dispatcher is considered as the default result type.

struts.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>
<package name="default" extends="struts-default">

<action name="painting" class="myPack.Painting">
<result name="success">Final.jsp</result>
</action>
</package>
</struts>

```

Step 6 : Create view components (Final.jsp)

The s:property tag returns the value for the given name, stored in the action object.

Final.jsp

```

<%@ taglib uri="/struts-tags" prefix="s" %>
Painting Id:<s:property value="pid"/><br/>

```

Painting Name: <s:property value="pname"/>

Painting Price: <s:property value="price"/>

Step 7 : Load the jar files

To run this application, we need to have the struts 2 jar files. Download it and put jar files in the lib folder of your project.

Step 8 : Start server and deploy the project

Finally, start the server and deploy the project and access it. Run and see the output.

6.5.2 Struts Example with MyEclipse IDE-

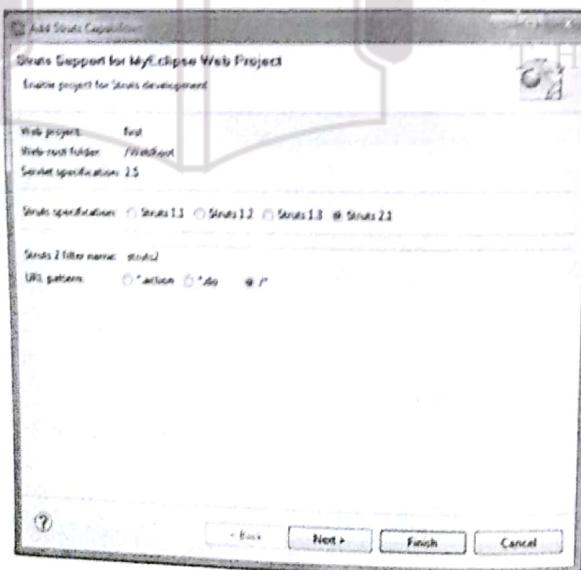
Here, we don't need to care about the jar files because MyEclipse provides these jar files. Following are the steps :

Step 1 : Create a web project

To create web project, click on the file menu - new - project - web project - write the project name e.g. Mystruts - finish.

**Step 2 : Add struts 2 capabilities**

To add struts 2 capabilities, select your project - click on the myeclipse menu - add project capabilities - add struts capabilities.



Select the 2.1 and /* as the url pattern - finish.

**Step 3 : Create input page (First.jsp)**

- It uses struts core tags to create a form with fields.
- First.jsp (refer to above example same)

**Step 4 : Create the action class (Painting.java)**

- It is the simple action class containing properties with setters and getters. It contains the execute method also for defining the business logic.
- Painting.java (refer to above example same)

**Step 5 : Map the request in (struts.xml) file and define the view components**

- This xml file registers the action and view components.
- struts.xml (refer to above example same)

**Step 6 : Create view components (Final.jsp)**

- This jsp page displays the information set in the action object.
- Final.jsp (refer to above example same)

**Step 7 : Start server and deploy the project**

To start the server and deploy the project, right click on your project - Run As - MyEclipse server application.

Syllabus Topic : Configuration Files**6.6 Configuration Files**

- Explain the significance of struts.xml and web.xml file in Strut application.
- Write a short note on any two configuration file of Struts.

Struts contains few important configuration files like **web.xml**, **struts.xml**, **strutsconfig.xml** and **struts.properties**, out of which **web.xml** and **struts.xml** plays an important role.

6.6.1 struts.xml File

The struts.xml file contains the configuration information required for an action element of Strut2 application.

Elements of struts.xml file

- (i) **<struts>** : is the root tag element, within that different package details are included using **<package>** tags.
- (ii) **<constant>** : is used to override the properties defined by default properties. It consists of two attributes as name and value.
- (iii) **<package>** : allows separation and modularization of the configuration. It is useful for large project where it can be divided into different modules.

For example if an application has three modules - client_module, supplier_module and inventory_module, then three packages could be created and store all associated actions in the respective package.

Attributes of package tag

- **Name** : unique identifier for the package.
- **Extends** : name of the package it is extend from. By default, struts-default is the base package.
- **Abstract** : If true, the package won't be available for end user.
- **Namespace** : Unique namespace for the actions.

(iv) **<action>** : It corresponds to every URL(name)with its associated class and method name. Method will be executed when the URL get accessed.

(v) **<result>** : String result returned from the action method is specified in the name attribute of **<result>** tag. Results have optional name and type attributes. The default name value is "success".

Example : struts.xml for Hello World application

```

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

<package name="helloworld" extends="struts-default">
<action name="hello" class="mystrut.HelloWorldAction" method="execute">
<result name="success">/HelloWorld.jsp</result>
</action>

<!-- more actions can be listed here -->
</package>

```

<!-- more packages can be listed here -->

</struts>

Struts.xml file content can split into multiple xml files and import it later as:-

```

<struts>
<include file="my-struts1.xml"/>
<include file="my-struts2.xml"/>
</struts>

```

my-struts1.xml and my-struts2.xml are two separate xml files where Struts.xml contents are split and stored

6.6.2 web.xml File

- The web.xml configuration file is a J2EE configuration file that determines how elements of the HTTP request are processed by the servlet container. It needs to be configured for Struts2 applications.
- Web.xml file provides an entry point for a web application.
- The entry point of Struts2 application will be a filter defined in the deployment descriptor. Thus an entry is defined by FilterDispatcher class in web.xml.

Example : Content of web.xml file for Strut2 application

```

<web-app>
<display-name>Struts 2</display-name>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>

<filter>
<filter-name>struts2</filter-name>
<filter-class>
mystrut.FilterDispatcher
</filter-class>
</filter>

<filter-mapping>
<filter-name>struts2</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>

```

6.6.3 The Struts-config.xml File

- The struts-config.xml configuration file is a link between the View and Model components in the Web Client. Generally we don't touch this file.

Elements of configuration file

- The configuration file basically contains following main elements –

Interceptor	Description
struts-config	This is the root node of the configuration file.
form-beans	This is where you map your ActionForm subclass to a name. You use this name as an alias for your ActionForm throughout the rest of the strutsconfig.xml file, and even on your JSP pages.
global forwards	This section maps a page on your webapp to a name. You can use this name to refer to the actual page. This avoids hardcoding URLs on your web pages.
action-mappings	This is where you declare form handlers and they are also known as action mappings.
controller	This section configures Struts internals and rarely used in practical situations.
plug-in	This section tells Struts where to find your properties files, which contain prompts and error messages

The Struts.properties File

- It allows us to change the default behavior of the framework.
- Majorly all properties of struts.properties configuration file can also be configured in the web.xml using the init-param, also using the constant tag in the struts.xml configuration file. If we like to keep it separate and more struts specific, then we can create this file under the folder WEB-INF/classes.
- The values configured in this file will override the default values configured in default.properties which is contained in the struts2-core-x.y.z.jar distribution.
- Few properties can be changed using struts.properties file –

```
struts.devMode = true
struts.i18n.reload = true
struts.configuration.xml.reload = true
struts.url.http.port = 8080
```

Syllabus Topic : Actions

6.7 Actions

Q. Explain the role of actions in struts.

- Actions are the core of the Struts2 framework where each URL is mapped to a specific action. It provides the processing logic necessary to service the request from the user.
- Action class contains the code specifications to implement the business logic for the specific HTTP request.
- Action helps the framework to determine which result needs to render the view that can be returned.

Q. Action Mapping

- Action mapping are used in the configuration file (struts.xml) that allow the framework to select the appropriate Action class.
 - Input values from the request are transferred to the properties of the Action class.
 - Framework calls the default method of the Action to invoke the business logic.
- Consider the action mapping for email validation application

```
<action-mappings>
<action name="bean1" path="/login" scope="request"
type="com.myapp.struts.LoginAction" validate="false">
<forward name="success" path="/success.jsp"/>
<forward name="failure" path="/failure.jsp"/>
</action>
<action path="/Welcome" forward="/welcomeStruts.jsp"/>
</action-mappings>
```

- If execute() method of Action class returns "success" then success.jsp view is returned to the browser.
- If execute() method of Action class returns "failure" then failure.jsp view is returned to the browser.

Q. Action Classes

- Action classes act as the controller in the MVC pattern. Action classes respond to a user action, execute business logic and return a result that tells Struts what view to render.
- Struts 2 Action classes usually extend the ActionSupport class.
- ActionSupport provides default implementations for execute and input methods and also implements useful Struts 2 interfaces.

Role of Action

(A) Perform as a Model and determines single/multiple Results.

- Action act as a Model by encapsulating the business logic within the execute() method.
- Action determines the Result that will render the View to be returned as response.
- It returns the control string that selects the appropriate result as configured.
- **Foe Example :** execute() method of email validation application

```
public ActionForward execute(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response) throws Exception
{
    bean1 b1=(bean1)form;
    String m=b1.getEmail();
    if (m == null || m.equals("") || m.indexOf("@") == -1)
    {
        return mapping.findForward(FAILURE);
    }
    else
    {
        return mapping.findForward(SUCCESS);
    }
}
```

(B) Serve as Data Carrier

- Action serves as data carrier from request to the view, where data get stored as JavaBean properties. It allows accessing the data locally during the execution of business logic.
- Java bean code specifications to store the username and email from HTTP request:

```
private String username;
private String email;
public String getEmail()
{
    return email;
}
public void setEmail(String email)
{
```

```
this.email = email;

}

public String getUsername()
{
    return username;
}

public void setUsername(String username)
{
    this.username = username;
}
```

Syllabus Topic : Interceptors

6.8 Interceptors

- a. Explain the role of interceptors in struts.
- b. Enlist and explain major struts 2 framework interceptor.

- Interceptors allow executing separate functionality other than the action element of strut application. Interceptors can be used to :

- i. Provide pre-processing logic before the action is called.
- ii. Provide post-processing logic after the action is called.
- iii. Catching exceptions so that alternate processing can be performed.

* Interceptors are used for the following requirements :

- It can be used to implement the exception handling, file uploading, lifecycle call-backs and validation etc.
- Interceptors can execute code before and after an Action is invoked. Strut framework's core functionalities like security checking, bottleneck, tracing and logging are implemented as Interceptors.
- Interceptor is pluggable; to decide exactly which features an Action needs to support.
- Custom Interceptors can be mixed-and-matched with the framework Interceptors.
- Interceptors have access to the environmental variables and execution properties.

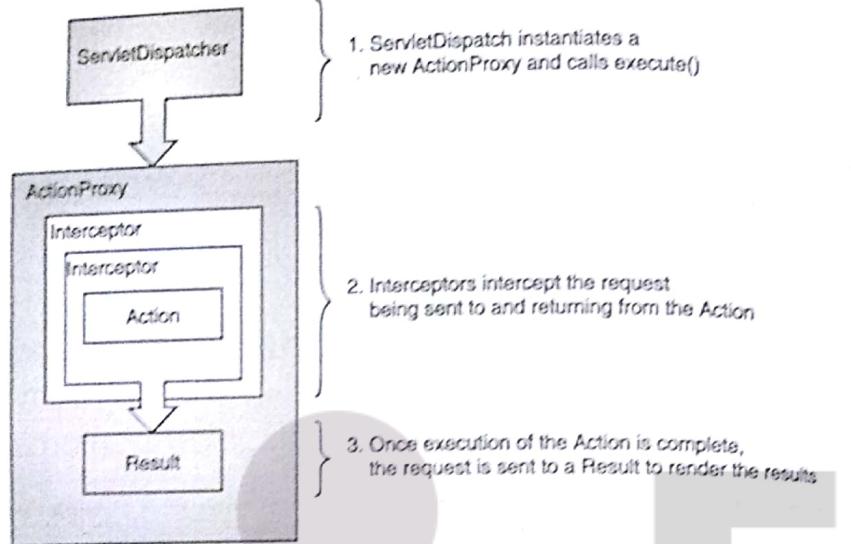


Fig. 6.8.1 : Format of Interceptors in strut applications

* Major Struts 2 Framework Interceptors

1. alias : Allows parameters to have different name aliases across requests.
2. createSession : Automatically creates an HTTP session if it does not exist.
3. exception : Allowing automatic exception handling via redirection.
4. fileUpload : Facilitates easy file uploading.
5. validation : Provides validation support for actions.

Syllabus Topic : Result and Result Types

6.9 Results and Result Types

Q. What is result and result type in struts? Explain it.

In strut framework when an Action class method completes, the response need to render back to the client. Action class method returns String values which are used to select a result element.

6.9.1 Results

- Action mapping is used to describe the set of Results representing the different possible outcomes after the Action execution.
- A set of Results are defined by the ActionSupport base class.

The result element basically handles two things :

- (i) It provides a logical name - Like an Action can return either "success" or "error" without knowing the implementation details.
 - (ii) It provides a Result Type - Results can be forward to a server page or template
 - A view in the Struts2 MVC framework displays the response using the <results> tag after executing the action business logic.
 - The Action class method returns a string after executing the business process. The value of the string can be used to select the Result view, using the <result> tag configuration
- * Example : Let execute() method returns either the string "success" or "error" leads to the Result HelloWorld.jsp or NoEntry.jsp. Then the configuration is :

```
<action name="hello" class="mystrut.HelloWorldAction" method="execute">
<result name="success"> /HelloWorld.jsp </result>
<result name="error"> /NoEntry.jsp </result>
<action>
```

JSP file can be written using a <param name="location"> tag within the <result> element as :

```
<result name="success" type="dispatcher">
<param name="location">/HelloWorld.jsp
</param>
</result>
```

* View Technologies

JSP (JavaServer Pages) are the most common way of rendering the Results. In addition Strut2 support different view technologies as :

- (i) Velocity Templates
- (ii) Freemarker Templates
- (iii) XSLT Transformations

6.9.2 Result Type

Q. What is role of result type? Discuss it with suitable example.

Result type provides the implementation details for the type of information that is returned to the user.

(i) FreeMarker Result type

Freemarker is a templating engine that generates the output using predefined templates. Let HelloWorld.fm is a Freemarker template file need to render after execute() method returns the string "success".

Then the configuration is :

```
<result name="success" type="freemarker">
<param name="location">/HelloWorld.fm</param>
</result>
```

(ii) redirect result type

It calls the response.sendRedirect() method, which indeed causes the browser to create a new request to the given URL.

URL can be written either in the <result...> element or in <param name="location"> element.

Example

```
<result name="success" type="redirect">
<param name="location">
    /HelloWorld.jsp
</param >
</result>
```

Syllabus Topic : The Value Stack

6.10 The Value Stack

Q. What is value stack in struts? State and explain the execution flow of value stack.

Q. What are the main value stack object? Discuss them in detail.

Q. List and explain different value stack methods to manipulate object.

- The value stack is a set of several objects stored in a specific order that holds all the data associated with the processing of a request in Strut framework based applications.
- Before an Action execute() method is called for business processing, framework moves all the data to the value Stack for request processing.
- Framework manipulates the value stack data during Action execution, render the result and send the response page.

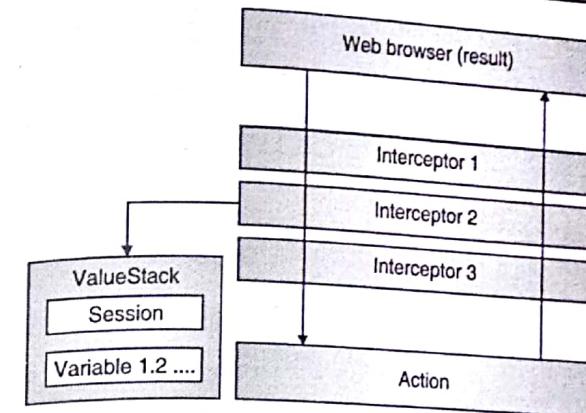


Fig. 6.10.1 : Struts 2 ValueStack Component

Four ValueStack Objects

(i) Temporary Objects : Temporary objects are created during the execution of a page. Example - the current iteration value, looped over a JSP tag etc.

(ii) Model Object : If Model object is used in a strut application, the current model object is placed on the value stack before the action execution.

(iii) The Action Object : It is the current action object being executed.

(iv) Named Objects : It includes application, session, request, attributes and parameters and refers to the corresponding servlet scopes.

Accessing the Value Stack

- The value stack can be accessed via tags provided for JSP, Velocity or Freemarker.

- ValueStack object can be accessed inside the Action as :

```
ActionContext.getContext().getValueStack()
```

ValueStack Methods to manipulate the object

1. **Object findValue(String expr) :** It finds a value by evaluating the given expression against the stack
2. **void setValue(String expr, Object value) :** it sets a property on a bean in the stack with the given expression
3. **Object peek() :** it gives the object on the top of the stack without changing the stack.
4. **Object pop() :** it gives the object on the top of the stack and remove it from the stack.
5. **void push(Object o) :** Put the object onto the top of the stack.
6. **int size() :** it gives the number of objects in the stack.

Syllabus Topic : Object-Graph Navigation Language (OGNL)

6.11 Object-Graph Navigation Language (OGNL)

- Q.** What is OGNL? What does it consist of? How its contents can be accessed?
Q. Write a short note on OGNL.

- Object-Graph Navigation Language (OGNL) is mainly used for data transfer and type conversion in struts applications.
- It is an expression language used to refer and manipulate data on the ValueStack.

☞ OGNL supports

- (i) Type conversion
 - (ii) Calling methods
 - (iii) Collection manipulation
 - (iv) Expression evaluation
- Searching or evaluating a particular expression in the Value Stack can be done using OGNL.
 - It is based on the rule of having a root or default object within the context. The properties of the root object can be referenced using the mark-up notation (#).
 - OGNL allows navigating object graphs using a dot notation and evaluate expression.
 - OGNL can be used to access the ActionContext map that consists of :
 - (i) Application - application scoped variables
 - (ii) Session - session scoped variables
 - (iii) Root / value stack - All the action variables
 - (iv) Request - Request scoped variables
 - (v) Parameters - Request parameters
 - (vi) Attributes - The attributes stored in page, request, session and application scope - Objects in the ActionContext are referred using the pound symbol.

☞ OGNL Execution flow

- (i) Struts framework receives the client request name and value in the form of string.
- (ii) OGNL expression scans the value stack to locate the destination property where value needs to be saved.
- (iii) OGNL performs the type conversion of the request data and move the value to the property by invoking the corresponding Setter method.
- (iv) Results access the value stack using OGNL expression tags to render the response.

- (v) While rendering the view, value is converted from java type to string type.
 - Struts 2 places application, session, action, request and parameters on the value stack that can be accessed using OGNL tags as :

Name	Value
#application['xyz'] or #application.xyz	ServletContext attributes 'xyz'
#session['xyz'] or #session.xyz	session attribute 'xyz'
#action['xyz'] or #action.xyz	current action getter (getXyz())
#request['xyz'] or #request.xyz	request attribute ['xyz'] (request.getAttribute())
#parameters['xyz'] or #parameters.xyz	request parameter ['xyz'] (request.getParameter())
#attr['xyz'] or #attr.xyz	Access to PageContext if available

- Objects in the value stack can be referenced directly. If 'name' is a property of an action class, it can be referenced as :

```
<s:property value="name"/>
```

- If 'login' is an attribute in session, it can be retrieved as :

```
<s:property value="#session.login"/>
```

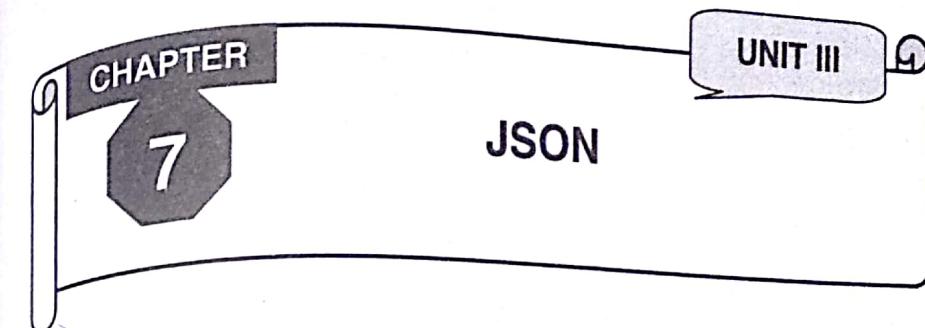
- OGNL supports collections known as Map, List and Set. Example to display a dropdown list of colors.

```
<s:select name="color" list="{red,yellow,green}" />
```

Review Questions

- Q. 1 Explain the role of actions in struts.
- Q. 2 What is value stack in struts? State and explain the execution flow of value stack.
- Q. 3 Explain struts framework.
- Q. 4 Explain the features/advantages of struts 2.0.
- Q. 5 Enlist and explain the filter dispatcher, one of the core components of struts.
- Q. 6 Explain the role of actions in struts.

- Q. 7 Explain the role of interceptors in struts.
- Q. 8 What is value stack in struts? State and explain the execution flow of value stack.
- Q. 9 What is OGNL? What does it consist of? How its contents can be accessed?
- Q. 10 Explain the significance of struts.xml and web.xml file in Strut application.
- Q. 11 Explain the application flow in MVC with respect to struts.
- Q. 12 Explain the role of actions in struts.
- Q. 13 Explain the role of interceptors in struts.
- Q. 14 Explain the role of actions in struts.
- Q. 15 What is value stack in struts? State and explain the execution flow of value stack.
- Q. 16 Explain the application flow in MVC with respect to struts.
- Q. 17 Explain the role of interceptors in struts.
- Q. 18 What is value stack in struts? State and explain the execution flow of value stack.
- Q. 19 What is OGNL? What does it consist of? How its contents can be accessed?

**Syllabus Topics**

JSON : Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java

Syllabus Topic : Overview of JSON

7.1 Overview

- Q.** What is JSON?
Q. Explain the concept of JSON in detail.
Q. Why to use JSON? Explain it in detail.

☞ **JSON :** JSON stands for JavaScript Object Notation. Basically it is syntax for storing and exchanging data. JSON is text, written with JavaScript object notation. JSON is "self-describing" and easy to understand. It is also considered as a lightweight data-interchange format.

- JSON uses JavaScript syntax, but the JSON format is text only that's why JSON is language independent. Text can be read and used as a data format by any programming language.

☞ Exchanging Data

- Data exchange between a browser and a server, is always in text format. We can convert any JavaScript object into JSON, and send JSON to the server.
- We can also convert any JSON received from the server into JavaScript objects. This how simply we can work with the data as JavaScript objects, with no complications.