**Struts Short Notes 1.x:-**

**Struts Flow:-**

[Struts 1.x Flow - in 10 Simple Steps](http://www.shareinfo.co.in/2013/05/struts-1x-flow-in-10-simple-steps.html)

Hi All,   
  
Here are the simple 10 steps for how Struts Framework will work - It is a general question in many Interviews!   
  
Initially let us assume /login.do is our action - the result will be successful login.  Here I will explain 15 steps how/what Struts framework will do ?   
  
**1)**As end user triggerred / clicked on /login.do - > as usual it is a web application - first it will look for Web.xml   
 So here - **Web.xml loading is the first step!**  
  
**2)** In Web.xml it looks for **\*.do mapping and it finds the Struts plug**-in  i.e. ActionServelt and it loads the same.   
  
*Example of Mapping:*  
*<servlet-name>action</servlet-name>*  
*<servlet-class>org.apache.struts.action.ActionServlet</servlet-class>*  
*.....*  
 *<servlet-mapping>*  
*<servlet-name>action</servlet-name>*  
*<url-pattern>\*.do</url-pattern>*  
*</servlet-mapping>*  
  
**3)** As part of loading **ActionServlet calls the init()**as every other servlet does.  
*<init-param>*  
*<param-name>config</param-name>*  
***<param-value>/WEB-INF/struts-config.xml</param-value>***  
*</init-param>*  
**So it loads the struts-config.xml**- *Same here you can configure multiple strtus config xml files*  
*(Note: ActionServlet extends HttpServlet only)*  
  
**4)** Then the **ActionServlet calls the process()**method of**RequestProcessor**class.  
  
**5)** **process() method checks for the appropriate action** in the action mapping for current request and maps the URI to the action in the struts-config.xml.  Then it **creates an object to ActionForm**associated to that action.  
  
**6)** It then**calls the setters and reset() methods** in the form bean or POJO or Java Bean class (which extends ActionForm) corresponds to the mapped action. *(Or else you can avoid this class by writing DynaActionForm - i.e. you can create beans in config xml itself)*  
  
**7)** If the action tag in **struts-config.xml contains validate "true"**then the **validate method**will call. So that you can check for normal validations like string length or mismatch of strings/ passwords etc. If any validation messages exist - add them to **MessageCollection**and iterate them in your front end jsp page. If not it will redirect to the Action class - to perform actual Business Logic.   
  
**8)**Inside Action class - execute() method performs the business logic by calling internal methods/DAO calls*(you can use Hibernate whatever - as this is not related to Struts concept)*which you provide and based on these method calls it returns with an **ActionForward**string key. *(of course you have to set the result in****session.setAttribute****to get these result in resulting jsp)*  
  
**9)** Now the returned string key**will search for corresponding forward jsp***(Inside <action> tag using <forward> tag you can mention the forward names)*  
  
**10)** It looks for the appropriate view component and performs the getter on that action form corresponding to this view component and **loads the view page in the browser.***(Iterate the result from session.getAttribute which you have set while you returned from the DAO calls)!*  
 *Here is the Sample Action Tags for your reference!*  
 *<action path="/loginEntry" type="com.sharath.frontend.action.LoginAction"*  
 *name="loginForm" parameter="method" validate="true">*  
 *<forward name="success" path="mainEntryPage" />*  
 *<forward name="loginEntry" path="loginEntryPoint" />*  
 *<forward name="registerEntry" path="registerEntryPoint" />*  
*</action>*  
 *<form-bean name="loginForm" type="com.sharath.frontend.LoginForm" />*

# **Struts1 Framework Architecture**

*by*[ADMIN](http://www.javaguru.biz/author/admin/)*on*May 12, 2013*•*4:36 pm[No Comments](http://www.javaguru.biz/struts1-framework-architecture/#comments)

**Overview of Struts Framework Architecture**

Struts is an open source framework that extends the Java Servlet API and employs a [Model, View, Controller](http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html)(MVC) architecture. It enables you to create maintainable, extensible, and flexible web applications based on standard technologies, such as JSP pages, JavaBeans, resource bundles and XML.  
Strutsis an open source framework from Apache Software Foundation for developing well-architected [Web](http://www.webopedia.com/TERM/S/World_Wide_Web.html)applications. Based on the Model-View-Controller (MVC 2) design.

**Model View Controller (MVC) 2 Architecture:**

MVC 2 Architecture and its derivatives are the cornerstones for all serious and industrial strength web applications designed in the real world. Hence it is essential for you understand this paradigm thoroughly. The following figure shows the MVC 2 Architecture.

The main difference between MVC 1 and MVC 2 is that in MVCl 2, a controller handles the user request instead of another JSP. The controller is implemented as a Servlet. The following steps are executed when the usersubmits the request.

1. The Controller Servlet handles the user’s request. (This means the hyperlink in the JSP should point to the controller servlet).

2. The Controller Servlet then instantiates appropriate JavaBeans based on the request parameters (and optionally also based on session attributes).

3. The Controller Servlet then by itself or through a controller helpercommunicates with the middle tier or directly to the database to fetch therequired data.

4. The Controller sets the resultant JavaBeans (either same or a new one) in one of the following contexts – request, session or application.

5. The controller then dispatches the request to the next view based on the request URL.

6. The View uses the resultant JavaBeans from Step 4 to display data.  
Note that there is no presentation logic in the JSP. The sole function of the JSP in MVC 2 architecture is to display the data from the JavaBeans set in the request, session or application scopes.

**Model:**

Model is responsible for providing the data from the database and saving the data into the data store.

All the business logics are implemented in the Model.

Data entered by the user through View are checked in the model before saving into the database.

Data access, Data validation and the data saving logic are part of Model.

**View:**

View represents the user view of the application and is responsible for taking the input from the user.

Dispatching the request to the servletcontroller and then receiving response from the controller to display the result to the user.

HTML, JSP files are the part of view component.

**Controller:**

Controller acts as a bridge between Model and View.

Controller is responsible for receiving the request from client.

Once request is received from client it executes the appropriate business logic from the Model and then produce the output to the user using the View component.

**Struts1 Framework Architecture:**  
Since this is your first look at Struts, now, let us concentrate on the basics of Struts Framework.

**ActionServlet class:**

**Action Servlet is a built in classes provided by struts F/W.**

**Action Servlet is a back bone of struts app’s**

**Actionservlet is a controller component that handle the all the client request**

**It is extended from org.apache.struts.action.actionServlet is called actionServlet.**

In Struts, there is only one controller servlet for the entire web application. This controller servlet is called ActionServlet and resides in the package **org.apache.struts.action**. And, struts ‘controller servlet’ which extends the**javax.servlet.http.HttpServlet** class.

The framework provides you with a controller servlet (ActionServlet) which is defined in the Struts libraries  
that are included in the IDE and it is configured as Servlet in the web.xml file.

The ActionServlet then instantiates a Handler. The Handler class name is obtained from an XML file based on the URL path information. This XML file is referred to as Struts configuration file and by default named as **struts-config.xml**.

The Handler is called **Action**in the Struts terminology. This class is created by extending the Action class in **org.apache.struts.action.Action** package.

The **Action** class is abstract and defines a single method called **execute()**. You override this method in your

own Actions and invoke the business logic in this method. The **execute()** method returns the name of next view (JSP) to be shown to the user.

The controller servlet uses a struts-config.xml file to map incoming requests to Struts Action objects and instantiate any ActionForm objects.

**ActionForm class:**

A Struts **ActionForm** is used to persist data between requests. (ActionForm is a normal JavaBeans class.). It has several attributes corresponding to the HTTP request parameters and getter, setter methods for those attributes.

You have to create your own **ActionForm**for every HTTP request handled through the Struts framework by extending the **org.apache.struts.action.ActionForm** class.

Consider the following HTTP request for App1web application –http://localhost:8080/App1/create.do?firstName=John&lastName=Doe.

The **ActionForm** class for this HTTP request is shown below. The class LoginForm extends the**org.apache.struts.action.ActionForm** class and contains three attributes – name and email. It also has getter and setter methods for these attributes.

package com.myapp.struts;

import javax.servlet.http.HttpServletRequest;  
import org.apache.struts.action.ActionErrors;  
import org.apache.struts.action.ActionMapping;  
import org.apache.struts.action.ActionMessage;

public class LoginForm extends org.apache.struts.action.ActionForm {

private String name;  
private String email;

public String getEmail() {  
return email;  
}

public void setEmail(String email) {  
this.email = email;  
}

public String getName() {  
return name;  
}

public void setName(String string) {  
name = string;  
}

public LoginForm() {  
super();  
}

public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {

ActionErrors errors = new ActionErrors();

if (getName() == null || getName().length() < 1) {

errors.add(“name”, new ActionMessage(“error.name.required”));

// TODO: add ‘error.name.required’ key to your resources  
}  
return errors;  
}  
}  
 **Action class:**

**Struts guys are developed some pre-defined action classes . all these action classes are available in org.apache.struts.action** package.

**. all these action classes are available in struts-extra.jar file**.

This class is created by extending the Action class in **org.apache.struts.action** package. The Action object processes the data stored in the form bean using its execute method. Once the Action object processes a request, it forwards the results to the appropriate view.

Action class acts as wrapper around the business logic and provides an inteface to the application’s Model layer. Action Class process the specified HTTP request, and create the corresponding HTTP response.  
To use the Action, we need a  Subclass of **org.apache.struts.action.Action** and overwrite the **execute()** method

The return type of the execute method is **ActionForward** which is used by the Struts Framework to forward  
the request to the file.

**Example of Action Class:**

package com.myapp.struts;

import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import org.apache.struts.action.Action;  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;

public class LoginAction extends Action {

public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request,                                                                HttpServletResponse response) throws Exception    {

ActionForward nextPage = null;  
LoginForm logForm = (LoginForm) form;

String firstName = logForm.getFirstName();  
String mail = logForm.getEmail();

nextPage = mapping.findForward(“success”);  
return nextPage;  
}  
}

**Parameters of execute method Action:**

mapping – The ActionMapping used to select this instance  
form – The optional ActionForm bean for this request (if any)  
request – The HTTP request we are processing  
response – The HTTP response we are creating

**Types of action class in struts framework?**

There are 9 action classes are there  
1.action  
2.dispatch action  
3.mapping dispatch action  
4.lookup dispatch action  
5.include action  
6.forward action  
7.switch action  
8.locate action  
9.download action

**2. Dispatch action**

Dispatch Action is an abstract classes extends actions. In this class doesn’t contain any abstract methods

While developing any app’s if you want to implement multiple logic in a single action class then we can implements all the logic in side the execute method

In case of dispatch action class we can write multiple method in a single action class where each method follows signature of execute method like add(),update(),delete()

* **We can write multiple business logic in single action class**
* **Each business logic follows signature of execute method.**

Dispatch Action provides a mechanism for grouping a set of related functions into a single action, thus eliminating the need to create separate actions for each functions. In this example we will see how to group a set of user related actions like add user, update user and delete user into a single action called UserAction.

The class UserAction extends org.apache.struts.actions.DispatchAction. This class does not provide an implementation of the execute() method as the normal Action class does.

The Dispatch Action uses the execute method to manage delegating the request to the individual methods based on the incoming request parameter. For example if the incoming parameter is "method=add", then the add method will be invoked. These methods should have similar signature as the execute method.

public ActionForward add(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response)

throws Exception {

UserForm userForm = (UserForm) form;

userForm.setMessage("Inside add user method.");

return mapping.findForward(SUCCESS);

}

**2. LookupDispatchAction:**

LookupDispatchAction class is an abstract class extended form dispatch action.

This LookupDispatchAction contains there is a single abstract method called “getKeyMethodMap()” so we can extended our actionclass from getkeymethodmap.

* It support I18N
* Request go to bundle take key for the value parameter

public class UserAction extends LookupDispatchAction {

private final static String SUCCESS = "success";

protected Map getKeyMethodMap() {

Map map = new HashMap();

map.put("UserForm.add", "add");

map.put("UserForm.update", "update");

map.put("UserForm.delete", "delete");

return map;

}

public ActionForward add(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {

UserForm userForm = (UserForm) form;

userForm.setMessage("Inside add user method.");

return mapping.findForward(SUCCESS);

}

Etc…

**3. Forward Action (**to forward a request to another resource in your   
application)

Forward action make sure that is forwarding the request to specific JSP through controller

It is a subclass of **org.apache.struts.action.forwardaction**

Forwardaction eliminates developing multiple blank action in the project.

**4. Include Action class**

The IncludeAction class provides a mechanism for including the contents of a specified URL. This action behaves similarly to ForwardAction, but instead of forwarding to the specified URL, the specified URL is included. This action is useful when you want to include the contents of one page in another.it is extended from org.apache.struts.actions.IncludeAction Class  
  
Using IncludeAction is quite easy. Just create action   
mapping entries in the Struts configuration file:  
  
<action-mappings>  
<action path="/menu"  
type="org.apache.struts.actions.IncludeAction"  
parameter="/menu.jsp/>  
</action-mappings>

For each page you want to include, you must create an   
action mapping. Each action mapping uses IncludeAction, but   
specifies a different path for the action. The parameter   
attribute specifies the URL that will be included when the   
specified path is accessed.

**5. Switch actions:**

The Switch Action class allows you to configure multiple application modules. SwitchAction (**org.apache.struts.actions.SwitchAction**) is a standard Action that can switch to another module and then forward control to a path within that module.

SwitchAction expects two request parameters to be passed:

* ***page***: A module-relative URI (beginning with /) to which control should be forwarded after switching.
* ***prefix***: The module prefix (beginning with /) of the application module to which control should be switched. Use a zero-length string for the default module. The appropriate ApplicationConfig object will be stored as a request attribute, so any subsequent logic will assume the new application module.

**Struts Action Forms classes:**

6 types of form beans classes

1. Action Form
2. DynaActionForm
3. Validator Form
4. ValidatorActionForm
5. DynaValidatorForm
6. DynaValidatorActionForm
7. **Action Form** -Action form for action class.

Form bean class is a sub classes of org.apache.struts.action.actionform.

In the form bean you must provide setter and getter method and

override reset() and validate () methods.

* Public void **reset**(actionmapping mapping, httpservletrequest request)
* Public actionerrror **validate** (actionmapping mapping, httpservletrequest request

1. **DynaActionForm**– No form bean is created. No Validations are done here. Form bean properties are specified in struts-config.xml file
2. **Validator Form**- Uses validation framework for validating the form bean.
3. **DynaValidatorForm**– it is asubclasses of dynaactionform and dynaactionform is available in **“org.apache.struts.validator.dynavalidatorform**” and it is also reset and validate methods . This is mainly used for it will takes of validator like client and server side validations.

form bean properties are specified in the struts-config.xml file. Validations here are done according to the to the form bean name.

1. **ValidatorActionForm** – Validations are done according to the action class name and not according to the form bean name as in ValidatorForm. Form bean is still created.
2. **DynaValidatorActionForm** –> Similar to the ValidatorActionForm but form bean is not created and form bean properties are specified in the struts-config.xml file. Validations are done according to the Action class name.

**How you will handle errors and exceptions using Struts**

struts provides a small, but effective exception-handling framework for the applications. There are two approaches available for the exception handling in struts.  
**1. Declarative:-** In this approach the exceptions are defined in the struts-config file and in case of the exception occurs the control is automatically passed to the appropriate error page. The tag is used to define the exception in the struts-config.xml file. The following are the attributes of the tag.  
1. Key:- The key defines the key present in MessageResources.properties file to describe the exception occurred.  
2. Type:- The class of the exception occurred.  
3. Path:- The page where the control is to be followed in case exception occurred.  
4. Handler:- The exception handler which will be called before passing the control to the file specified in path attribute  
There are following two sections in struts-config.xml where you can define the exceptions.  
\* With Any Action mapping:- In this approach the action class is identified which may throw the exception like password failure, resource access etc. For example:-

Code

1. **<action** path="/exception"

1. type="com.visualbuilder.ExampleAction"

1. parameter="method"

1. input="/index.jsp" **>**

1. **<exception** key="error.system" type="java.lang.RuntimeException" path="/index.jsp" **/>**

1. **</action>**



\* Defining the Exceptions Globally for the struts-config.xml :- If the application control is to pass on a single page for all similar type of exception then the exception can be defined globally. So if in any circumstance the exception occurs the control is passed globally to the exception and control is passed to the error page. For Example:-

Code

1. **<global-exceptions>**

1. **<exception** key="error.system" type="java.lang.RuntimeException" path="/index.jsp" **/>**

1. **</global-exceptions>**



**2. Programmatic: -** In this approach the exceptions are caught using normal java language try/catch block and instead of showing the exception some meaningful messages are displayed. In this approach the flow of control is also maintained by the programs. The main drawback of the approach is the developer has to write the code for the flow of the application.

# **T**[**op 50 Struts Interview Questions**](http://career.guru99.com/top-50-struts-interview-questions/)

**Q1. What are the components of Struts Framework?**

Ans: Struts framework is comprised of following components:

1. Java Servlets
2. JSP (Java [Server](http://career.guru99.com/category/server/) Pages)
3. Custom Tags
4. Message Resources

**Q2. What’s the role of a handler in MVC based applications?**

Ans:. It’s the job of handlers to transfer the requests to appropriate models as they are bound to the model layer of MVC architecture. Handlers use mapping information from configuration files for request transfer.

**Q3. What’s the flow of requests in Struts based applications?**

Ans: Struts based applications use MVC design pattern. The flow of requests is as follows:

* User interacts with View by clicking any link or by submitting any form.
* Upon user’s interaction, the request is passed towards the controller.
* Controller is responsible for passing the request to appropriate action.
* Action is responsible for calling a function in Model which has all business logic implemented.
* Response from the model layer is received back by the action which then passes it towards the view where user is able to see the response.

**Q4.  Which file is used by controller to get mapping information for request routing?**

Ans: Controller uses a configuration file “struts-config.xml file to get all mapping information to decide which action to use for routing of user’s request.

**Q5.  What’s the role of Action Class in Struts?**

Ans: In Struts, Action Class acts as a controller and performs following key tasks:

* After receiving user request, it processes the user’s request.
* Uses appropriate model and pulls data from model (if required).
* Selects proper view to show the response to the user.

**Q6. How an actionForm bean is created?**

**Surrogate**

Ans: actionForm bean is created by extending the class org.apache.struts.action.ActionForm

In the following example we have created an actionForm bean with the name 'testForm':



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | import javax.servlet.http.HttpServletRequest;  import org.apache.struts.action.\*;  public class testForm extends ActionForm {  private String Id=null;  private String State=null;  public void setId(String id){  this.Id=id;  }  public String getId(){  return this.Id;  }  public void setState(String state){  this.State=state;  }  public String getState(){  return this.State;  } |

**Q7. What are the two types of validations supported by Validator FrameWork?**

Ans: Validator Framework is used for form data validation. This framework provides two types of validations:

1. Client Side validation on user’s browser
2. Server side validation

**Q8. What are the steps of Struts Installation?**

Ans: In order to use Struts framework, we only need to add Struts.Jar file in our development environment. Once jar file is available in the CLASSPATH, we can use the framework and develop Strut based applications.

**Q9. How client side validation is enabled on a JSP form?**

Ans: In order to enable client side validation in Struts, first we need to enable validator plug-in in struts-config.xml file. This is done by adding following configuration entries in this file:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <!--  Validator plugin -->  <plug-in className="org.apache.struts.validator.ValidatorPlugIn">  <set-property  property="pathnames"  value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>  </plug-in> |

Then Validation rules are defined in validation.xml file. If a form contains email field and we want to enable client side validation for this field, following code is added in validation.xml file:



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <form name="testForm">  <field  property="email"  depends="required">  <arg key="testForm.email"/>  </field>  </form> |

**Q10. How action-mapping tag is used for request forwarding in Struts configuration file?**

Ans: In Struts configuration file (struts-config.xml), forwarding options are defined under action-mapping tag.

In the following example, when a user will click on the hyperlink **test.do**, request will be forwarded to**/pages/[testing](http://career.guru99.com/category/testing/" \o "testing).jsp**using following configurations from struts-config.xml file:



|  |  |
| --- | --- |
| 1 | <action  path="/test" forward="/pages/testing.jsp"> |

This forwarding will take place when user will click on following hyperlink on the jsp page:



|  |  |
| --- | --- |
| 1 | <html:link</strong> page="/test.do</strong>">Controller Example</html:link> |

**Q11. How duplicate form submission can be controlled in Struts?**

Ans: In Struts, action class provides two important methods which can be used to avoid duplicate form submissions.

saveToken() method of action class generates a unique token and saves it in the user’s session. isTokenValid() method is used then used to check uniqueness of tokens.

**Q12. In Struts, how can we access Java beans and their properties?**

Ans: Bean Tag Library is a Struts library which can be used for accessing Java beans.

**Q13. Which configuration file is used for storing JSP configuration information in Struts?**

Ans: For JSP configuration details, Web.xml file is used.

**Q14. What’s the purpose of Execute method of action class?**

Ans: Execute method of action class is responsible for execution of business logic. If any processing is required on the user’s request, it’s performed in this method. This method returns actionForward object which routes the application to appropriate page.

In the following example, execute method will return an object of actionForward defined in struts-config.xml with the name “exampleAction”:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;    import org.apache.struts.action.Action;  import org.apache.struts.action.ActionForm;  import org.apache.struts.action.ActionForward;  import org.apache.struts.action.ActionMapping;    public class actionExample extends Action  {  public ActionForward execute(  ActionMapping mapping,  ActionForm form,  HttpServletRequest request,  HttpServletResponse response) throws Exception{  return mapping.findForward("exampleAction");  }  } |

**Q15. What’s the difference between validation.xml and validator-rules.xml files in Struts Validation framework?**

Ans: In Validation.xml, we define validation rules for any specific Java bean while in validator-rules.xml file, standard and generic validation rules are defined.

**Q16. How can we display all validation errors to user on JSP page?**

Ans: To display all validation errors based on the validation rules defined in validation.xml file, we use <html:errors /> tag in our JSP file.

**Q17. What’s declarative exception handling in Struts?**

Ans: When logic for exception handling is defined in struts-config.xml or within the action tag, it’s known as declarative exception handling in Struts.

In the following example, we have defined exception in struts-config.xml file for NullPointerException:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <global-exceptions>    <exception key="test.key"    Type="java.lang.NullPointerException"    Path="/WEB-INF/errors/error\_page.jsp"    </global-exceptions> |

**Q18. What’s DynaActionForm?**

Ans: DynaActionForm is a special type of actionForm class (sub-class of ActionForm Class) that’s used for dynamically creating form beans. It uses configuration files for form bean creation.

**Q19. What configuration changes are required to use Tiles in Struts?**

Ans: To create reusable components with Tiles framework, we need to add following plugin definition code in struts-config.xml file:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <plug-in className="org.apache.struts.tiles.TilesPlugin" >    <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />    <set-property property="moduleAware" value="true" />    </plug-in> |

**Q20. What’s the difference between Jakarta Struts and Apache Struts? Which one is better to use?**

Ans: Both are same and there is no difference between them.

**Q21. What’s the use of Struts.xml configuration file?**

Ans: Struts.xml file is one the key configuration files of Struts framework which is used to define mapping between URL and action. When a user’s request is received by the controller, controller uses mapping information from this file to select appropriate action class.

**Q22. How tag libraries are defined in Struts?**

Ans: Tag libraries are defined in the configuration file (web.xml) inside <taglib> tag as follows:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <taglib>    <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>    </taglib> |

**Q23. What’s the significance of logic tags in Struts?**

Ans: Use of logic tags in Struts helps in writing a clean and efficient code at presentation layer without use of scriptlets.

**Q24. What are the two scope types for formbeans?**

Ans: 1. Request Scope: Formbean values are available in the current request only

[](http://click.linksynergy.com/fs-bin/click?id=bt30QTxEyjA&offerid=323058.45&subid=0&type=4)

2. Session Scope: Formbean values are available for all requests in the current session.

**Q25. How can we group related actions in one group in Struts?**

Ans: To group multiple related actions in one group, we can use DispatcherAction class.

**Q26. When should we use SwtichAction?**

Ans: The best scenario to use SwitchAction class is when we have a modular application with multiple modules working separately. Using SwitchAction class we can switch from a resource in one module to another resource in some different module of the application.  
**Q27. What are the benefits of Struts framework?**

Ans: Struts is based on MVC and hence there is a good separation of different layers in Struts which makes Struts applications development and customization easy. Use of different configuration files makes Struts applications easily configurable. Also, Struts is open source and hence, cost effective.

**Q28. What steps are required to for an application migration from Struts1 to Struts2?**

Ans: Following Steps are required for Struts1 to Struts2 migration:

1. Move Struts1 actionForm to Struts2 POJO.
2. Convert Struts1 configuration file (struts-config.xml) to Struts2 configuration file (struts.xml)

**Q29. How properties of a form are validated in Struts?**

Ans: For validation of populated properties, validate() method of ActionForm class is used before handling the control of formbean to Action class.

**Q30. What’s the use of reset method of ActionForm class?**

Ans: reset method of actionForm class is used to clear the values of a form before initiation of a new request.

**Q31. What are disadvantages of Struts?**

Ans: Although Struts have large number of advantages associated, it also requires bigger learning curve and also reduces transparency in the development process.

Struts also lack proper documentation and for many of its components, users are unable to get proper online resources for help.

**Q32. What’s the use of resourcebundle.properties file in Struts Validation framework?**

Ans: resourcebundle.properties file is used to define specific error messages in key value pairs for any possible errors that may occur in the code.

This approach helps to keep the code clean as developer doesn’t need to embed all error messages inside code.

**Q33. Can I have html form property without associated getter and setter formbean methods?**

Ans: For each html form property, getter and setter methods in the formbean must be defined otherwise application results in an error.

**Q34. How many servlet controllers are used in a Struts Application?**

Ans: Struts framework works on the concept of centralized control approach and the whole application is controlled by a single servlet controller. Hence, we require only one servlet controller in a servlet application.

**Q35. For a single Struts application, can we have multiple struts-config.xml files?**

Ans: We can have any number of Struts-config.xml files for a single application.

We need following configurations for this:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <servlet>    <servlet-name>action</servlet-name>    <servlet-class>    org.apache.struts.action.ActionServlet    </servlet-class>    <init-param>    <param-name>config</param-name>    <param-value>    /WEB-INF/struts-config.xml    /WEB-INF/struts-config\_user.xml    /WEB-INF/struts-config\_admin.xml    </param-value>    </init-param>    .............    .............    </servlet> |

**Q36. Which model components are supported by Struts?**

Ans: Struts support all types of models including Java beans, EJB, CORBA. However, Struts doesn’t have any in-built support for any specific model and it’s the developer’s choice to opt for any model.

**Q37. When it’s useful to use IncludeAction?**

Ans: IncludeAction is action class provided by Struts which is useful when an integration is required between Struts and Servlet based application.

**Q38. Is Struts thread safe?**

Ans: Yes Struts are thread safe. In Struts, a new servlet object is not required to handle each request; rather a new thread of action class object is used for each new request.

**Q39. What configuration changes are required to use resource files in Struts?**

Ans:  Resource files (.properties files) can be used in Struts by adding following configuration entry in struts-config.xml file:

<message-resources parameter=”com.login.struts.ApplicationResources”/>

**Q40. How nested beans can be used in Struts applications?**

Ans: Struts provide a separate tag library (Nested Tag Library) for this purpose. Using this library, we can nest the beans in any Struts based application.

**Q41. What are the Core classes of Struts Framework?**

Ans: Following are the core classes provided by Struts Framework:

* Action Class
* ActionForm Class
* ActionMapping Class
* ActionForward Class
* ActionServlet Class

**Q42. Can we handle exceptions in Struts programmatically?**

Ans: Yes we can handle exceptions in Struts programmatically by using try, catch blocks in the code.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | try {    // Struts code    }    Catch (Exception e) {    // exception handling code    } |

**Q43. Is Struts Framework part of J2EE?**

Ans: Although Struts framework is based on J2EE technologies like JSP, Java Beans, Servlets etc but it’s not a part of J2EE standards.

**Q44.  How action mapping is configured in Struts?**

Ans: Action mappings are configured in the configuration file struts-config.xml under the tag <action-mapping> as follows:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <pre><action-mappings>  <action path="/login"  type="login.loginAction"  name="loginForm"  input="/login.jsp"  scope="request"  validate="true">  <forward name="success" path="/index.jsp"/>  <forward name="failure" path="/login\_error.jsp"/>  </action>  </action-mappings> |

**Q45. When should be opt for Struts Framework?**

Ans: Struts should be used when any or some of the following conditions are true:

* A highly robust enterprise level application development is required.
* A reusable, highly configurable application is required.
* A loosely coupled, MVC based application is required with clear segregation of different layers.

**Q46. Why ActionServlet is singleton in Struts?**

Ans: In Struts framework, actionServlet acts as a controller and all the requests made by users are controlled by this controller. ActionServlet is based on singleton design patter as only one object needs to be created for this controller class. Multiple threads are created later for each user request.

**Q47. What are the steps required for setting up validator framework in Struts?**

Ans: Following Steps are required to setup validator framework in Struts: – **Wrong Spelling**

1. In WEB-INF directory place valdator-rules.xml and validation.xml files.
2. Enable validation plugin in struts-config.xml files by adding following:



|  |  |
| --- | --- |
| 1  2  3  4 | <plug-in className="org.apache.struts.validator.ValidatorPlugIn">  <set-property property="pathnames" value="/WEB-INF/validator-rules.xml,  /WEB-INF/validation.xml"/>  </plug-in> |

**Q48. Which technologies can be used at View Layer in Struts?**

Ans: In Struts, we can use any of the following technologies in view layer:

* JSP
* HTML
* XML/XSLT
* WML Files
* Velocity Templates
* Servlets

**Q49. What are the conditions for actionForm to work correctly?**

Ans: ActionForm must fulfill following conditions to work correctly:

* It must have a no argument constructor.
* It should have public getter and setter methods for all its properties.

**Q50.  Which library is provided by Struts for form elements like check boxes, text boxes etc?**

Ans: Struts provide HTML Tags library which can be used for adding form elements like text fields, text boxes, radio buttons etc.