1) What is Struts?

Struts is a framework for developing MVC-based framework. Struts2 is the combination of Webwork and struts1 frameworks.[More details...](http://www.javatpoint.com/struts-2-tutorial)

2) What is the difference between struts1 and struts2?

|  |  |  |
| --- | --- | --- |
| **No.** | **Struts1** | **Struts2** |
| 1) | Action class is not POJO. You need to inherit abstract class. | Action class is POJO. You don't need to inherit any class or implement any interface. |
| 2) | Front controller is ActionServlet. | Front Controller is StrutsPrepareAndExecuteFilter. |
| 3) | It uses the concept of RequestProcessor class while processing request. | It uses the concept of Interceptors while processing the request. |
| 4) | It has only JSP for the view component. | It has JSP, Freemarker, Valocity etc. for the view component. |
| 5) | Configuration file name can be [anyname].xml and placed inside WEB-INF directory. | Configuration file must be struts.xml and placed inside classes directory. |
| 6) | Action and Model are separate. | Action and Model are combined within action class. |

3) What are the features of Struts?

* Configuration MVC components
* POJO based action
* AJAX Support
* Various Tag Support
* Various Result Types
* Integration Support

[More details...](http://www.javatpoint.com/struts-2-features-tutorial)

4) What is MVC?

MVC is a design pattern. MVC stands for Model, View and Controller. Model represents data, view represents presentation and controller acts as an interface between model and view.



[More details...](http://www.javatpoint.com/model-1-and-model-2-mvc-architecture)

5) What is interceptor?

Interceptor is an object i.e. invoked at preprocessing and postprocessing of a request. It is pluggable. [More details...](http://www.javatpoint.com/struts-2-interceptors-tutorial)

6) What are the life cycle methods of interceptor?

* public void init()
* public void intercept(ActionInvocation ai)
* public void destroy()

[More details...](http://www.javatpoint.com/struts-2-custom-interceptor-example-tutorial)

7) What is ValueStack?

ValueStack is a stack that contains application specific object such as action and other model. [More details...](http://www.javatpoint.com/struts-2-ValueStack-tutorial)

8) What is ActionContext?

ActionContext is a container in which action is executed. It is unique per thread. [More details...](http://www.javatpoint.com/struts-2-ActionContext-tutorial)

9) What is ActionInvocation?

ActionInvocation is responsible to invoke action. It holds action and interceptor objects. [More details...](http://www.javatpoint.com/struts-2-ActionInvocation-tutorial)

10) What is OGNL?

OGNL is an expression language of struts2. It stands for Object Graph Navigation Language. [More details...](http://www.javatpoint.com/struts-2-OGNL-tutorial)

11) What are the 5 constants of Action interface?

1. SUCCESS
2. ERROR
3. INPUT
4. LOGIN
5. NONE

[More details...](http://www.javatpoint.com/struts-2-action)

12) What does params interceptor?

The params (also known as parameters) interceptor sets all parameters on the ValueStack. [More details...](http://www.javatpoint.com/struts-2-params-interceptor-example)

13) What does execAndWait interceptor?

The execAndWait (also known as ExecuteAndWait) interceptor is used to display intermediate or wait result. [More details...](http://www.javatpoint.com/struts-2-execandwait-interceptor-example)

14) What does modelDriven interceptor?

The modelDriven interceptor makes other model as the default object of ValueStack. By default, action is the default object of ValueStack. [More details...](http://www.javatpoint.com/struts-2-modeldriven-interceptor-example)

15) What does validation interceptor?

The validation interceptor performs validation checks and adds field-level and action-level error messages. [More details...](http://www.javatpoint.com/struts-2-validation-by-bundled-validators)

16) What are the bundled validators?

* requiredstring
* stringlength
* email
* date
* int
* double
* url
* regex

[More details...](http://www.javatpoint.com/struts-2-validation-by-bundled-validators)

17) What is the difference between plain-validator and field-validator?

In plain-validator one validator can be applied to many fields. In field-validator many validators can be applied to single field.[More details...](http://www.javatpoint.com/struts-2-validation-by-bundled-validators)

18) What is the use of jsonValidation?

The jsonValidation interceptor is used to perform asynchronous validation. It works with validation and workflow interceptors.[More details...](http://www.javatpoint.com/struts-2-ajax-validation-jsonValidation-interceptor)

19) What are the aware interfaces in struts2?

Aware interfaces are used to store information in request, session, application and response objects. The 4 aware interfaces are given below:

* ServletRequestAware
* ServletResponseAware
* SessionAware
* ServletContextAware

[More details...](http://www.javatpoint.com/struts-2-aware-interfaces-tutorial)

20) What does i18n interceptor?

The i18n interceptor is used to provide multi lingual support for struts application. [More details...](http://www.javatpoint.com/struts-2-with-i18n-example-tutorial)

**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?**

[](http://career.guru99.com/wp-content/uploads/2012/03/Struts-Interview-Questions.jpg)

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

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.

**Question 1: What is Struts? Why you have used struts in your application or project.**

[struts interview questions answers j2ee](http://javarevisited.blogspot.com/2011/09/swing-interview-questions-answers-in.html)Ans: This is the first interview questions anyone asks in Struts to get the interview rolling. Most commonly asked during the less senior level. Struts are nothing but open source framework mostly used for making web application whenever we use the term framework means it comprises JSP, servlet, custom tags message resources all in one bundle which makes developer task very easy. Its is based on MVC pattern which is model view Controller pattern.

Now **why we use Struts**? So main reason is if we go with servlet all HTML code which is related to design part mostly will come inside java code which makes code unmaintainable and complex similarly if use JSP, all java code related to business come inside design part which again make code complex, that’s why MVC pattern come into existence and which separate the business, design and controller and struts were made based on this pattern and easy to develop web application.   
  
The keyword to answer this Struts interview questions is **MVC design pattern**, Front Controller Pattern and better flow management which mostly interviewer are looking to hear. You can read more design pattern interview question on my post [10 Interview question on Singleton Pattern in Java](http://javarevisited.blogspot.com/2011/03/10-interview-questions-on-singleton.html)

**Question 2: What are the main classes which are used in struts application?**

Ans 2: This is another *beginner’s level Struts interview question* which is used to check how familiar candidate is with Struts framework and API. Main classes in Struts Framework are:

**Action servlet: it’s** a backbone of web application it’s a controller class responsible for handling the entire request.

**Action class**: using Action classes all the business logic is developed us call model of the application also.

**Action Form**: it’s a java bean which represents our forms and associated with action mapping. And it also maintains the session state its object is automatically populated on the server side with data entered from a form on the client side.

**Action Mapping**: using this class we do the mapping between object and Action.

**ActionForward**: this class in Struts is used to forward the result from controller to destination.

**Question 3: How exceptions are handled in Struts application?**

Ans: This is little tough Struts interview question though looks quite basic not every candidate knows about it. Below is my answer of this interview questions on Struts:

There are two ways of handling exception in Struts:

**Programmatically handling:** using try {} catch block in code where an exception can come and flow of code is also decided by programmer .its a normal java language concept.

**Declarative handling: There are two ways again either we define <global-Exception> tag inside struts-config.XML file**

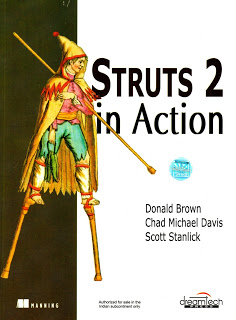
<exception  
  
      key="stockdataBase.error.invalidCurrencyType"  
  
      path="/AvailbleCurrency.jsp"  
  
      type="Stock.account.illegalCurrencyTypeException">

</exception>

The programmatic and Declarative way is sometimes also asked as follow-up questions are given candidate’s response on knowledge on Struts.

**Key:** The key represents the key present in MessageResource.properties file to describe the exception.

**Type:**The class of the exception occurred.  
**Path:** The page where the controls are to be followed is case exception occurred.  
  
See [Struts in Action](http://www.amazon.com/Struts-2-Action-Don-Brown/dp/193398807X?tag=javamysqlanta-20) for more details:

[](http://www.amazon.com/Struts-2-Action-Don-Brown/dp/193398807X?tag=javamysqlanta-20)

**Question 4: How validation is performed in struts application**?

Ans: Another classic Struts interview question it’s higher on a level than previous interview questions because it’s related to important validation concept on a web application. In struts validation is performed using validator framework, Validator Framework in Struts consist of two XML configuration files.

1. **validator-rules.xml** file: which contains the default struts pluggable validator definitions. You can add new validation rules by adding an entry in this file. This was the original beauty of struts which makes it highly configurable.

2.**Validation.xml** files which contain details regarding the validation routines that are applied to the different Form Beans.

These two configuration files in Struts should be placed somewhere inside the /WEB-INF folder of the application to keep it safe from the client and make it available in [Classpath](http://javarevisited.blogspot.com/2011/01/how-classpath-work-in-java.html).

<!--  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>

Now the next step towards validation is created error messages inside the message resource property file which are used by validator framework.

**Message resource Contain:**

1. CurrencyConverterForm.fromCurrency = From Currency

2. CurrencyConverterForm.toCurrency=To currency

3. errors.required={0} is required.

Then validation rules are defined in validation.xml for the fields of form on which we want desire validation

**Form bean code that extend DynaValidatorForm**

Eg; <form-beans>

<form-bean name="CurrencyConverterForm" type="org.apache.struts.validator.DynaValidatorForm">

<form-property name="fromCurrency" type="java.lang.double" />

<form-property name="toCurrecny" type="java.lang.double" />

</form-bean>

</form-beans>

**Validation.xml file contains**

<form-validation>

<formset>

<form name=" CurrencyConverterForm ">

<field property=" fromCurrency " depends="required">

<arg key=" CurrencyConverterForm. fromCurrency "/>

</field>

<field property=" toCurrecny " depends="required ">

<arg key=" CurrencyConverterForm.toCurrency "/>

</field>

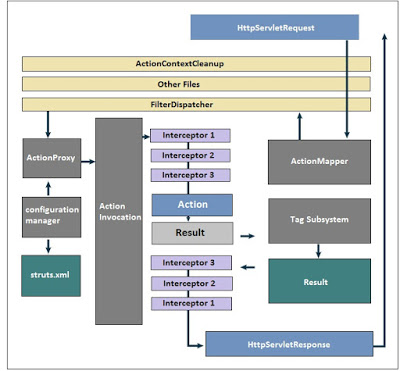
</form>

</formset>

</form-validation>

To associate more than one validation rule to the property we can specify a comma-delimited list of values. The first rule in the list will be checked first and then the next rule and so on. Answer of this Struts questions gets bit longer but it’s important to touch these important concepts to make it useful.

### Top 10 Interview Questions asked in Struts

[](https://1.bp.blogspot.com/-qg4VAEF2OYw/VurGLwezAWI/AAAAAAAAFIw/GGvhWaIFg-4YV65PTUZk6geAMN1o4yZKw/s1600/Struts%2B2%2BArchitecture.jpg)

**Question 5: What is the Difference between DispatchAction and LookupDispatchAction in Struts Framework?**

This ***Struts interview question***is pretty straight forward and I have put the differences in tabular format to make it easy to read.

|  |  |
| --- | --- |
| **Dispatch Action** | **LookupDispatchAction** |
| It’s a parent class of  LookupDispatchAction | Subclass of Dispatch Action |
| DispatchAction provides a mechanism for grouping a set of related functions into a single action, thus eliminating the need to create separate actions for each function. | An abstract **Action** that dispatches to the subclass mapped executes method. This is useful in cases where an HTML form has multiple submit buttons with the same name. The button name is specified by the parameter property of the corresponding ActionMapping. |
| If not using Internalization functionality then dispatch action is more useful. | Lookup Dispatch Action is useful when we are using Internalization functionality |
| DispatchAction selects the method to execute depending on the request parameter value which is configured in the XML file. | **LookupDispatchAction** looks into the resource bundle file and finds out the corresponding key name. We can map this key name to a method name by overriding the getKeyMethodMap() method. |
| **DispatchAction** is not useful for I18N | LookupDispatchAction is used for I18N |

**Question 6: How you can retrieve the value which is set in the JSP Page in the case of DynaActionForm?**

Ans: DynaActionForm is a popular topic in Struts interview questions. **DynaActionForm** is a subclass of ActionForm that allows the creation of form beans with dynamic sets of properties, without requiring the developer to create a Java class for each type of form bean. DynaActionForm eliminates the need of FormBean class and now the form bean definition can be written into the **struts-config.XML** file. So, it makes the FormBean declarative and this helps the programmer to reduce the development time.

For Example, we have a CurrencyConverterForm and we don't want a java class.   
CurrencyConverterForm has properties fromCurrency**, toCurrency**  
  
in the **struts-config.xml** file, declare the form bean

<form-bean name=" CurrencyConverterForm "   
type="org.apache.struts.action.DynaActionForm">   
<form-property name=" fromCurrency " type="java.lang.String"/>   
<form-property name=" toCurrency " type="java.lang. String "/>   
</form-bean>   
  
Add **action mapping in the struts-config.xml** file:   
  
<action path="/convertCurrency" type="com.techfaq.action.ConvertCurrencyAction"   
name=" CurrencyConverterForm "   
scope="request"   
validate="true"   
input="/pages/ currencyConverterform.jsp">   
  
<forward name="success" path="/jsp/success.jsp"/>   
<forward name="failure" path="/jsp/error.jsp" />   
  
</action>   
  
In the Action class.

public class ConvertCurrencyAction extends Action   
{   
public ActionForward execute(   
ActionMapping mapping,   
ActionForm form,   
HttpServletRequest request,   
HttpServletResponse response) throws Exception{   
  
DynaActionForm currencyConverterForm = (DynaActionForm)form;

// by this way we can retrieve the value which is set in the JSP Page

String fromCurrency = (String) currencyConverterForm.get("fromCurrency ");   
String toCurrency = (String) currencyConverterForm.get("toCurrency ");   
return mapping.findForward("success");   
}   
}   
}   
  
In the JSP page

<html:text property=" fromCurrency " size="30" maxlength="30"/>   
<html:text property=" toCurrency " size="30" maxlength="30"/>

**Question 7: what the Validate () and reset () method does?**

Ans: This is one of my personal favorite Struts interview questions. validate(): validate method is Used to validate properties after they have been populated, and this , method is  Called before FormBean is passed  to Action. Returns a collection of ActionError as ActionErrors. Following is the method signature for the validate() method.   
  
public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {   
                           
ActionErrors errors = new ActionErrors();   
if ( StringUtils.isNullOrEmpty(username) && StringUtils.isNullOrEmpty(password)){   
     errors.add(ActionErrors.GLOBAL\_ERROR, new ActionError("error.usernamepassword.required"));   
}   
return errors;   
}

**reset():** reset() method is called by Struts Framework with each request that uses the defined ActionForm. The purpose of this method is to reset all of the ActionForm's data members prior to the new request values being set.   
  
Example :   
public void reset(ActionMapping mapping, HttpServletRequest request) {   
this.password = null;   
this.username = null;   
}   
  
Set null for every request.

**Question 8: How you will make available any Message Resources Definitions file to the Struts Framework Environment?**

Ans: Message Resources Definitions file are simple .properties files and these files contain the messages that can be used in the struts project. Message Resources Definitions files can be added to the struts-config.xml file through < message-resources / > tag. Example: < message-resources parameter= MessageResources / >

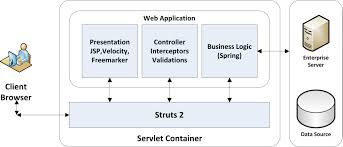
Message resource definition files can available to the struts environment in two ways   
1. using web.xml as  
<servlet>  
<servlet-name>action<servlet-name>  
servlet-class>org.apache.struts.action.ActionServlet<servlet-class>  
<init-param>  
<param-name>application<param-name>  
<param-value>resource.Application<param-value>  
</servlet>  
  
2.  
<message-resource key="myResorce" parameter="resource.Application" null="false">

**Question 9: What configuration files are used in Struts?**

**Ans:**ApplicationResources.properties and struts-config.xml these two files are used to between the Controller and the Model**.**

Question 10: Explain Struts work Flow?

Ans: **Sometime this Struts interview questions asked as first questions but I recall this now J . Here is the**answer to this Struts interview questions

[](https://2.bp.blogspot.com/-zflsNjLB5oo/VurGMhLDJ0I/AAAAAAAAFI0/XqlafRyZrNw_fK6aenTJ96_HMLe3xeswA/s1600/How%2Bto%2Buse%2BStruts2%2Bframework.jpe)

1) A request comes in from a Java Server Page into the ActionServlet.

2) The ActionServlet having already read the struts-config.xml file knows which form bean relates to this JSP, and delegates work to the validate method of that form bean.

3) The form bean performs the validate method to determine if all required fields have been entered, and performs whatever other types of field validations that need to be performed.

4) If any required field has not been entered, or any field does not pass validation, the form bean generates ActionErrors, and after checking all fields returns back to the ActionServlet.

5) The ActionServlet checks the ActionErrors that were returned from the form beans validate method to determine if any errors have occurred. If errors have occurred, it returns to the originating JSP displaying the appropriate errors.

6) If no errors occurred in the validate method of the form bean, the ActionServlet passes control to the appropriate Action class.

7) The Action class performs any necessary business logic, and then forwards to the next appropriate action (probably another JSP).

That’s all on **Struts interview question answers**, for now, there are lots many interview questions on Struts which I have not covered which you guys can contribute and I will include it here. If you are looking to find the answer to any question asked on Struts interview then please put it here and I will try to find an answer to those questions.

Read more: [http://javarevisited.blogspot.com/2011/11/struts-interview-questions-answer-j2ee.html#ixzz46tWuFJUv](http://javarevisited.blogspot.com/2011/11/struts-interview-questions-answer-j2ee.html" \l "ixzz46tWuFJUv)

**1.What is MVC?**

Model-View-Controller (MVC) is a design pattern put together to help control change. MVC decouples interface from business logic and data.

* **Model :** The model contains the core of the application's functionality. The model encapsulates the state of the application. Sometimes the only functionality it contains is state. It knows nothing about the view or controller.
* **View:** The view provides the presentation of the model. It is the *look* of the application. The view can access the model getters, but it has no knowledge of the setters. In addition, it knows nothing about the controller. The view should be notified when changes to the model occur.
* **Controller:**The controller reacts to the user input. It creates and sets the model.

## [More about Model-View-Controller Architecture >>](http://www.developersbook.com/articles/MVC-Architecture.php)

**2.What is a framework?**

A framework is made up of the set of classes which allow us to use a library in a best possible way for a specific requirement.

**3.What is Struts framework?**

Struts framework is an open-source framework for developing the web applications in Java EE, based on MVC-2 architecture. It uses and extends the Java Servlet API. Struts is robust architecture and can be used for the development of application of any size. Struts framework makes it much easier to design scalable, reliable Web applications with Java.

**4.What are the components of Struts?**

Struts components can be categorize into Model, View and Controller:

* **Model:** Components like business logic /business processes and data are the part of model.
* **View:** HTML, JSP are the view components.
* **Controller:** Action Servlet of Struts is part of Controller components which works as front controller to handle all the requests.

**5.What are the core classes of the Struts Framework?**

Struts is a set of cooperating classes, servlets, and JSP tags that make up a reusable MVC 2 design.

* JavaBeans components for managing application state and behavior.
* Event-driven development (via listeners as in traditional GUI development).
* Pages that represent MVC-style views; pages reference view roots via the JSF component tree.

**6.What is ActionServlet?**

ActionServlet is a simple servlet which is the backbone of all Struts applications. It is the main Controller component that handles client requests and determines which Action will process each received request. It serves as an Action factory – creating specific Action classes based on user’s request.

**7.What is role of ActionServlet?**

ActionServlet performs the role of Controller:

* Process user requests
* Determine what the user is trying to achieve according to the request
* Pull data from the model (if necessary) to be given to the appropriate view,
* Select the proper view to respond to the user
* Delegates most of this grunt work to Action classes
* Is responsible for initialization and clean-up of resources

**8.What is the ActionForm?**

ActionForm is javabean which represents the form inputs containing the request parameters from the View referencing the Action bean.

**9.What are the important methods of ActionForm?**

The important methods of ActionForm are : validate() & reset().

**10.Describe validate() and reset() methods ?**

***validate()*** : Used to validate properties after they have been populated; Called before FormBean is handed to Action. Returns a collection of ActionError as ActionErrors. Following is the method signature for the validate() method.

**public** ActionErrors validate(ActionMapping mapping,HttpServletRequest request)

***reset()***: reset() method is called by Struts Framework with each request that uses the defined ActionForm. The purpose of this method is to reset all of the ActionForm's data members prior to the new request values being set.

public void reset() {}

**11.What is ActionMapping?**

Action mapping contains all the deployment information for a particular Action bean. This class is to determine where the results of the Action will be sent once its processing is complete.

**12.How is the Action Mapping specified ?**

We can specify the action mapping in the configuration file called struts-config.xml. Struts framework creates ActionMappingobject from <ActionMapping> configuration element of struts-config.xml file

<action-mappings>  
 <action path="/submit"  
 type="submit.SubmitAction"  
  name="submitForm"  
  input="/submit.jsp"  
  scope="request"  
 validate="true">  
 <forward name="success" path="/success.jsp"/>  
 <forward name="failure" path="/error.jsp"/>  
 </action>  
</action-mappings>

**13.What is role of Action Class?**

An Action Class performs a role of an adapter between the contents of an incoming HTTP request and the corresponding business logic that should be executed to process this request.

**14.In which method of Action class the business logic is executed ?**

In the execute() method of Action class the business logic is executed.

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

execute() method of Action class:

* Perform the processing required to deal with this request
* Update the server-side objects (Scope variables) that will be used to create the next page of the user interface
* Return an appropriate ActionForward object

**15.What design patterns are used in Struts?**

|  |
| --- |
|  |

Struts is based on model 2 MVC (Model-View-Controller) architecture. Struts controller uses the *command design pattern* and the action classes use the *adapter design pattern*. The process() method of the RequestProcessor uses the *template method design pattern*. Struts also implement the following J2EE design patterns.

* Service to Worker
* Dispatcher View
* Composite View (Struts Tiles)
* Front Controller
* View Helper
* Synchronizer Token

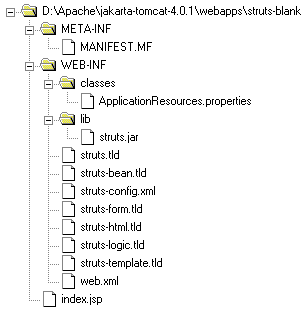
**16.Can we have more than one struts-config.xml file for a single Struts application?**

Yes, we can have more than one struts-config.xml for a single Struts application. They can be configured as follows:

<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-admin.xml,  
 /WEB-INF/struts-config-forms.xml**  </param-value></init-param>  
.....  
<servlet>

**17.What is the directory structure of Struts application?**

The directory structure of Struts application :

  
  
**18.What is the difference between session scope and request scope when saving formbean ?**

when the scope is *request*,the values of formbean would be available for the current request.  
when the scope is *session*,the values of formbean would be available throughout the session.

**19.What are the important tags of struts-config.xml ?**

The five important sections are:



**20.What are the different kinds of actions in Struts?**

The different kinds of actions in Struts are:

* ForwardAction
* IncludeAction
* DispatchAction
* LookupDispatchAction
* SwitchAction

**21.What is DispatchAction?**

The DispatchAction class is used to group related actions into one class. Using this class, you can have a method for each logical action compared than a single execute method. The DispatchAction dispatches to one of the logical actions represented by the methods. It picks a method to invoke based on an incoming request parameter. The value of the incoming parameter is the name of the method that the DispatchAction will invoke.

**22.How to use DispatchAction?**

To use the DispatchAction, follow these steps :

* Create a class that extends DispatchAction (instead of Action)
* In a new class, add a method for every function you need to perform on the service – The method has the same signature as theexecute() method of an Action class.
* Do not override execute() method – Because DispatchAction class itself provides execute() method.
* Add an entry to struts-config.xml

[**DispatchAction Example  »**](http://www.developersbook.com/struts/dispatch-action-example.php)

**23.What is the use of ForwardAction?**

The ForwardAction class is useful when you’re trying to integrate Struts into an existing application that uses Servlets to perform business logic functions. You can use this class to take advantage of the Struts controller and its functionality, without having to rewrite the existing Servlets. Use ForwardAction to forward a request to another resource in your application, such as a Servlet that already does business logic processing or even another JSP page. By using this predefined action, you don’t have to write your own Action class. You just have to set up the struts-config file properly to use ForwardAction.

**24.What is IncludeAction?**

The IncludeAction class is useful when you want to integrate Struts into an application that uses Servlets. Use the IncludeAction class to include another resource in the response to the request being processed.

**25.What is the difference between ForwardAction and IncludeAction?**

The difference is that you need to use the IncludeAction only if the action is going to be included by another action or jsp. UseForwardAction to forward a request to another resource in your application, such as a Servlet that already does business logic processing or even another JSP page.

**26.What is LookupDispatchAction?**

The LookupDispatchAction is a subclass of DispatchAction. It does a reverse lookup on the resource bundle to get the key and then gets the method whose name is associated with the key into the Resource Bundle.

**27.What is the use of LookupDispatchAction?**

LookupDispatchAction is useful if the method name in the Action is not driven by its name in the front end, but by the Locale independent key into the resource bundle. Since the key is always the same, the LookupDispatchAction shields your application from the side effects of I18N.

|  |
| --- |
|  |

**28.What is difference between LookupDispatchAction and DispatchAction?**

The difference between LookupDispatchAction and DispatchAction is that the actual method that gets called in LookupDispatchAction is based on a lookup of a key value instead of specifying the method name directly.

**29.What is SwitchAction?**

The SwitchAction class provides a means to switch from a resource in one module to another resource in a different module. SwitchAction is useful only if you have multiple modules in your Struts application. The SwitchAction class can be used as is, without extending.

**30.What if**<action>**element has**<forward>**declaration with same name as global forward?**

In this case the global forward is not used. Instead the <action> element’s <forward>takes precendence.

**31.What is DynaActionForm?**

A specialized subclass of ActionForm that allows the creation of form beans with dynamic sets of properties (configured in configuration file), without requiring the developer to create a Java class for each type of form bean.

**32.What are the steps need to use DynaActionForm?**

Using a DynaActionForm instead of a custom subclass of ActionForm is relatively straightforward. You need to make changes in two places:

* In struts-config.xml: change your <form-bean> to be an org.apache.struts.action.DynaActionForm instead of some subclass of ActionForm

<form-bean name="loginForm"type="org.apache.struts.action.DynaActionForm" >  
    <form-property name="userName" type="java.lang.String"/>  
    <form-property name="password" type="java.lang.String" />  
</form-bean>

* In your Action subclass that uses your form bean:
  + import org.apache.struts.action.DynaActionForm
  + downcast the ActionForm parameter in execute() to a DynaActionForm
  + access the form fields with get(field) rather than getField()

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;  
import org.apache.struts.action.ActionMessage;  
import org.apache.struts.action.ActionMessages;  
  
  
**import org.apache.struts.action.DynaActionForm;**  
  
public class DynaActionFormExample extends Action {  
 public ActionForward execute(ActionMapping mapping, ActionForm form,  
 HttpServletRequest request, HttpServletResponse response)  
            throws Exception {          
 **DynaActionForm loginForm = (DynaActionForm) form;**  
                ActionMessages errors = new ActionMessages();          
        if (((String) loginForm.get("userName")).equals("")) {  
            errors.add("userName", new ActionMessage(  
                            "error.userName.required"));  
        }  
        if (((String) loginForm.get("password")).equals("")) {  
            errors.add("password", new ActionMessage(  
                            "error.password.required"));  
        }  
        ...........

**33.How to display validation errors on jsp page?**

<html:errors/> tag displays all the errors. <html:errors/> iterates over ActionErrors request attribute.

**34.What are the various Struts tag libraries?**

The various Struts tag libraries are:

* HTML Tags
* Bean Tags
* Logic Tags
* Template Tags
* Nested Tags
* Tiles Tags

**35.What is the use of <logic:iterate>?**

<logic:iterate> repeats the nested body content of this tag over a specified collection.

<table border=1>   
 <logic:iterate id="**customer**" name="customers">   
 <tr>   
 <td><bean:write name="customer" property="firstName"/></td>   
 <td><bean:write name="customer" property="lastName"/></td>   
 <td><bean:write name="customer" property="address"/></td>   
 </tr>   
 </logic:iterate>   
</table>

**36.What are differences between <bean:message> and <bean:write>**

**<bean:message>**: is used to retrive keyed values from resource bundle. It also supports the ability to include parameters that can be substituted for defined placeholders in the retrieved string.

<bean:message key="prompt.customer.firstname"/>

**<bean:write>**: is used to retrieve and print the value of the bean property. <bean:write> has no body.

<bean:write name="customer" property="firstName"/>

**37.How the exceptions are handled in struts?**

Exceptions in Struts are handled in two ways:

* **Programmatic exception handling** :

Explicit try/catch blocks in any code that can throw exception. It works well when custom value (i.e., of variable) needed when error occurs.

* **Declarative exception handling** :You can either define <global-exceptions> handling tags in your struts-config.xml or define the exception handling tags within <action></action> tag. It works well when custom page needed when error occurs. This approach applies only to exceptions thrown by Actions.

<global-exceptions>  
 <exception key="some.key"  
 type="java.lang.NullPointerException"  
 path="/WEB-INF/errors/null.jsp"/>  
</global-exceptions>

**or**

<exception key="some.key"   
 type="package.SomeException"   
 path="/WEB-INF/somepage.jsp"/>

**38.What is difference between ActionForm and DynaActionForm?**

* An ActionForm represents an HTML form that the user interacts with over one or more pages. You will provide properties to hold the state of the form with getters and setters to access them. Whereas, using DynaActionForm there is no need of providing properties to hold the state. Instead these properties and their type are declared in the struts-config.xml
* The DynaActionForm bloats up the Struts config file with the xml based definition. This gets annoying as the Struts Config file grow larger.
* The DynaActionForm is not strongly typed as the ActionForm. This means there is no compile time checking for the form fields. Detecting them at runtime is painful and makes you go through redeployment.
* ActionForm can be cleanly organized in packages as against the flat organization in the Struts Config file.
* ActionForm were designed to act as a Firewall between HTTP and the Action classes, i.e. isolate and encapsulate the HTTP request parameters from direct use in Actions. With DynaActionForm, the property access is no different than using request.getParameter( .. ).
* DynaActionForm construction at runtime requires a lot of Java Reflection (Introspection) machinery that can be avoided.

|  |
| --- |
|  |

**39.How can we make message resources definitions file available to the Struts framework environment?**

We can make message resources definitions file (properties file) available to Struts framework environment by adding this file to struts-config.xml.

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

**40.What is the life cycle of ActionForm?**

The lifecycle of ActionForm invoked by the RequestProcessor is as follows:

* Retrieve or Create Form Bean associated with Action
* "Store" FormBean in appropriate scope (request or session)
* Reset the properties of the FormBean
* Populate the properties of the FormBean
* Validate the properties of the FormBean
* Pass FormBean to Action