Struts

1. [What is struts2?](#_What_is_Struts2?)
2. Components
3. [What is interceptor in Struts2?](#_What_is_interceptor)
4. [What is struts-default package and what are it’s benefits?](#_What_is_struts-default)
5. [What is the default suffix for Struts2 action URI and how can we change it?](#_What_is_the)
6. [What is the default location of result pages and how can we change it?](#_What_is_the_1)
7. [How can we upload files in Struts2 application?](#_How_can_we)
8. [Which design pattern is implemented by Struts2 interceptors?](#_Which_design_pattern)
9. [Does Struts2 action and interceptors are thread safe?](#_Does_Struts2_action)
10. [Which class is the Front Controller in Struts2?](#_Which_class_is)
11. [What is ValueStack and OGNL?](#_What_is_ValueStack)
12. [What is the use of namespace in action mapping in Struts2?](#_What_is_the_2)
13. [Which interceptor is responsible for mapping request parameters to action class Java Bean properties?](#_Which_interceptor_is)
14. [What is the difference in using **Action interface** and **ActionSupport** class for our action classes, which one you would prefer?](#_What_is_the_3)
15. [How can we get Servlet API Request, Response, HttpSession etc Objects in action classes?](#_How_can_we_1)
16. [What is the use of execAndWait interceptor?](#_What_is_the_4)
17. [How can we integrate log4j in Struts2 application?](#_How_can_we_2)
18. [exceptions\_strus](#exceptions_strus)
19. [frontControlloer](#frontControlloer)
20. [strtus2frmework\_diagram](#strtus2frmework_diagram)
21. [differ\_strus1\_stru2](#differ_strus1_stru2)
22. [struts\_config\_contents](#struts_config_contents)
23. [tags\_struts](#tags_struts)
24. [validation\_plugin](#validation_plugin)
25. [web\_xml\_contents](#web_xml_contents)
26. [validate\_reset](#validate_reset)
27. [struts\_Conststns](#struts_Conststns)
28. [thread\_safe\_frntCtrl](#thread_safe_frntCtrl)

### What is Struts2?

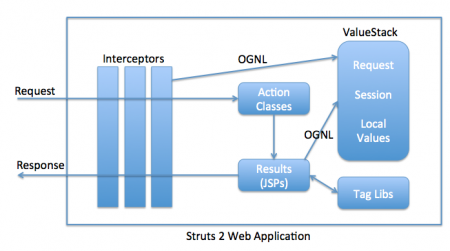
**Apache Struts2** is an open source framework to build web applications in Java. Struts2 is based on **OpenSymphony WebWork** framework. It’s highly improved from Struts1 and that makes it more flexible, easy to use and extend. The core components of Struts2 are Action, Interceptors and Result pages.

Struts2 provides many ways to create Action classes and configure them via struts.xml or through annotations. We can create our own interceptors for common tasks. Struts2 comes with a lot of tags and uses OGNL expression language. We can create our own type converters to render result pages. Result pages can be JSPs and FreeMarker templates.

1. What are Struts2 core components?

Struts2 core components are:

* 1. Action Classes
  2. Interceptors
  3. Result Pages, JSP or FreeMarker templates
  4. ValueStack, OGNL and Tag Libraries

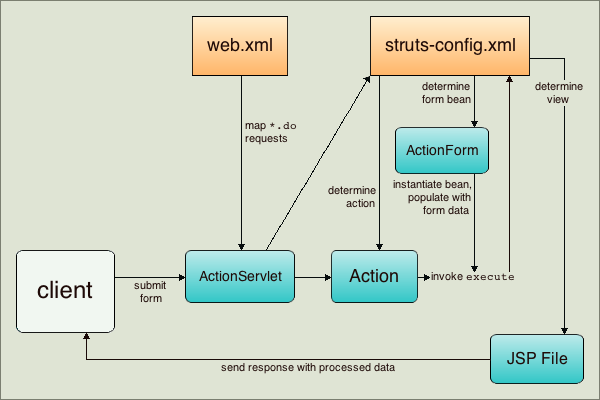
[](http://cdn1.journaldev.com/wp-content/uploads/2013/09/Struts-2-Architecture-Diagram.png)

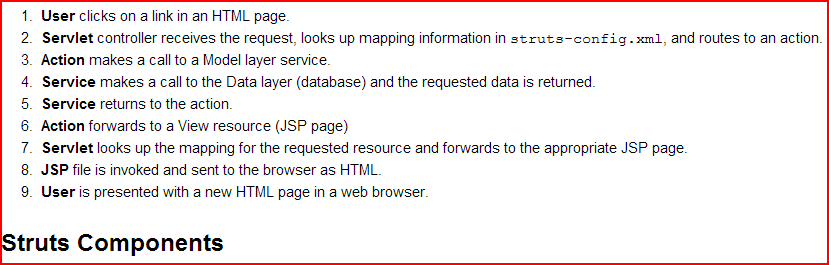
### What is interceptor in Struts2?

Interceptors are the backbone of Struts2 Framework. Struts2 interceptors are responsible for most of the processing done by the framework, such as passing request params to action classes, making Servlet API request, response, session available to Action classes, validation, i18n support, etc.

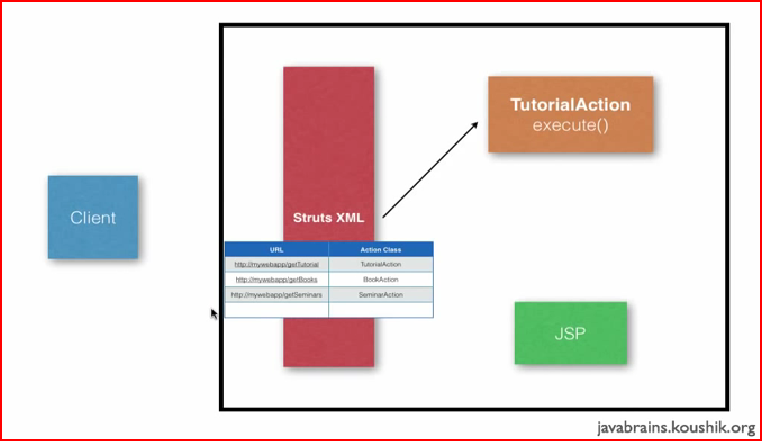
ActionInvocation is responsible to incapsulate Action classes and interceptors and to fire them in order. The most important method for use in ActionInvocation is invoke() method that keeps track of the interceptor chain and invokes the next interceptor or action. This is one of the best example of Chain of Responsibility pattern in Java EE frameworks.

<https://netbeans.org/images_www/articles/72/web/struts/workflow.png>









<http://www.tutorialspoint.com/struts_2/struts_examples.htm>

<http://www.journaldev.com/2354/struts2-interview-questions-and-answers#interceptor-life-cycle>

**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 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>

bydefining <global-Exception> tag inside struts config.xml file.

Programmatic and Declarative way is some time also asked as followup questions given candidate’s response on knowledge on Struts.

**Key:** The key represent 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.

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

Ans: Another classic Struts interview question it’s higher on level than previous interview questions because it’s related to important validation concept on 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 file in Struts should be place somewhere inside the /WEB-INF folder of the application to keep it safe from 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>

Read more: <http://javarevisited.blogspot.com/2011/11/struts-interview-questions-answer-j2ee.html#ixzz345cn4Kgi>

**Difference between Struts 1 and Struts 2 ?**  
  
Ans. Filters were introduced in Struts 2. Action classes are singleton in Struts 1. Base Action is an abstract class in Struts 1 whereas its  an interface in Struts 2. Struts 2 uses **Value Stack technology** for View.

**4****. Difference between filters and Interceptors ?**  
  
Ans.  
  
Interceptors were introduced **with Struts 2.**  
**Interceptors are mapped to Actions where as Filters are mapped to URL.**  
Interceptors are called before Actions where Filters are called before Servlet.  
Filters are used for tasks like Authentication , URL Redirection whereas Interceptors are used for tasks like Validation , Data loading etc.  
Interceptors are used for providing solution for common concerns between actions where Filters are used for common concerns between different url requests.

**What are the classes used as part of struts framework ?**  
  
Ans. Action servlet , Action class , Action Form , Action Mapping , ActionForward

**7. Can we have multiple struts config files ?**  
  
Ans. Yes we can have multiple struts config files. We need to add the names of those config files in web.xml.

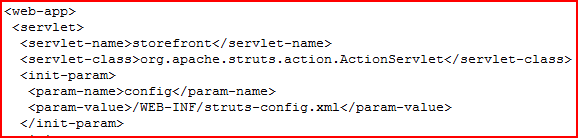
**What are the contents on web.xml in struts application ?**  
  
Ans. Servlet definition , Servlet Mapping , Struts Config files , Landing Page , Plugins - Validation , taglib etc.

<taglib>

<taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-logic.tld</taglib-location>

</taglib>



**How is declarative handling of exceptions done in Struts ?**  
  
Ans. By defining <global-Exception> tag inside struts config.xml file.

### What is struts-default package and what are it’s benefits?

**struts-default** is an abstract package that defines all the Struts2 interceptors and commonly used interceptor stack. It is advisable to extend this package while configuring our application package to avoid configuring interceptors again. This is provided to help developers by eliminating the trivial task of configuring interceptor and result pages in our application.

### What is the default suffix for Struts2 action URI and how can we change it?

The default URI suffix for Struts2 action is .action, in Struts1 default suffix was .do. We can change this suffix by defining struts.action.extension constant value in our Struts2 configuration file as:

|  |  |
| --- | --- |
| 1 | <constant name="struts.action.extension" value="action,do"></constant> |



**10.** **What do you know about Validation plugin ?**  
  
Ans. Validation plugin can be enabled by making an entry in web.xml. There are 2 xml files -  
  
validation.xml that holds the form bean, its elements and the validation to be performed on them. and validator-rules.xml holds the rules for the validation specified in validation.xml.

<!--  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>

**12.** **What configurations are stored in struts configuration file ?**  
  
Ans. URL - Action - Form Bean - View Mapping , Interceptors ,  Message Resources.

**1.** **What validate() and reset() method does ?**  
  
Ans. Validate and reset methods are defined in Form Beans. Validation method is used to validate the request parameters where reset method clear Action Form Data members.

What is the default suffix for Struts2 action URI and how can we change it?

The default URI suffix for Struts2 action is .action, in Struts1 default suffix was .do. We can change this suffix by defining struts.action.extension constant value in our Struts2 configuration file as:

|  |  |
| --- | --- |
| 1 | <constant name="struts.action.extension" value="action,do"></constant> |

### What is the default location of result pages and how can we change it?

By default Struts2 looks for result pages in {WEBAPP-ROOT}/{Namespace}/ directory but sometimes we want to keep result pages in another location, we can provide **struts.convention.result.path** constant value in Struts2 configuration file to change the result pages location.

Another way is to use @**ResultPath** annotation in action classes to provide the result pages location.

### How can we upload files in Struts2 application?

File Upload is one of the common task in a web application. Thats why Struts2 provides built in support for file upload through FileUploadInterceptor. This interceptor is configured in struts-default package and provide options to set the maximum size of a file and file types that can be uploaded to the server.

What are the differences between Struts1 and Struts2 or how Struts2 is better than Struts1?

Struts2 is designed to overcome the shortcomings of Struts1 and to make it more flexible, extendable. Some of the noticeable differences are:

|  |  |  |
| --- | --- | --- |
| **Components** | **Struts1** | **Struts2** |
| Action Classes | Struts1 action classes are forced to extend an Abstract Class that makes it not extendable. | Struts2 action classes flexible and we can create them by implementing Action interface, extending ActionSupport class or just by having execute() method. |
| Thread Safety | Struts1 Action Classes are Singleton and not thread safe, that makes extra care on developer side to avoid any side effects because of multithreading. | Struts2 action classes gets instantiated per request, so there is no multithreading and makes them thread safe. |
| Servlet API coupling | Struts1 APIs are tightly coupled with Servlet API and Request and Response objects are passed to action classes execute() method. | Struts2 API is loosely coupled with Servlet API and automatically maps the form bean data to action class java bean properties that we mostly use. If however we need reference to Servlet API classes, there are \*Aware interfaces for that. |
| Testing | Struts1 action classes are hard to test because of Servlet API coupling. | Struts2 Action classes are like normal java classes and we can test them easily by instantiating them and setting their properties. |
| Request Parameters mapping | Struts1 requires us to create ActionForm classes to hold request params and we need to configure it in the struts configuration file. | Struts2 request params mapping is done on the fly and all we need is to have java bean properties in action classes or implement ModelDriven interface to provide the java bean class name to be used for mapping. |
| Tag Support | Struts1 uses JSTL Tags and hence are limited. | Struts2 uses OGNL and provide different kinds of UI, Control and Data Tags. It’s more versatile and easy to use. |
| Validation | Struts1 supports validation through manual validate() method | Struts2 support both manual validation as well as Validation framework integration. |
| Views Rendering | Struts1 uses standard JSP technology for providing bean values to JSP pages for views. | Struts2 uses ValueStack to store request params and attributes and we can use OGNL and Struts2 tags to access them. |
| Modules support | Struts1 modules are complex to design and looks like separate projects | Struts2 provides “namespace” configuration for packages to easily create modules. |

### Which design pattern is implemented by Struts2 interceptors?

Struts2 interceptors are based on intercepting filters design pattern. The invocation of interceptors in interceptor stack closely resembles Chain of Responsibility design pattern.

1. What are different ways to create Action classes in Struts2?

Struts2 provide different ways to create action classes.

* 1. By implementing Action interface
  2. Using Struts2 **@Action** annotation
  3. By extending ActionSupport class
  4. **Any normal java class with execute() method returning String can be configured as Action class.**

### Does Struts2 action and interceptors are thread safe?

Struts2 Action classes are thread safe because an object is instantiated for every request to handle it.

Struts1 interceptors are singleton classes and a new thread is created to handle the request, so it’s not thread safe and we need to implement them carefully to avoid any issues with shared data.

### Which class is the Front Controller in Struts2?

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter is the Front Controller class in Struts2 and every request processing starts from this class. Earlier versions of Struts2 uses org.apache.struts2.dispatcher.FilterDispatcher as Front Controller class.

1. What are the benefits of Interceptors in Struts2?

Some of the benefits of interceptors are:

* + Interceptor plays a crucial role in achieving high level of separation of concerns.
  + Struts2 interceptors are configurable, we can configure it for any action we want.
  + We can create our own custom interceptors to perform some common tasks such as request params logging, authentication etc. This helps us in taking care of common tasks at a single location, achieving low maintenance cost.
  + We can create interceptors stack to use with different actions.

### What is ValueStack and OGNL?

ValueStack is the storage area where the application data is stored by Struts2 for processing the client requests. The data is stored in ActionContext objects that use ThreadLocal to have values specific to the particular request thread.

Object-Graph Navigation Language (OGNL) is a powerful Expression Language that is used to manipulate data stored on the ValueStack. As you can see in architecture diagram, both interceptors and result pages can access data stored on ValueStack using OGNL.

Provide some important Struts2 constants that you have used?

Some of the Struts2 constants that I have used are:

1. **struts.devMode** to run our application in development mode. This mode does reload properties files and provides extra logging and debugging feature. It’s very useful while developing our application but we should turn it off while moving our code to production.
2. **struts.convention.result.path** to configure the location of result pages. By default Struts2 look for result pages at {WEBAPP-ROOT}/{Namespace}/ and we can change the location with this constant.
3. **struts.custom.i18n.resources** to define global resource bundle for i18n support.
4. **struts.action.extension** to configure the URL suffix to for Struts2 application. Default suffix is .action but sometimes we might want to change it to .do or something else.

### What is the use of namespace in action mapping in Struts2?

Struts2 namespace configuration allows us to create modules easily. We can use namespace to separate our action classes based on their functionality, for example admin, user, customer etc.

### Which interceptor is responsible for mapping request parameters to action class Java Bean properties?

com.opensymphony.xwork2.interceptor.ParametersInterceptor interceptor is responsible for mapping request parameters to the Action class java bean properties. This interceptor is configured in struts-default package with name “params”. This interceptor is part of basicStack and defaultStack interceptors stack.

### What is the difference in using Action interface and ActionSupport class for our action classes, which one you would prefer?

We can implement Action interface to create our action classes. This interface has a single method execute() that we need to implement. The only benefit of using this interface is that it contains some constants that we can use for result pages, these constants are SUCCESS, ERROR, NONE, INPUT and LOGIN.

ActionSupport class is the default implementation of Action interface and it also implements interfaces related to Validation and i18n support. ActionSupport class implements Action, Validateable, ValidationAware, TextProvider and LocaleProvider interfaces. We can override validate() method of ActionSupport class to include field level validation login in our action classes.

Depending on the requirements, we can use any of the approaches to create struts 2 action classes, my favorite is ActionSupport class because it helps in writing validation and i18n logic easily in action classes.

### How can we get Servlet API Request, Response, HttpSession etc Objects in action classes?

Struts2 action classes doesn’t provide direct access to Servlet API components such as Request, Response and Session. However sometimes we need these access in action classes such as checking HTTP method or setting cookies in response.

Thats why Struts2 API provides a bunch of \*Aware interfaces that we can implement to access these objects. Struts2 API uses dependency injection to inject Servlet API components in action classes. Some of the important Aware interfaces are SessionAware, ApplicationAware, ServletRequestAware and ServletResponseAware.

### What is the use of execAndWait interceptor?

Struts2 provides execAndWait interceptor for long running action classes. We can use this interceptor to return an intermediate response page to the client and once the processing is finished, final response is returned to the client. This interceptor is defined in the struts-default package and implementation is present in ExecuteAndWaitInterceptor class.

Check out [Struts2 execAndWait interceptor](http://www.journaldev.com/2296/struts2-execandwait-interceptor-example-for-long-running-actions)

### How can we integrate log4j in Struts2 application?

Struts2 provides easy integration of log4j API for logging purpose, all we need to have is log4j configuration file in the WEB-INF/classes directory.

Struts2 automatically configures the log4j for us and we can use directly in the action classes. Notice the Logger initialization in the action class and that we are using in execute method for logging.

What are different Struts2 tags? How can we use them?

Struts2 provides a lot of custom tags that we can use in result pages to create views for client request. These tags are broadly divided into three categories- Data tags, Control tags and UI tags.

We can use these tags by adding these in JSP pages using taglib directive.

|  |  |
| --- | --- |
| 1 | <%@ taglib uri="/struts-tags" prefix="s" %> |