1. **Which design pattern the Interceptors in Struts2 is based on ?**  
   Interceptors in Struts2 are based on Intercepting Filters.
2. **Can annotation-based and XML based configuration of actions coexists ?**  
   Yes
3. **What is struts.devMode and why it is used ?**  
   struts.devMode is a key used in struts.properties file (Can also be configured in struts.xml file as <constant name="struts.devMode" value="true" />) , to represent whether the framework is running in development mode or production mode by setting true or false. If set to development mode, it gives the following benefits : -  
   **>** Resource bundle reload on every request; i.e. all localization properties file can be modified and the change will be reflected without restarting the server.  
   **>** struts.xml or any configuration  files can be modified without restarting or redeploying the application  
   **>**The error occurs in the application will be reported, as oppose to production mode.  
   Also remember that struts.devMode should be marked as false in production environment to reduce impact of performance. By default it is "false".
4. **What is the difference between empty default namespace and root name space ?**  
   If the namespace attribute is not defined in the package tag or assigned "" value then it is called empty default namespace.While if "/" is assigned as value to the namespace attribute then it is called as root namespace.  
   The root namespace is treated as all other explicit namespaces and must be matched. It’s important to distinguish between the empty default namespace, which can catch all request patterns as long as the action name matches, and the root namespace, which is an actual namespace that must be matched.
5. **Which interceptor is responsible for setting action's JavaBean properties ?**  
   com.opensymphony.xwork2.interceptor.ParametersInterceptor is the interceptor class who sets the action's JavaBean properties from request.
6. **What is the difference between Action and ActionSupport ?**  
   Action is interface defines some string like SUCCESS,ERROR etc  and an execute() method. For convenience Developer implement this interface to have access to String field in action methods. ActionSupport on other hand implements Action and some other interfaces and provides some feature like data validation and localized error messaging  when extended in the action classes by developers.
7. **How do you get the HttpServletRequest object in an interceptor ?**  
   Here is the intercept method

[?](http://www.bullraider.com/java/struts2/interview-questions)

|  |  |
| --- | --- |
| 1  2  3  4  5 | public String intercept(ActionInvocation invoke) throws Exception {    ActionContext action=invoke.getInvocationContext();    HttpServletRequest req=(HttpServletRequest)action.get(StrutsStatics.HTTP\_REQUEST);    return null;  } |

In the similar way you can get the response, by using StrutsStatics.HTTP\_RESPONSE in get() method as above.

1. **What is execute and wait interceptor ?**  
   The ExecuteAndWaitInterceptor is great interceptor provided out of box in Struts2 for running long-lived actions in the background while showing the user a nice progress meter or a progress bar. For example while uploading a large file to the server we can use this interceptor to display a nice running progress bar instead of leaving the user in confusion that the application is not responding.This also prevents the HTTP request from timing out when the action takes more than 5 or 10 minutes.
2. **Does the order in which interceptors execute matters ? If yes then why ?**  
   Well, the answer is yes and no.Some Interceptors are designed to be independent so the order doesn't matter,but some interceptors are totally dependent on the previous interceptors execution.For example the validation and workflow interceptors,the validation interceptors checks if there is any error in the form being submitted to the action, then comes the workflow interceptor who checks if validation ( occured in the last) has any error,in presence of error it will not let the rest of the interceptors ( if any ) in the stack to execute.So this is very important that the validation interceptors execute first before the workflow. On the other hand lets say you wrote an interceptors who is doing the authentication and you have the user credential provided ( by the time this executes) it doesn't matter where this interceptors is placed( It is a different fact that you would like to keep it in the top ).
3. **Who loads the struts.xml file ? Which Struts2 API loads the struts.xml file?**In Struts2 FilterServlet is the responsible class for loading struts.xml file as we deploy the application on the container.Unlike Servlet (as with Struts1) needs the load-on-startup tag to load the front controller,a filter doesn't need to have load on startup tag to be loaded as the application is deployed. As with servlet specification a filter is loaded/executed as the application  starts up.
4. **What is the difference between RequestAware and ServletRequestAware interface ?**  
   RequestAware and ServletRequestAware both makes your action to deals with the servlet request, but in a difffrent ways,RequestAware gives you the attributes in the servlet request as a map( key as attribute name and value is the object added),But ServletRequestAware gives you the  HttpServletRequest object itslef giving you more flexibility, with a price that ServletRequestAware makes your  Action class too much tied to the Servlet environment making it dificult to unit test. So whenever a need to access only the attributes use the RequestAware interface.
5. **What is the difference between EL and OGNL ?**  
   OGNL is much like EL in JSPs,a language to traverse or manupulate objects like request , session or application in web context.OGNL stands for Object graph navigation language,which is used internally by Struts2, however we are not bound to use OGNL in our JSPs, we can use EL.But OGNL provides much more facilities than plain EL.For example while El interacts with the objects by means of getters/setters, OGNL supports whatever EL does along with lambda experssion, helps create functions on fly. OGNL has more flexible ways to deal with collection of objects.
6. **What are the difference between ActionContext and ServletActionContext ?**  
   ActionContext represents the context in which Action is executed, which itself doesn't hold any web parameters like session, request etc. ServletActionContext, extends the ActionContext and provides the web parameters  to the Action.