**Struts**

**Overview**

It is used to design MVC architecture.

M Model

V View

C Controller

* Model describes business logic of an application which can be constructed using EJB or beans.
* View describes presentation(output) of an application which can be implemented using JSP.
* Controller describes input of an application which can be developed by Servlet.

**Required packages to develop struts: -**

org.apache.struts.action

org.apache.struts.actions

**Required programs to develop Struts: -**

1) jsp file(view1)

2) Action class

3) ActionForm class (form bean class)

4) either success or failure page(jsp(view2))

5) struts-config.xml(It describes the configuration about struts)

**org.apache.struts.action.Action**

* This class should be inherited by a class to create action classes.
* An Action is an adapter between the contents of an incoming HTTP request and the corresponding business logic that should be executed to process this request.
* The controller will select an appropriate Action for each request, create an instance (if necessary), and call the execute method.

Method: -

public ActionForward execute(ActionMapping, ActionForm, HttpServletRequest,HttpServletResponse);

**org.apache.struts.action.ActionForm (Abstract class)**

* It acts as Model for processing the values.
* An ActionForm is a JavaBean. Such a bean will have had its properties initialized from the corresponding request parameters before the corresponding execute method of Action class is called.
* This class must be subclassed in order to be instantiated. Subclasses should provide property getter and setter methods for all of the bean properties they wish to expose.

methods: -

public ActionErrors validate(ActionMapping, HttpServletRequest);

* for validating given request.

public void reset(ActionMapping,HttpServletRequest);

* for clearing the values from given request.

**org.apache.struts.action.ActionMapping**

To forward a page which means that server to server redirection.

method: -

public ActionForward findForward(String urlCasename);

**Write the following statements in "struts-config.xml”: -**

<struts-config>

<form-beans>

<form-bean name="anyname" type="ActionFormClass" scope="">

</form-bean>

</form-beans>

<action-mappings>

<action type="ActionClass" path="/actionpage(alias name for actionclass)" name="formbeanname" input="/jsp file(view1)" scope="">

<forward name="anyname" path="/jspfile" />

</action>

</action-mappings>

</struts-config>

**Predefined TLD files for html and bean tags: -**

1) http://struts.apache.org/tags-html

2) http://struts.apache.org/tags-bean

**Note: -**

To get value of particular property which has defined in form bean class, write the following statement in .jsp file:-

<bean:write name="beanname" property="propname"/>

**ApplicationResource.properties**

* This file contains set of keys and corresponding values.
* In which we can also create keys and values as required.

To use those keys and values in a jsp page write the following statement: -

<bean:message key="keyname"/>

**org.apache.struts.actions.DispatchAction(Abstract class)**

* An abstract Action that dispatches to a public method that is named by the request parameter "whose name is specified by the parameter property of the corresponding ActionMapping".
* This Action is useful for developers who prefer to combine many similar actions into a single Action class, in order to simplify their application design.

To configure the use of this action in our struts-config.xml file, create an entry like this:

<action path="/actionpath" type="org.apache.struts.actions.DispatchAction" name="formbeanname" scope="request/session" input="/home.jsp" parameter="method"/>

**org.apache.struts.actions.LookupDispatchAction (Abstract class)**

It is mostly like as Dispatch Action class but one exception that in which submit button values and corresponding methods of Action class may differ, but they will be referred (looked up) by keys which are given in applicationresource.properties and Map object in Action class.

subclass of LookupDispatchAction must implement getKeyMethodMap.

method: -

protected Map getKeyMethodMap();