Java means DURGA SOFT..

JAVA FRAMEWORKS

STRUTS

Syllabus:

1. Web Application Development Models
2. Struts Frame Work:-
3. Base Struts Components
4. Front controller
5. Application Controller
6. Inversion Of Controll(Ioc)
7. Data Transfer Object(Dto)/Value Object
8. Struts Initialization
9. Application Development In Eclipse Ide
10. Using Resource Bundile In Struts
11. Applying Programmatic Validation
12. File Uploading In Struts
13. Developing A Struts Application In My Eclipse
14. Adding A Server In Ide
15. Exception Handling In Struts
16. Global-Forwards

WEB APPLICATION DEVELOPMENT MODELS:--

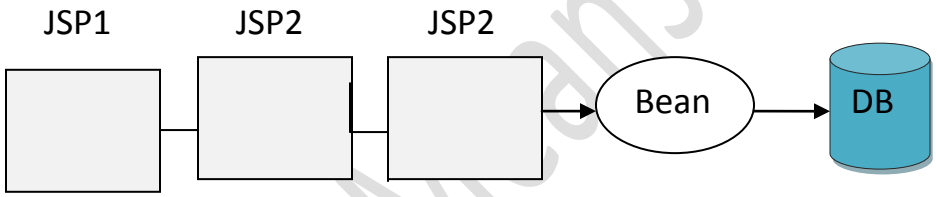
When designing an app by the developers for the same problem diff developers provide diff designs. So designs may not be same but they output the app same.

If one developer wants to understand the designs given by another developer then feel uncomfortable, because the designs may not be I the same way as that developer expected.

So in order to solve the designing problems and also to inform the developers about what technology is used at what place, we got web application development models.

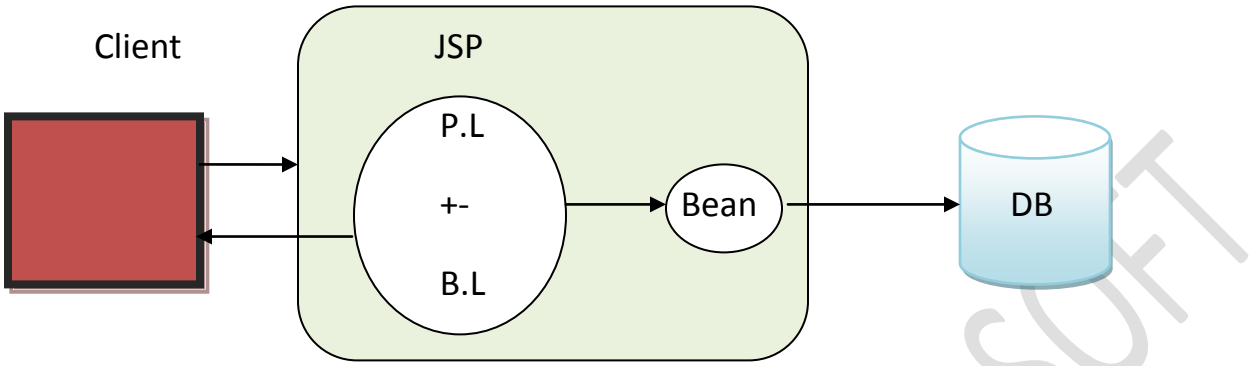
We got two models: - 1) models1 and2) model2

MODEL1:- this model introduced along with JSP technology, in this model a JSP is a responsible to provide all functionality required for a web application may contain any no. of (or) series of JSP pages in order to provide the functionalities.



1) MVC designe pattern.
2) MVG,MVC2 are architectures .

The following diagram rep. Models architecture:-



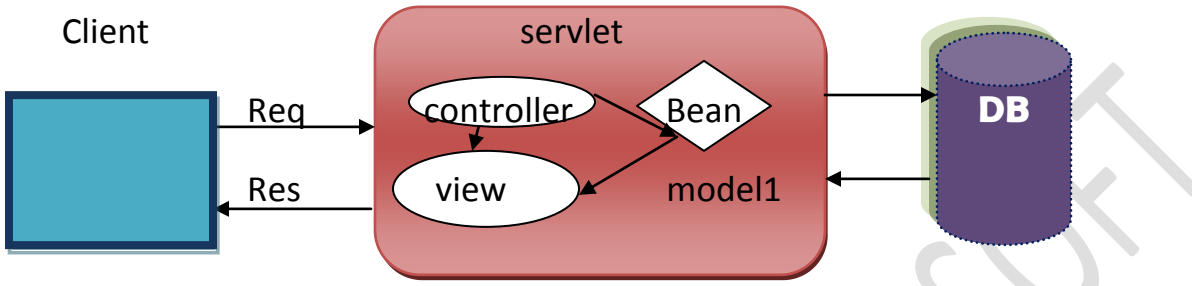
1) **Limitation of Model1**:- the application may contain a series of pages an &if any changes are done in one page then, if need the changes other pages also we call this problem as a page-centric problem.

Because one page is depends on another, all the pages should be developed by a simple developer only, it means division of labor is no possible.

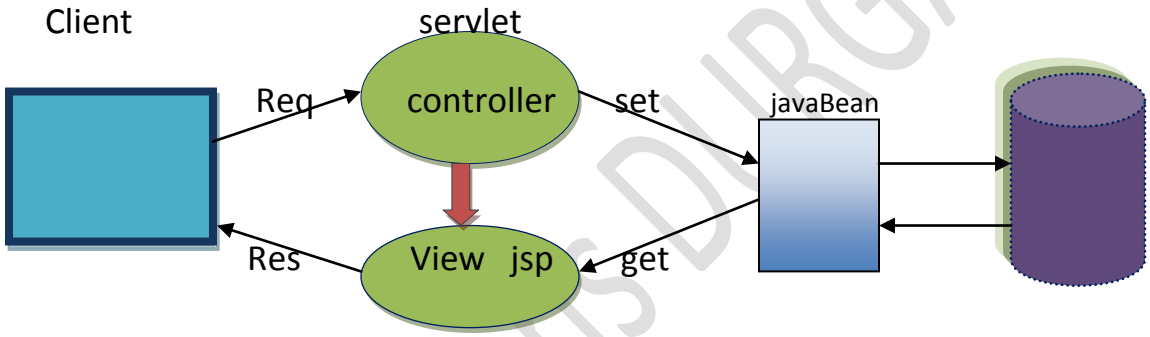
Each page is responsible for providing (or) implementing the logics like input verification, Session management, security……etc. so it increases the size of a JSP and also burden on the developer this application code we call as redundancy.

Implementing B.L in a JSP is not in industry standard, because JSP should contain tags, but not code.

In order to overcome the limitations of model1 we got model2



MODEL:-2



Model2 (MVC) is the combination of servlet and JSP technology

Actually sun micro system introduced, JSP technology as a replacement for servlet technology, but industries realize that for the web app development both servlet and jsp required, and introduced model3 that is MVC architecture.

In the MVC applications, controller is a servlet, iew is JSP (or) html and javaBean (or) ejb (or) spring are Model .

Controller is a component , which containes all primary logics required like sessions and security and also it component an entire flow of an application.

Model component concentrate on business logic of an application.

View component concentrate on presentation.

We call these MVC application also called servlet centric.

In Model2 P.L and B. Logic are separated from each other, so page centric problem is eliminated.

ADVANTAGES:- Business logic is implemented in javabean. So we can reuse the same bean component in other web application also.

all logics are separated, so changes from one logic does not effect other logic, it means we got flexibility.

Because each component contain a specific logic so complexity is reduced.

Division of laboure is possible, because the logics are devided into diff component , so that application time reduced.

RULES TO FOLLOW IN MVC:-

1) An application can contain any number view and Model components, but it contain only single controller
2) Two presentation pages (jsp's) can not communicate each other directly if communicated then we get page centric Problem again.
3) Model and view components can not talk each other. It should be done all ways through controller as a mediator.
4) Each component should contain, its specific logic. We can not mix multiple logics in a single component.

Question:-why servlet is a controller?

1) If a jsp used as a controller, then we need to insert java code in jsp. But this is against industry standard. So jsp is not suitable as a controller.
2) If we use java bean as controller then it cannot receive (or) handle http request. So bean cannot be used as controller.
3) Finally servelt is suitable for **used** controller.

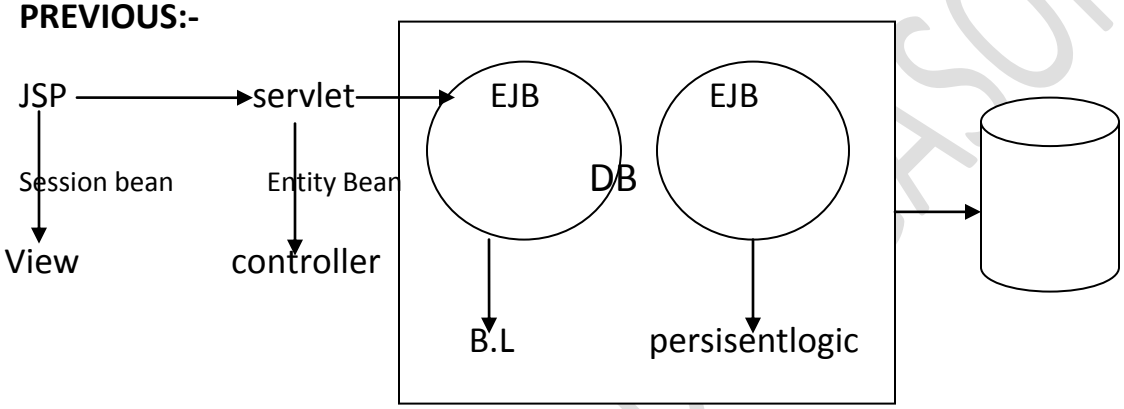
Question:- why jsp as view?

1) If we use servlet as view then we need to write presentation logic inside java code. the second reason is if we modify the presentation logic then we need reduced play and re compile and restart the server.
2) If javaBean is used as view then it cannot given reponse page.
3) Finally jsp is suitable to use as view, because if any presentation logic is modified, then it will be automatically effected.

IMPQUESTION:-Why Bean as a model?

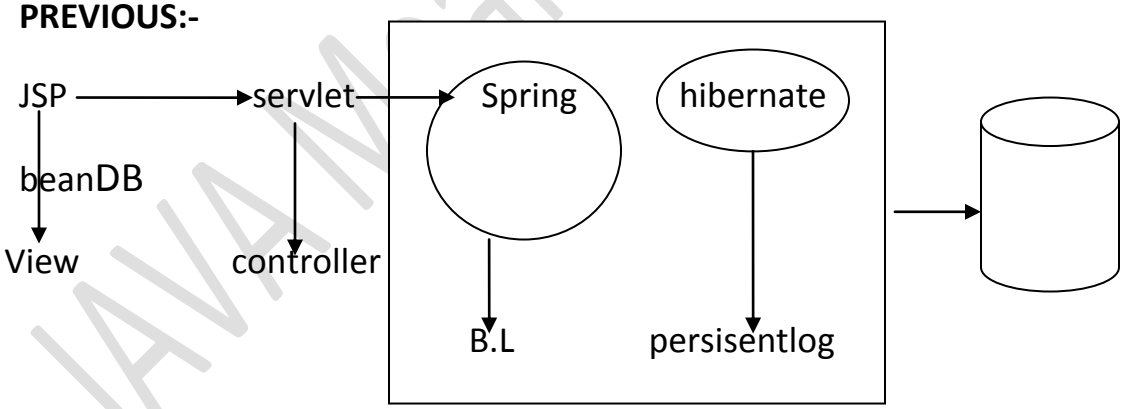
1) If we use servlet as a model then we cannot get middle ware services required for the business logic.
2) If we use jsp as a model then again we should insert java code in jsp this against industry standard
3) Finally a Bean component is suitable for implementing business logic of an application. So it used as model.

PREVIOUS:-



PRESENT:--

PREVIOUS:-



STRUTS FRAME WORK:-

WHAT IS FRAME WORK?

1) Every system (i.e pc) will follow some commonalities. All applications will accept inputs from key board, produces output on to the screen and shares a common memory.
2) The above concept is the foundation per the development of the frame work.
3) A frame work contain set of libraries (or) classes. Each library will provide a basic foundation for the development of an application.
4) We can define a framework as, it is a software, provides reusable components with basic functionality and the makes an application development faster and easyer.
5) A framework will provide an abstraction layer and allowes the programmer to develop their application as faster and reduces burden on the developers.

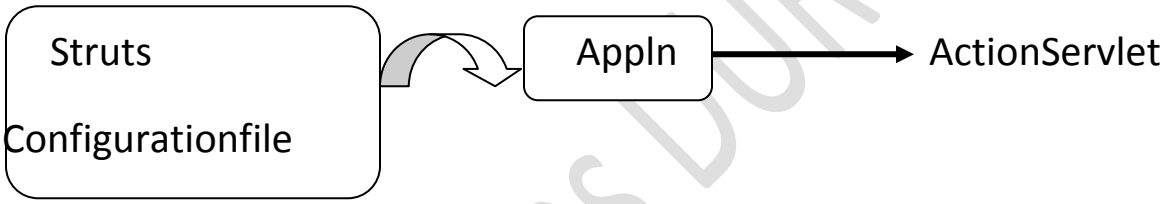
COMMON FUNCTIONALITY (OR) PRINCIPLE OF FRAMEWORK:-

Every frame work will follow a standard notation it is an application contains one (or) more central configuration files and these files are loaded into an application, so that an high level object is generated and this object is responsible for maintaining an entire application

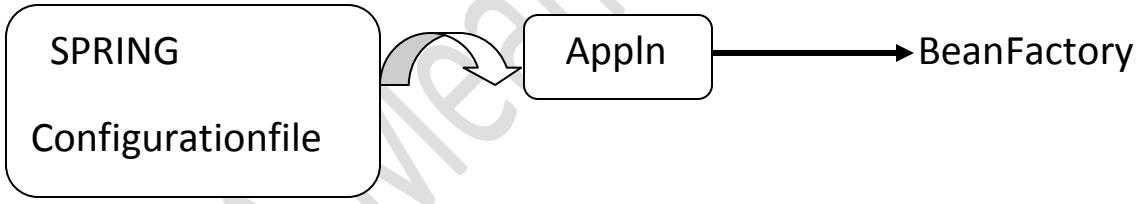
For example, in case of struts, we write struts configuration files, which are loaded into an application and produces Action servlet object this object maintaine struts application.

The same rule applicable for all frameworks like spring, hibernate…etc.

STRUTS:-



SPRING:-



q) what is struts?

Struts is an open source framework given by Apache software foundation(ASF) , under one of its project is called jakarla project

Struts is a frame work , used to develop web applications for java using mvc2 , architecture.

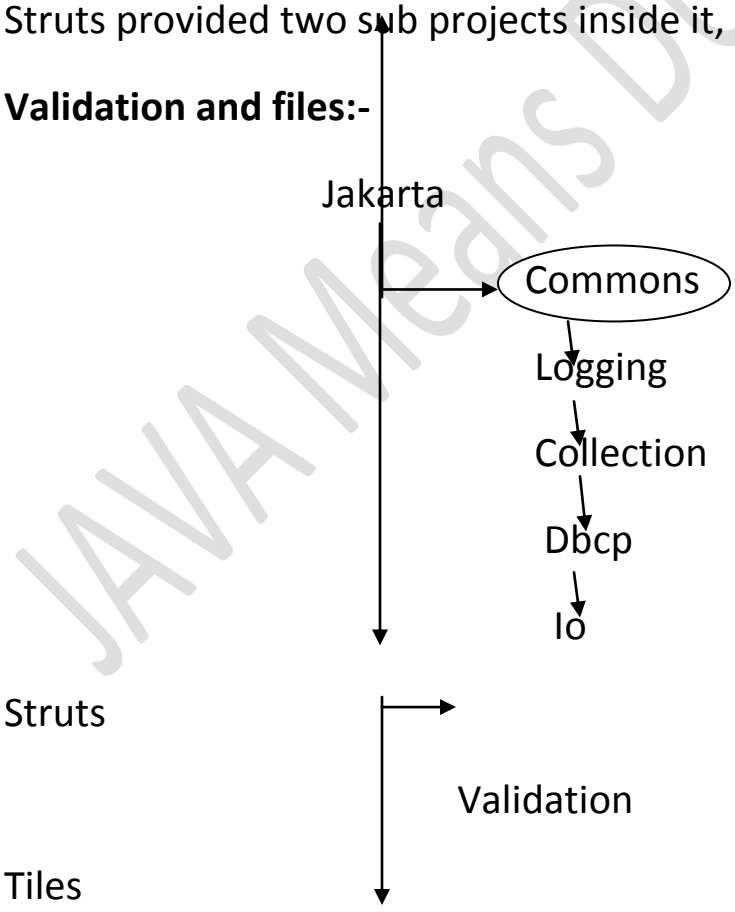
Jakarta project provided two sub projects.

1) Commons.
2) Struts.

Commons projects is having approximately 30's of projects like logging collections, dbcpo…. Etc.

Struts provided two sub projects inside it, those are validation and files.

Validation and files:-



We can define struts frame work also as an implementation of MVC2 archetecture of JSP.

Types of frameworks:--

Literarly (or) generally frame work are devided into two types

1) Invasive
2) Non-invasive

Invasive frame work means it will force the programmer to create their classes by extending (or) implementing their classes from predefined classes (or) interfaces.

Non-invasive frame work means, it does not force the programmers to extends (or) implement their classes from any predefined classes (or) interfaces given by that frame works.

For example, struts is an invasive frame work and hibernate and spring are non-invasive frame work.

Struts→calanahan→may-2004 introduced.

Hibernate-> gavinking→ jbass server also developed

Spring→ rod Johnson→ interface21

1) Validation
2) I 18n(internationalization)

q) **why struts is so popular?**

Because of the following two reasons.

Struts support extensive validation, where no other frame work can support.

Struts is having inbuild support for I 18N

Struts frame work is provided the following 3 categories of caomponents for struts programmers (or) developers, inorder to develop web application using these component

1) Base frame work
2) Jsp tag library
3) Internal frame work's

Base frame work provides many struts components used for implementing same logic fr struts app.

Jsp tag libries are the given for the development view component either for accepting input (or) produce o/p.

Internal frame work's:- are validation frame work and files frame work's these frame will add additional capabilities to a struts application.

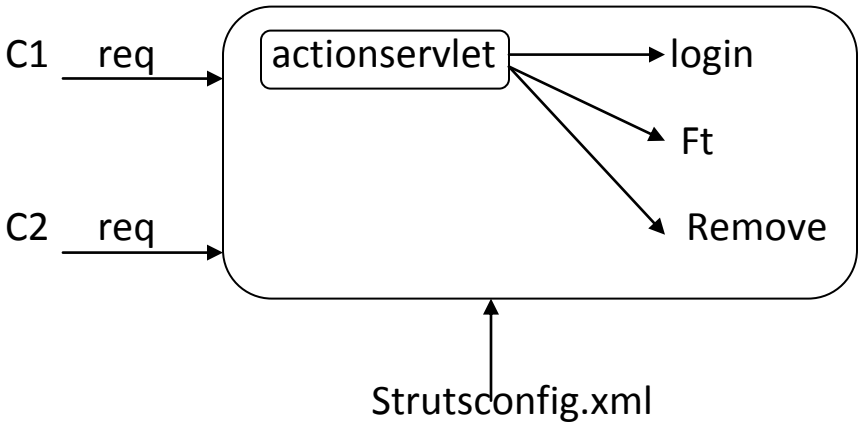
Note:- struts frame work's is capable to integrate with any after frame work's, that is available in the market today(or) even that is going to come tomorrow.

Base struts components:-

Action Servlet:- it is heart of struts app.

It is controller

It is entry(or) exit and control the entire apP



P c Actionservlet extends. Hhp servlet

{

}

Iq) servlets implements only one ways that is implementing servlet interface.

Actionservlet taking help from request processer class.

Public class AS extends HHP

{

Public void init(sc)

{

Object of RP;

Initialize

}

P v doet()

{

Process()

}

P v do post()

{

Process of RP

}

}

Action from component:-

To store the input values at server

Public class loginform extends A.F

{

Setter and getter and some validation methods.

}

Base components of struts:-

Action servlet:-this component is the front controller of a struts application.

In a struts application every request enters into a struts application and every response from a struts app will be traveld through this component only.

This component is acting as entry point for the request and exit point for the request.

The entire flow of a struts application will be managed by this component only.

Action servlet is only one servlet component that exits in a struts application, we can either use this class directly in our instated of using this class.

This component will get the flow details of an application through a struts configuration file.

This class is extends from HHPservlet and it has implementation of init(), decret(),dopost() and destroyer.

Note:--Actionally action servlet does not maintain the flow off an application it takes help of another component internally called request processor. This class is said to be an application controller.

When a servlet is an acting as front controller o client will be send a request to the servlet directly. the servlet only received that request while it is travelling. So per a front controller servlet, we can not use exact match url pattern either we should use extension mach (or) directory match.

2)**Actionform:-** this component is also called a formBean.

This component is used to store the input vales given by a client at a client side, on the input values in this component only.

We call action from as a bean and it is reusable.

In order to store input values given by a client, we need some properties and the request parameter names and the property names, both must be same.

We can not use directly actionBean class. We need to create a derived class struts class by extending actionform.

3)Action:- action class containe logic action classis a place, where we implement business logic (or) wrapper around b.logic (integration logic)

If an app need every small b.logic then we implement it inside action class only and if complex b.logics are required then we will communicate the outside model components like, ejb, spring..etc from action class.

Action class is a place from the control goes out of struts framework's to out side model and the action class place in to where the control comes back to the struts frame work

Action class is not a servlet but followes servlet model. It means single instance and multiple threads model.

While developing real time applications, we should not write b.logic implementation in action classs because the action class will not get the

real time services like transaction and security from the container automatically.

In case of real time app action class access mediator to call the b.logic implemented at out side model componenets.

By default action class is not thread safe but it is possible to convert action class into thread safe, by applying synchronization mechanism.

4)ActionForward:- this component containes a string value, by depending on which a view is selected by the controller(Action servlet)

Actionforward object will be transferred from action class back to actionservlet aftet b.logic is execution's is completed.

5)Actionmapping:-this component is dynamically constructed and it containes in formation about an actionform an action class and resulting views (like success)

Internally actionservlet communicate will actionmapping object to get these details, whenever a request is given.

Configuring actionservlet:- action servlet is a servlet, so we need configuration it in web.xml

We need inform the container that an object of this class should be created . before any request is given, so we need include <load-on-startup-value>

In struts application , except this action servlet all the remaining component need to be configured in struts configuration file.

Actionservlet loads struts.configuration file as an initparameter to the servlet.

<servlet>

<servlet.name>action</->

<servlet-class org.apache.struts.action.actionservlet<-

<load-on-startup>/

<init-param>

<param-name>config</>

<param-value>/WEB-INF/struts.config.xml</-

</init-param>

</servlet>

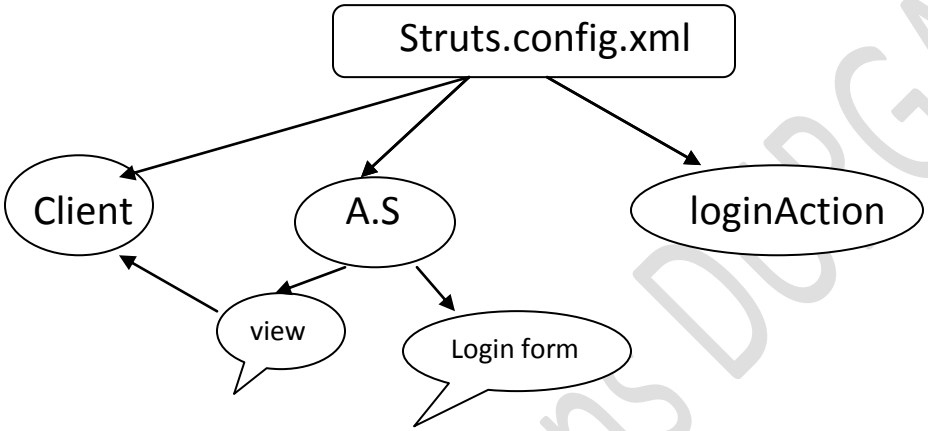
<servlet.mapping>

<servlet-name>action</servlet-name>

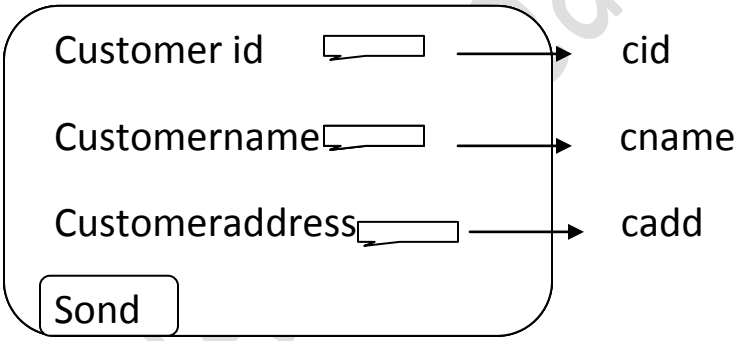
<url-pattern>*.do</url.>

</jer-mapping>

</web-app>



Input.jsp



Public class customerform extends actions

{

Cidd, cname,cadd

Setters&getters

Reset()

Validate()

}

Reset method is used for before sending the data the default values are assinge.

Scope=”request”” if client request to time then only object action class objects created only one object is only perclient.

i.q) **configuring (and) creating action form**

1) actionform objects in struts is for starting client input and then to apply validation on the given input.

2) in action form class we create properties, in order to store input values and the request parameter names and property names in the form bean class, both must be same.

3) in a form bean class, we include the following 4 methods. For storing and validating the input.

1)setter methods.

2) getter methods.

3) reset method

4) validate method.

4) reset method is used to initialize(or) reset the bean properties with default values, before storing the new values in to the variables.

Validate method is apply validation on the user input.

i.q) action form object will be created by the controller servlet (or) A.S and by default

this object will be stored in session scope. It means form bean object per each client session.

If we want a formbean object per each request then we need to modify it's scope from session to request.

In struts frame work's , we have totally the following 6 types of actionform.

1) Org.apache.struts.actionform.
2) Org.apache.struts.action.dynaactionform.
3) Org.apache.struts.validator.validatorform
4) Org.apache.struts.validator.dynavalidatorform
5) Org.apache.struts.validator.validatorActionform
6) Org.apache.struts.validator.dynavalidatorActionform

The following is an example of actionform.

Public class loginform extends actionform

{

Private string uname,pwd;

Setters

Getters

Public void reset(Actionmapping mapping,httpservletrequest req)

{

}

Public actionerrors validate(Actionnmapping mapping,httpservletrequest req)

{

}

}

Row to work strut-config.xml:-

<strut-config>

<form-beans>

<form-beanname=”frm1”type=”loginform”/>

</form-beanss>

</struts-config>

Note:- we need configure formBean in struts configuration file the struts configuration file name can be <any name> .xml.in general we provide the name as struts-config.xml the following example represents configuration of formbean in strutcon-figurationfle.

<strut-config>

<form-beans>

<form-bean name=frm1" type="loginform"/>

</form-beans>

</strut-config>

Note :- the elementformBeans> must be immediately after root the element <strut-config>. It means we can not include any other elements In b/w them.

Creating and configuration an action:-

Action clas is a place where we insert either b.logic (or) integration logic.

Inside actionclass the only method, in which are we should implement our logic is execute().

After logic is executed it resume an actionformword object back to actionservlet.

By depending on actionforward object one view is selected along multiple views.

Example:- public class loginaction extendaction

{

Public actionforward execute(actionmapping mapping)

Actionform form, httpservletRequest req, hhpservletresponse res)

Throws exception

{

--

--

If(---)

Return mapping find forward(“success”);

Else

Return mapping.find forward(“failure”);

}

}

Each action class of struts should be configure in struts configuration file.

All action classes must be configured under-action mapping element.

While configuring an action, we should include it's <forward> elements also.

Example:-struts-config.xml

<s-c>

<formBeans>

<action-mapping>

<actionpath="/login name="frm1" type="loginaction">

<forward name="success" path="/success.jsp"/>

<forward name="failure" path="/failue.jsp"/>

</action></action-mapping>

</s-c>

How to make your class as singleton:-

Public class sample

```
{
Private stafic Boolean
Public sample()
{
If(flag)
{
Flag=true;
}
Else
Throw new exception();
}
}
```

Finally this class allowes only one object then it is a singleton.

Note:- Design pattern used in struts:-

Singleton:-if a class allowes as to create only one object then that class is said to be a singleton class.

In struts p.work, actionservlet is a singleton class

The struts framework developers explicitly made actionservlet class as singleton

If we want make a java class as singleton then it is possible in two approaches.

1)by throwing an exception , when typing to create more than one object

2)by creating a private constructed a factory method.

Example:- public class sample

{

Private static sample s1=new sample();

Private sample()

{

}

Public static geetinnstance()

{

Returns;

}

}

Static methods are allowed.

Static variables and static methods

Static methods can not allow super,this key works.

Sample ob1=sample .getinstance()

Sample ob2=sample.getinstance()

In the above code ob1 and ob2 will containe same objects which means ob1==ob2 simple address same. With two objects are equal to==then we say that both are rep single object.

I.Q):what is the diff an ordinary servlet and an action servlet?

In same cases an ordinary servlet may not be singleton, in all cases actionservlet is singleton.

The reason is , inn case of an ordinary servlet a web container will make it as singleton. But in case of actionservlet, this strutsframework developers only madeit as singleton.

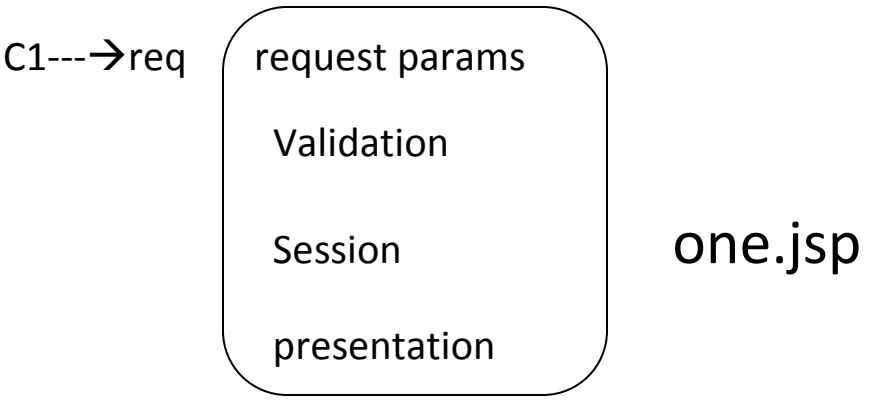
I.Q):how to make a java class as imuutable?

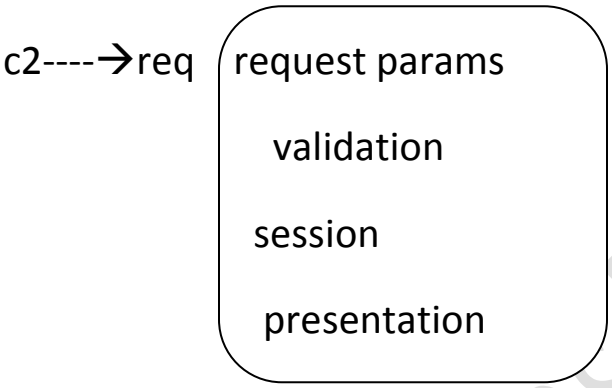
We need to follow the rules.

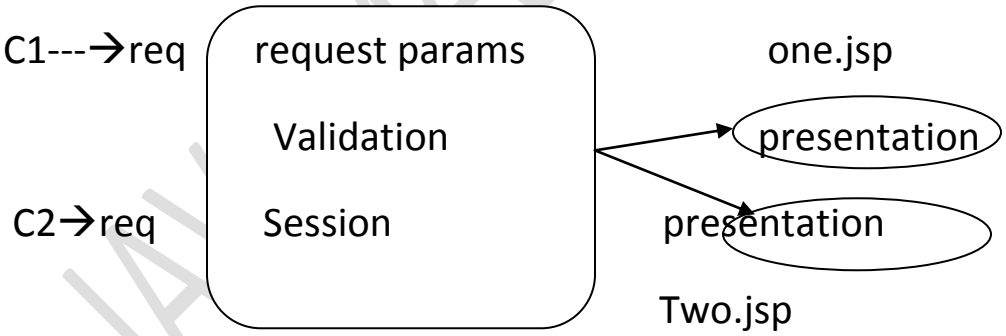
1) Create only private, final variables in the class.
2) Create a parameterized constructor.
3) Only implement getterr methods in the class, without setter methods.

Frontcontroller:-while developing MVc based web application centerlized common logics into one component is possible through front controller.

If no front controller is used then the common logics are de centerlized , so that we will get duplication of the logics. When developing struts applications, we need a front controller servlet and it is an actionservlet and it is singleton class.







Application controller :-

An helper component of front controller is called an application controller.

In an MVc based applications, there exists a separate application controller component per each module but only one front controller will exits for all module.

While developing MVc based applications, it may containe more than one module. So if there is only frontcontroller then the front controller has to maintaine the entire flow. It becomes complex.

So, in order to reduce the burden on frontcontroller each module flow will be maintained controller are maintained by frontcontroller.

In an MVc application, there can be any number of application controller, but there must be a single frontcontroller.

INVERSION OF CONTROLL(IOC):-

Separation an implementation of an object from how that object is created is called IOC.

The constructing of an object will be taken care by an external entity and the programmer who created the class, does not any knowledge of creating object of for it.

Inverting the control means redirecting the object construction from a programmer to an external person in ioc, we have two categories.

1) Dependency lookup
2) Dependency injection

When working with struts framework, we have both lookup and injecting in the application.

In case of lookup, if a first object is depending on second object then the first object should be get the explicitly by second object.

In case of injection if a first object is depending an second object, then while creating first object, automatically second object is inserted, by the external person who is creating the first object.

In struts framework's action class object depends on actionforrm object and actionform object is automatically injected into actionclass, so this is a dependency injection mechanism .

In lookup mechanism, explicitly one object with get the other object. Mostly this lookup we are going to observer when struts communicate with ejb

Data transfer object(dto)/value object:-

When developing app's for realtime, it may contain multiple modules (or) layers and data transfer is required from one layer to another layer.

If we transfer individual values from one layer to another layer it increases no.of round trips b/w the layer and automatically he performance of application will be decreased.

In order to overcome the above problems, instated of sending individual values, store all the value into an object and then transfer at time all the values, by transferring that object this pattern will call as a DTO pattern.

In struts framework, Actionform is an object stores input values and transfer these values to the business layer(Action class), in the form of an object. So an Actionform object is a DTO in struts.

I.Q)Note:- the common design pattern used by all java applications (or) projects is DTO.

Struts instalization:-www.struts.org.apach

Struts framework is not a setup file to install in a system.

All framework's are not installable software.

Working with struts framework means, adding jar files into the lib folder of our application and adding the jar files required at the class path.

Visit struts.apache.org and down loade struts-versions-all-zip.

Unzip the above zip file and take struts-blank-version.war fro apps folder.

Copy the above war file into a newly created folder and extract the war file by using jartool.

jar xvf struts-blank-version-war

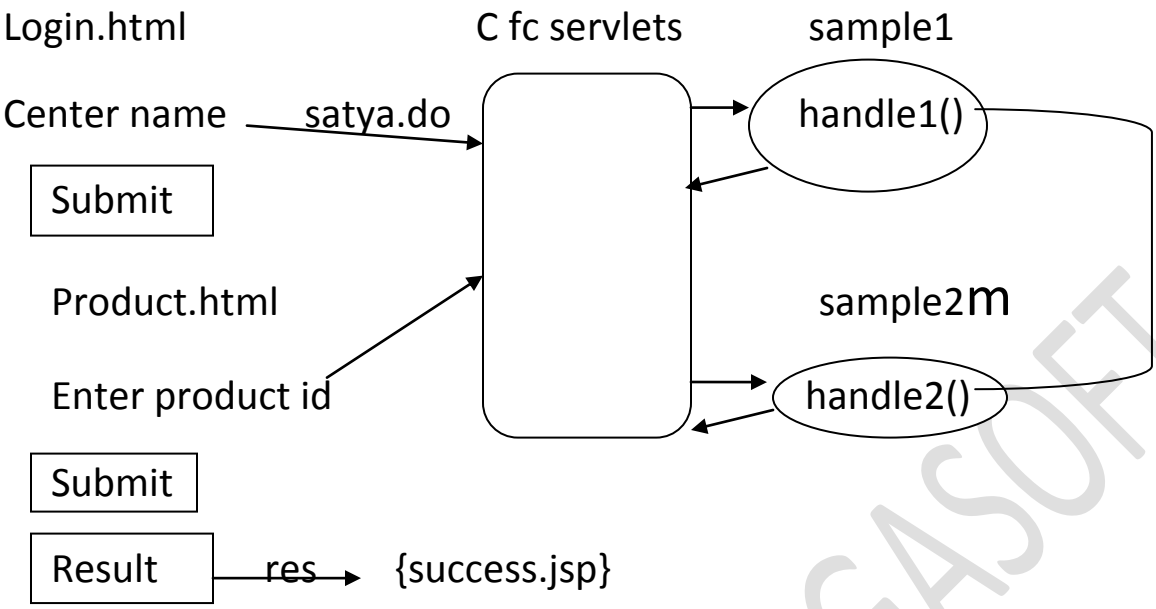
under WEB-INF/lib folder, we will get the jar files related to struts framework.

Into our web app, we need add these jar files into lib folder.

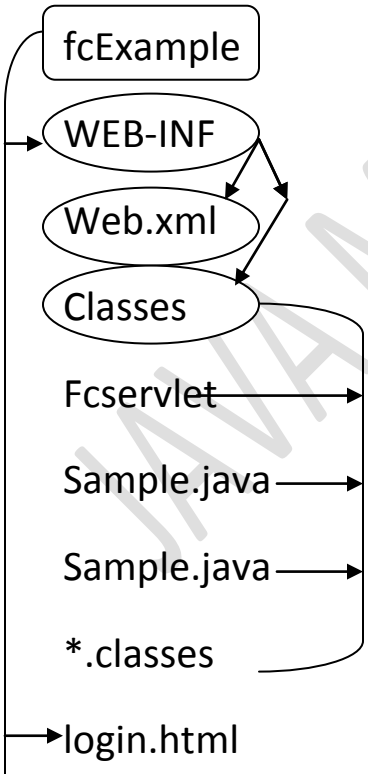
Note:-the best version of struts is struts-1.3.8

Example:- the following example is based on MVc model and the example is creating for our front controller servlet.

The frontcontroller servlet maintaice the flow of the application.



Directory structure:-



→product.html

↘success.jsp

↘failure.java

i.q):- which collection is best one in collection

hashmap() object

map m1new hashmap();

m.put("/sathya.do",newsample())

m.put("/product",newsample());

instance of operator duty is check the condition

login.html:-

<formaction="sathya.do">

Entername:<inputtype=textname="t1">

<input type=submit value=”click”>

</form>

Product.html:-

<form action=”prod.do>

Enter productid=<input type=text name=”p1”>

<input type=submit value=”click”>

</form>

</centr>

Web.xml:-

<web-app>

<servlet>

<servlet.name>frontcontroller</→

<servlet-class>fcservlet</s….>

<load-on-startup>1<1>

</servlet>

<s.m>

<or>*.do>

<15-m>

</web-app>

Fcservlet.java:-

Import java*.servlet*;

Import java.io

Import java.urtil.*;

Public class fcservlet extends Httpservlet

{

Map m;

Public void init(servletconfig sc) throws servlet exceptions

{

M=new hashmap();

m.put(“/sathya.do”, new sample1());

m.put(“prod.do”,new sample2());

```
}
Public void abcret(htt req, res) throw SI,IOE
{
String result;
String rpath=req.get.servletpath();
Objecto=m.get(r.path);
If(o instance of sample1)
{
Sample1 s1=(sample1)o;
Result=s1 handle1(req,res);
}
Else{
Sample2 s2=(sample2)o;
Result=s2.handle2(req,res);
}
Request is pather rd=req.getrequestispather("t1");
{
Rd.forward(req.res);
}
```

Sample1.java:-

```
Import javax.servlet.http.x;

Public class sample1

{

{

Public string handler(htt res(). H res)

{

String str=req.getparameter(“t1”);

If(str.equals(“satya”));

Return”success”

Else

Return “failure”;

}

}
```


Sample.java

Import

```
Public class sample2
{
Public string handle2(m req hres)
{
String str=req.getparameter(“p1”);
If(str.equals(“p123”))
Return “success”;
Else
Return “failure”;
}
}
```


Servlet api.jar will provide tomcat server.

Ant also deployment.

Context path is nothing but application name

Success.jsp:-

```
<%out.println("<h1>success</h1>");
%>
```

Failure.jsp:-

```
<%out.println("<h1>failure</h1>");
```

NOTE:- in the above example, in Fcservlet class, we used hashmap object for storing the flow details.

Hashmap object is the best collection in java collection api

In the Fcservlet class, we used instanceof operator, it's perpouse is to know whether a particular object belongs to a particular class type (or) not if ok it returns true other wise it returns. False

Note:- this instance of operator is mostly used in java messaging service (5mb)

If we want compile the above servlet and both sample1 and sample2

classes ten we need to set seervlet api.java

%classpath% rep old value of class path.

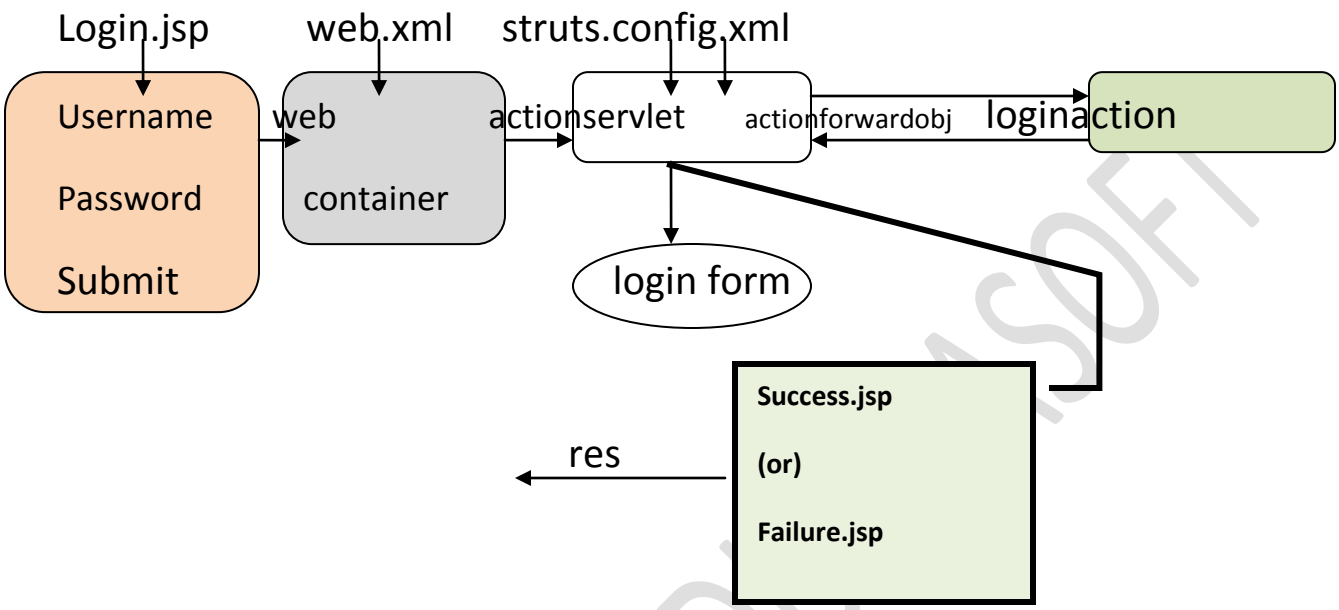
Deploye app→c:toncats.s/webaps folder

Start the server

Open the browser ad send request.

http://localhash:Rov/Fcexample/logic.html

finally in the above example we have a controller, two model component and 4 presentation pages. So this is an example related to mvc model.



struts.config.xml:-

<s-c>

<form-beans>

<form-bean name=”f1” type=”loginform”/>

</form-beans>

<action mapping>

<action path=”/check” name=”f1” type=”loginaction”>

<forward name=”success” path=”/success.jsp”>

<forward name=”failure” path=”/failure.jsp”>

</action>

</action.mappinng>

</s-c>

1) When a client request is given, a web container receives the request, because the request is given for a active resource dynamic.
2) Web container reads web.xml(DDF) and identifies a servlet suitable for the given request.
3) The container identified that actionservlet is suitable to receive this request and then transfer the contain to the actionservlet.
4) Actionser vlet reads struts configuration file.
5) Actonservlet verifies whether the given request path is matching with an actionpath(or) not
6) If action path matches then the controller servlet verifies whether logical name of action and logical names of formbean, both are equal(or) match.

7) If names are equal then the controller servlet first creates actionfor(logicalform)ed then the controller servlet
8) After form bean executed actionclass. In action class execute() will be called.
9) If comple b.logic required then control is moved to the outside model component.

10) Persistence logic is also a part of b.logic . so the model component is communicate with db.

11) The model component returns control along with result back to action class.

12) Execute() method of action class returns actionforward object back to actionservlet.

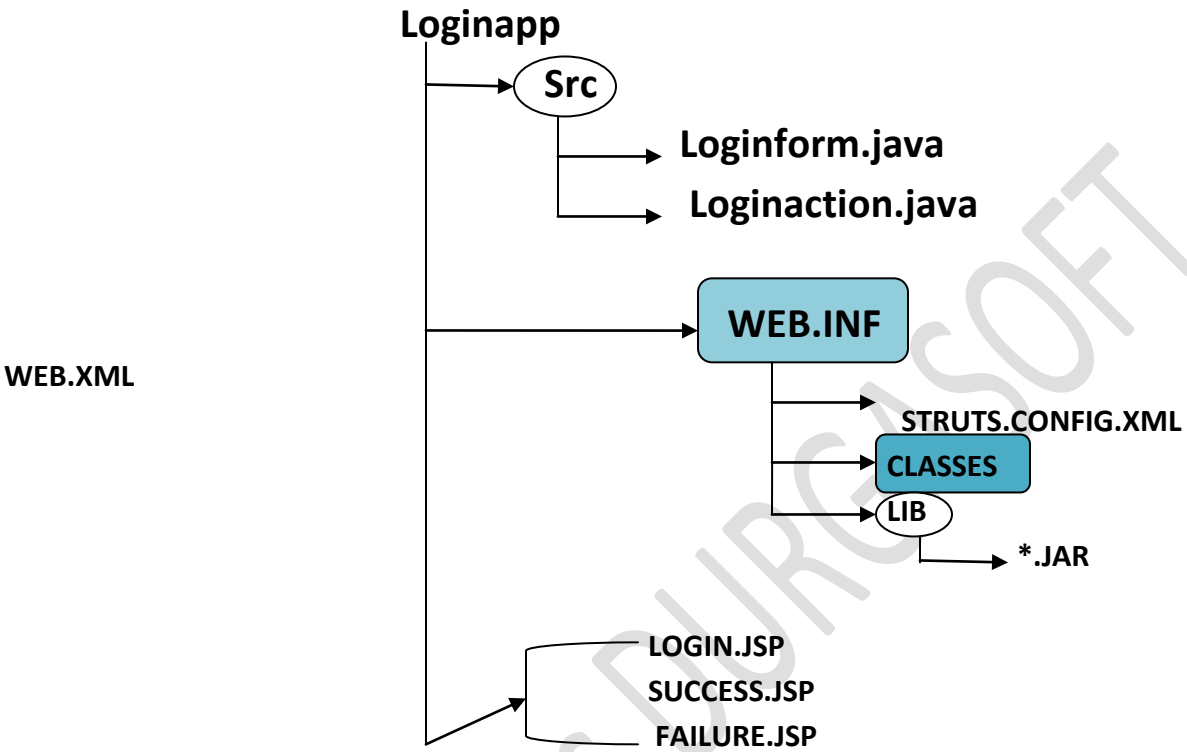
13) Actionservlet again reds struts configuration file and verifies a suitable forward. Per the returned int value (or) object.

14) Action servlet selects an appropriate view component

15) Finally action servlet uses request distachers forward mechanism and forward a suitable view as a response back to the client browser for the given request.

Note:-activally actionservlet doesnot perform this entire flow management internally actionservlet takes help of request processor class, to do this entire flow management for a given request.

Example:- the following example is a login application using struts framework.



<init-params>

<p-n>config</p-n>

<p-v>/<WEB-INF/classes/struts.config.xml<p-v>

</init-params>

Application development in eclipse IDE:-

Step1:-start eclipse ide dbl click on eclipse.exe from eclipse folder and enter same work space name→ok

Step2:-click on file menu→ new→ project→ javaproject→next→ enter projectname if version is change(select version)(loginapp)→ next→finish

Step3:- RT click o WEB-INF →new→folder→enter folder name as classes. Similarly lib folder also

Note:-to shift (or) transfer from package explorer to navigator ,click an windowmenu→show view→navigator.

Struts provided same predefined tags those tags we can use the JSP pages

Step5:-click an projectname→new→file→enterfilename(login.jsp)→finish

<!—login.jsp---

%@taglib uri="http://struts.apache.org/tag-html" prefix="html"%

<html:form action="login.do">

Username:<html:text property="uname"/>

Password:<html:password property="pwd"/><lrr>

<html:submit value="chick"/>

</html:form>

Struts html.tld no need write from struts words.

Note:- while creating the above jsp, we used html tags given by struts framework. The reason is , when we submit the request then immediately these input values will be stored automatically in a framebean object.

2)if we use ordinary html tags in jsp then the input values will not be stored in a formbean object

3)to use these html tags provided by struts, we need to add a standard uri and prefox in our jsp

<!—success-jsp---this components can be same html,xml,jsp-

<%out.println("<h1>login success</h1>");

Failure.jsp:-<%out.println("<h1>failure</h1>"); %>

Step6:-RT click on project name→properties→select java build path at leftside→select source tab at rightside→select project name->remove→add folder→ select src→ok→click on browser button→ select classes→ok→select libraries tab→ click on add external jars button→ select two jar files.

1) Struts-core-1.3.8-jar
2) Servlet-api.jar→ok

Step7:-RT click on src→ new→classes→ enter class name→ super class name(Action form)→ finish.

LOGIN FORM.JAVA:- import org.apache.struts.action.actionform;

```
Public class loginform extends actionform
{
Private string uname,pwd;
Public string getpwdu{
Return pwd;
}
```

```
Public void setpwd(string pwd)
{
This.pwd=pwd;
}
Public string getname()
Return uname;
}
Public void setuname (string uname){
This.uname= uname;
}
}
```

STEP 8:- RT click on src→new→class→name(login action)->superclassname→(org.apache.strutsaction.action)→finish

//loginaction.java:

```
Import javax.servlet.http.httpservletRequest;
Lmport org.apache.struts.action.action;
Import org.apache.struts.action.actionmapping;
Public class loginAction extends action
{
Public actionFordward execute(Actionmapping mapping)
```

```
Actionform form, htt req,htt res) throws exception
{
Loginform lf=(loginform) form;
String s1= lf.getname()
String s2= lf.getpwd()
Actionforward of= null;
If(s1.ewab(“satya) && s2.equab(“satya”))
{
Af= mapping.findward(“success”);
}
Else
{
Af=mapping.figforward(“failure”);
}
Return of
}
}
```


STEP 9:- RT click on WEB-INF→new→file(web.xml)→finish

```
< !=- web.xml→
<web-app>
<servlet>
<servlet-name>action</->
```

<servlet-class>org.apache.struts.action.Actionservlet</->
<load-on-startup>1</→
<init-param>
<param-name> config</param-name>
<param-value>/WEB-INF/struts-config-xml<1->
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>action</->
<url-pattern>*.do</->
</serv-mapping>
</web-app>

STEP:- RT click on web-inf→ new→file→ enter filename→struts-config.xml)→ok

Copy the DTD in the existing project.

</DOCTYPE

---1-3>

<struts-config>

<form-beans>

<form-bean name= “frm1” type=”login form”/>

</form-beans>

<action-mapping>

<action path=”/login” name=”frm1” type=”loginAction”>

<forward name=”success” path=”/success.jsp”/>

<forward name=”failure” path=”/failure”.jsp”/>

</action>

</action-mapping>

</struts-config>

STEP :- copy struts jar file into lib folder of our web application. Because container does not know about the struts fw’s that way we inform the tomcat by processing some jar files.

Tomcat know about only jsp and servlet.

USING RESOURCE BUNDILE IN STRUTS:-

1)in a struts application, we use resource bundle, in order to get lables(or) error messages.

2) in a resource bundle, the data should be stored in the form of key value pairs, where both key and values are used defined.

3) while creating a jsp, instated of manually typing lables, we can get them from a resourcebundle.

4) while applying validations. On the user input in a form bean class, if a validation fails, then the error message must be taken from the resource bundle only.

5)while creating input jsp pages, if we want to get lables from a resource bundle, then we need to use<bean:message>

The bean tags are provided by struts fw in a told file called struts-bean-tld.

If we want use bean tags in our jsp then we need the following taglib directive in our jsp.

<%taglib uri=http://struts.apache.org/tags.beanprefix=”bean”%>

For example we want to get a lable from username from a resourcebundle then we need the following tag.

<bean:message key=”input.user”/>

When we use resourcebundle in a struts application then we need to do the following two things.

1)store the bundle at classes folder.

2)configure the bundle in struts.configuration file.

In struts-config.xml,we need the following element after closing <action.mapping>

<message-resources parameter-“sathya”/>

APPLYING PROGRAMATIC VALIDATION:- →in a struts application, we can apply validation on the user input values, either programtically declaratively

→programatically means manual coding get declarative means through xml.

If we want apply validation on the input then we need to override validation() in our formbean class.

→while applying validation, if any errors are occure then we need to get error messages from the resource bundle only.

→while validation , if an error is occurred then we need to create an object of actionmessage class by passing key of the error message.

→we need to store all actionmessage class obstructs in to actionerrors class object.

→actionerrors class object stores group of actionmessage class objects

FOR EXAMPLE: public actionerror validate(mapping,req)

{

Actionerrors errors=new actionerror();

If(uname.length()==0)

{

Actionmessage am1=new actionmessage("userwrong");

Errors.add("uname",am1);

}

Return errors;

}

- ➔ While executing validation, If any errors occurred then we should get back againe our input page, to inform about this information, we should write validate and input attribute to the action element in struts.configurationfile.
- ➔ Example: <struts-config>

-

--

<action path=”/login” name=”frm1” type=”loginaction”validate=”true” input=”/login.jsp>

Bydefault false

</action>

In order to display error messages on the browser in our input page, we should write <html:error/>

NOTE:- while displaying the error messages, we can display either all errors at one place (or) individual errors for each property separately

→if we want display individual errors then we need to pass property name.

<html : errors property=”uname”/>

Example:- the following example is for applying validations on the input values and to use resource bundle in the application.

The following example is an enhancement to the previous applications only some files need changes and trainings files are used as it is.

</-login.jsp→

<% @ loglib uri=http://struts.apache.org/tags-html/prefix="html"%>

<% @ taglib uri=http://strut.apache.org/tags-beam prefix="html" %>

<html:form action="logindo">

<bean: message key="input.user"/>: <html:text property="uname"/><html:errors property="uname"/><hr>

<bean:message key= "input.pwd"/> :<html:password property ="pwd"/><html:errors property="pwd"/>

<html:submit value="click"/>

<html:form>

LOGIN FORM.JAVA:-

Import org.apache.struts.action.actionfrom;

Import org.apache.struts.action.ActionMapping;

Import org.apache.struts.action.ActionMessage;

Import org.apache.struts.action.ActionErrors;

Import javax.servlet.http.HttpservletRequest;

Public class loginform extends Actionform;

{

Private string uname,pwd;

Setter & getters.

Public actionerrors validate (ActionMapping mapping, HttpservletRequest req)

{

Actionerrors aes= new actionerrors();

If(“”equals(uname) \\uname.length()==0)

In real time use one condition

{

ActionMessage am1=new ActionMessage(“user.error”);

Aes.add(“uname”,am1);

}

If(“”.equals(pwd)|| pwd.length()==0

{

Actionmessage am1=new actionmessage(“pwd.error”);

Aes.add(“pwd”,am2);

}

Return aes;

}

}

In struts.config.xml, we need the following change and the remaining configuration is same as previous application.

</action-mappings>

<message-resources parameter=”sathya”/>

Store the following satya.properties file in the classes folder

#SATHYA.PROPERTIES:-

Input.user=username

Input.pwd=password

User.error=<front color=red>username is required </fat>

Pwd.error=<front color=blue>password is required

NOTE:-while applying programmatic validation by overriding validate() then we have the following drawbacks.

1) If the number input values are more then the size of the validate() will be increased. And it increases the burdon on the programmer.
2) In validate() mostly we duplicate the same logic for multiple fileds. this increases to data redundancy more input values.
3) In order to over comes the above problem, in struts we can apply declaration validation.

DYNAACTION FORM:- in actionform, we have the following two mager draw backs.

1) If presentation jsp page has modified after action form class is generated . then againe we need to modify our action form. And after that recompile and redeploye and restarting the server is required.
2) If the number of presentation (or) input pages more with diff fields then we need to manually create that many number of actionform classed it increases the burdon on the programmer.
3) In ofer to overcome the above drawbacks of an actionform, we got dynaActionform.

4) dynaActionform is class given in org.apache.struts.action package and it is extended from actionform.
5) In case of dynaActionform , we does not required create manulate actionform classes
6) If case of dynactionform, we need to directly configure dynactionform class is to struts-config file.

For example:- <form-beans>

<form-bean
name="f1"type="org.apache.struts.action.dynaActionform">

<form-property name="uname"type="java.lang.string"/>

<from-property name="pwd" type="java.lang.string"/>

</form-bean>

</form-beans>

DYNAACTION:- public class dynaActionform extends AF

{

Private hashmap dynavalues= new hashmap()

```
Public void set(string key,object value)
{
Dynavalues.set(key,value)
}
Public object get(string eky)
{
Return dynavalues.get(key); daynavalues
}
Public map getMap()
{
Return dynavalues()
}
}
Public class login action extends Action
{
P AF execute(------)
{
DAF daf=(DAf) from;
Object o1=daf.get(“uname”);
String u=(string)o1;
```

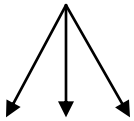
--

--

}

}

Collection **map**



List set queue HashMap© storedMap(interface)

DynaActionform is faster than the actionform (slow) because so many setters & getters.

NOTE :- DyActionform class is going to store input values given by a client into hash map object.

→in DAF class, we do not have setters and getter method for each property. Instated, we have set and get method

When a form is submitted, internally for each request parameter set() will be called and it will store the input values into Hashmap object. Finally Hashmap object is going to containe the input value submitted by a client.

→in action class of our application, we need to call get() of Dyafrom, to read input values given by a client

→in DAF class, another method given called get map() it return the HashMap object and it containe all input values given by a client in the form of (key, values) pair.

→the internal source code of DAF. Will look like as follow.

Code:- public class DAF extend AF implements DynaBean.

{

Private HashMap dynavalues=new HashMap()

Public void set(string key, object value)

{

Dynavalues. Set(key, value);

}

Public object get(string key)

{

Return dynavalues.get(key);

}

Public Map getMap()

{

Return dynavalues;

}

}

}

→if we use req then tight coupling is accure. If we change servlet .API then compulsory change Action class also.

Note:- interface satya

{

Class java

Nested class

{

--

--

}

}

Durga.java

Map.entry:- if we use static then it is called nested class.

if we can't use static then it is called inner class.

→in the execute method of Action class, we need to type caste ActionForm object into DyAFrom and then call get() by passing key(property name); to read the values.

Ex:- public class loginAction extends Action

{

Public AF execute(----)--------exception

{

DAF daf=(DAY)form;

Object 01=daf.get("uname");

String u=(string)01;

String p=(string)daf.get("pwd");

--

--

}

}

→comparatively DAF provides high performance then ordinary AF. Because DAF used HashMap provides high performance among also calculations of java.

DRAWBACK:- if we use DAF then input validations are not possible at FormBean level.

in this case we need write validations at action class level. But writing input validation in action level is not good approach.

Note:- if we apply client side validations using java script then we can eliminate the draw back of dynaActionForm.

i.**q:-** why can't we use HttpServletRequest object of execute method, to read input values given by a client.

a) if we use a request object API(methods) then our action class becomes tightly coupled with servlet API, it means, if servlet API has modification class also.

In order to provide losse coupling b/w Action class and servlet API, we do not use request object in the execute() for reading input.

IMPLEMENTING VALIDATION IN ACTION CLASS:-

➔ In case of DAF, we need to apply serverside validation in execute () no of Action class only.

➔ While validating the input, if any errors occure object (Actionmessage object) into ActionErrors object.

➔ Finally we need to store ActionErrors object into request object. To do this we need to call a method called saveErrors()

➔ Ex:- {

```
Public class loginAction extends Action
{
Public AF execute(--------) throws Exception
{
ActionErrors aes=new ActionErrors();
If(-----)
{
--
--
}
If(! Aes.isEmpty())
{
saveErrors(req,res);
}
```

--

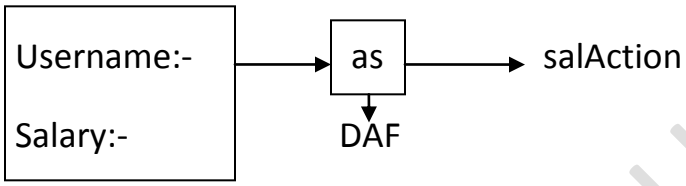
--

}}

If we save ActionErrors object in to request object then any it is display the error message on browser it is using <html:errors)> on the browser

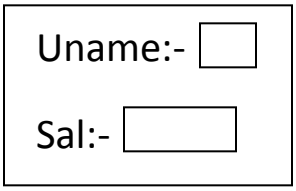
→DAF does not support validation at Action level.

Enter.jsp



U entered salary 8,200 it is invalid

Enter.jsp



In java two types datatypes.

1) Value type→ converting value type into reference type is called boxing
2) Reference type→ revers is called unboxing.

Int k=49; →value type

Integer i= new integer(k)→ reference type

Int k= i.intvalue(); } unboxing

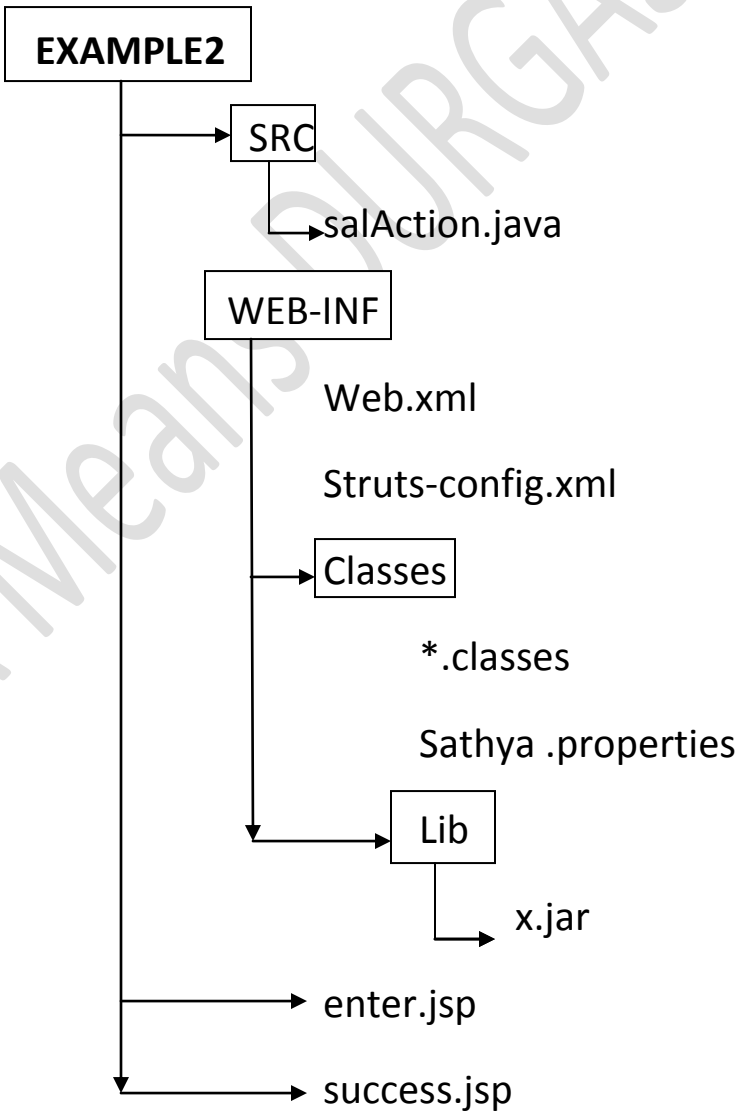
-->save Errors(req,res) →this method is given base class. Ie. Action that way no need to necessary objects directly we can call

Servletcontext ctx= getserveltcontext() –directly call.

Ex 3:- the following example is for using DAForm and validating the input at action class.

If input salary is valid then it return success and bonus value will be printed if not valid then againe input page will be given back and dynamic error message is printed on the browser.

EX2:-



< %-- enter.jsp--%>

< % @ taglib uri=

< % @ taglib uri=

< html: errors/>

<html : form action= “test.do”>

<bean:message key=”enter.name”/>:<html:text property=”ename”/>

<bean:message key=”enter.sal”/> :<html: text property=”esal”/>

<html.submit value=”click”/>

</html:form>

STRUTS.CONFIG.XML:-

<strut-config>

<form-beans>

<form-bean name=”f1” type=”org.apache.struts.action.DAF”>

<form-property name=”ename” type=”java.lang.string”/>

<form-property name=”esal” type=”java.lang.double”/>

</form-bean>

</form-beans>

<action pathj=”test” name=”f1” type=”salAction”>

<forward name=”success” path =”/success.jsp/>

<forward.name=”failure” path=”/enter.jsp”/>

<action>

</action mapping>

<message-resource parameter =”sathya”/>

</s-c>

salAction.java

import org.apache.struts.action.*;

import javax.servlet.httpx;

public class salAction Extend Action

```
{
Public AF execute(--------)
{
DAF df=(DAF) form;
String on=(string)df.get("ename");
Double d=(Double)df.get("esal");
Double s=d.double value();
ActionErrors aes=new ActionErrors();
If(S>8000)
{
ActionMessage =new ActionMessage()" sal.invalid",d);
Aes.add(esal",e);
}
If(!aes.is Empty())
saveErrors (req,res);
if(s<=8000)
{
Integer i1=new integer(500);
Req.setAttribute("bonus",i1);
Retrun mapping.find forward("success");
```

}

Else

Ref

}}

In the above Action class while constructing AM object, along with the key, we are passing double object also as a parameter, because the error message is a dynamic message. And if we want replace a value in place of a parameter of message then we need to pass an object but not a variable.

In the above Action class we called saveError() directly. It means this method is given action base class. And we can call base class method directly in the derived class.

If the given salary is <: 8000 then we are assigning some bonus and this bonus is printed in the browser by the success page. In order to transfer the bounus value from Action. Class to jsp , we are using attributes api of request object.

If we want to transfer a value from action class to a jsp then we need to write setAttribute() in the Action class and getAttribute() in the

JSP:- durga.properties.file

sathya.properties

Enter.name= username

Enter.sale=salary

Sal.invalid= <h2> o entered salary {0} , it is not valid </h2>

In the above we use parameter {0}. It is to replace dynamic value at run time. While constructing a message. We can use a maximum of parameters(from {0} to {3})

Success.jsp:- < % -- success.jsp--%>

< % object o=request.getAttribute(“bonus”);

Integer i=(integer)o;

Int r= i.intValue();

Out.println(“bonus”=+r %>

Upload.jsp

Select a file

Submit

<input type = file>->ordinary html

<html:file…./> struts is give

Public class upload extends AF

{

Private formFile ff;

When submit button then file path is stored in ff.

}

Public class uploadAction extends Action

{

P AF execute(----)

Upload uf= (upload) form;

Formfile f=uf.getFf();

Byte b[] f.getfileData();

<html:form action= “upload.do”

Enctype=”multiplication/form-data”>

Select a file=<html:file property:”ff”/>

<html.submit/>

</html:form>

FILE UPLOADING IN STRUTS:- while working with servlet (or) jsp individually, if we want apply (or) implement fileuploading mechanism then we have depend on third party API's provided.

So that uploading is possible.

forExample,we generally use java zoom API per uploading a file from client side to serverside in awebapplication.

If we use third party Apl's then our application depends on third party libraries.so we have to add those libraries in to our application lib folder and as well as into the class path.

In struts fw's we have build in mechanism for uploading a file on to a server.

In struts fw's a free defined class is provided for storing the uploaded file path and its data called FarmFile.

FarmFile is given in org.apache.struts.upload.* package this is the smallest package is struts, this package is containe only one class that is FormFile.

The following changes arre required in a struts application in order to develop file uploading.

1)in input jsp page, we should add enctype attribute for html.form with a value multipart formdata.

→multipart/formdata means the uploaded file can containe any type of data.

<html:form ation=”upload.do” enctype=”multipart”/fordata”>

In order to get a file text box that is a textbox with a browse button, we need to use on html tag called <html.file property=”ff”/>In actionform class,we need a property of type FormFile and then generate the setter and getter per it.

Example:- public class uploadform extends actionfrom

{

Private FormFile ff;

→setter

→getter.

}

4)in action class of struts, we need get the formFile object and then by using this methods of this class we need to read the uploadfile data.

Finally we need FileoutputStream per written the uploaded file date into a serverside file.

Forex:- public class uploadAction extends Action

{

Public AF execute(---) throws Exception.

}

Uploadform uf=(uploadForm)form;

FormFile f=uf.getFf()

Byte b[]=f.getFileDate();

String frame=f.getFileName();

FOS fos =new FOS (fname)

Fos.write (b)

…..

}

}

DEVELOPING A STRUTS APPLICATION IN MY ECLIPSE:-

Step1:- open my eclipse ide and enter some work space name

Step2:- click file menu→ new→ project→ expand→myEclipse→expand java enterprise projects->select web project → next→ enter project name(upload project)→ finish

Step3:- RT click on projectname→MyEclipse→add struts capabilities→ remove package for names and default application resources→ finish.

Step4:- RT click on proect name→ new→other →expand my eclipse→ expand web→struts→expand struts1.2-> select → struts1.2 Form→ next→name(frm1) superclass(Actionform) →form type(uploadForm) → finish

UPLOADFORM.JAVA:-

```
Import

Public class uploadform extends actionForm

{

Private FormFile ff;
```

```
Return ff;

}

P void setFf(FormFile ff);

This.ff=ff;

}

}
```


Step3:- RT click on project name→ new→ other→ myeclipse →expand→ web-struts→ struts1.2 expand→ select struts1.2 Action→ next→ path(upload)→ Type(uploadAction) → name(frm1)→ uncheck valibleForm→ select forwardstab→ add button→name.(success)→ path(/success.jsp)→ add similarly failure.--> finish.

UPLOAD ACTION:-

```
Public class uploadAction extends Action.

{

Public AF execute(….)

{
```

uploadForm uf=(uploadForm)form;

FormFile f=uf.getFf();

Byte b[]=f.getFileData();

String fname=f.get filename()

fileoutputStream fas=new file outputstream fname);

fos.write(b);

return mapping.findforward(“success”);

}

}

Step4:- RT click on project name→ new→ JSP→ JSPname(uploading)→ok

<!—upload.jsp→

%@taglib uri=”http:// struts.apache.org/tags-html”prefix=”html” %

<html:form action=”upload.do” enctype=”multipart/form-data”>

Select a file: <html:file property="ff"/>

<html:submit value"UPLOAD"/>

<html:form>

Success.jsp:-

<% out.println("<h2>upload done</h2>");%>

Failure.jsp:-

<%out.println("<h2>upload is failed</h2>");%>

ADDING A SERVER IN IDE:-

Click on window menu→ preferences→ select my eclipseat leftside and expand→ expand application servers→ expand tomcat5.x→ at Rightside select enable→ a click on browse button→ and select tomcat home directory→ click on apply → click ok.

DEPLOYEMENT:- RTclick on projectname→ my eclipse → select add and remove project deployements→ select project→ addbutton-→ select server (tomcat5.x) → finish → (successfully develoyee)→ ok

START THE SERVER:- click on serversicon→ tomcat5.x→ start

BROSWER(SEND A REQUEST):- click on window menu→ show view→ web browser and type the following request.

http:// localhost:2013/uploadExample/upload.jsp.</

reset() of formBean class:-

→whenever deployee the app then the tiie container will be created of object of action servlet.

Whenever.object is created first constructed will be called

In input containe checkbox then use reset() is used default scope of FormBean is session. Use reset() if the FormBean is request no need to use reset()

Java compiler is write constructor not JVM.

RESET() OF FORMBEAN:-

In a formsBean class, the input values gien by a client will be stored. To store the client input setter methods of the calling setter methods, reset() will be called.

just before calling setter methods, reset() will be called.

Reset() is to initialize form bean variable with default values, before storing the new values into formBean variables.

If input page contains textbox(or) dropdown box (or) radio buttons..etc then all parameter will be send along with request form client side.

If we don't write reset() then setter methos will be called directly and previous values are overridden with the new values.

The problem aries, when input page contains checkbox parameter. The problem in if we don't select a check box then that parameter does not come to server. So it's related setter method will be not called. And hence the previous values will not be overridden with the new value. So the previously selected value remains same.

If the previous value is not overridden with new value then we will get unexpected output. So inorder to overcome this problem, we need reset() in the FormBean class.

Reset() is required when FormBean object in sessionscope and if the page containes checkbox.

EXCEPTION HANDLING IN STRUTS:-

➔ Exception are rised in Action class but not ActionForm becaz we are implementing b.login Action class.

➔ When we write throws Exception are can be handle that exception

➔ If really exception is raised in execute() of action class.

Public class restAction extends Action

{

P AF execute(----)

{

Try

{

}catch (Exception) {}

}}

To handle the best approach is declaration but not programmatic

NOTE:- while developing struts applications, at FormBean level, there is no chance of getting on exception. But there is a change of getting the errors while validating the user input.

In an Action class of struts application , there is a chance of getting an exception .becoz the Action class containes business logic.

if we do not apply exception handling in struts the exception stacktance information is printed on client browser,whenever an exception is occure but the clients are unkown about the technologies, so a client can not understand the problem and finally our project become failed.

In order to get out of the above problem,we need apply exception handling for an Action class apply exception handling for an Action class I struts application.

To handle an exception, we have the following two approaches.1) programmatically.by including try and catch blocks.

3) Declaratively by including <exception> tag in struts configuration file.

Action path=”/login name=”frm”,type=”Testapplication”>

<exception.key=”key” resource bundle

Type=”java.lang.Exception”

Local exception path=”/error.jsp”/>

<f…..>

<f…..>

</action>

<global-exception>

<exception key=”key”

Type=”……..”

Path=”/error.jsp”/>

</global-exception>

<action mapping>

If we have handle exception in programmatically then for each modification done on exception,we have to recompile,redeployee restart the server so this approach becomes complex.

In order to overcome the above problem we can use declaration exception handling.

In declarative exception handling, we can configure an exception either locally (or) globally.

If we configure an exception locally with in an action class then that exception handling will be done only for that one actionclass.

If we configure an exception globally it wil handle the exception related all action classes.

If an exception is occurred while executing an action then, the controller first verifies for local exception handling and it is not available check for global exception handling.

If an exception occurs then the error message should be taken from the resource bundle and we need to define an error separately and that error page displace error message by using<html:error/>

Ex:- <action path=”/login name=”frm1” type=”login Action”

<Exception key=”key type=”java.long.exception path =”/error.jsp”/>

<f------/>

<f------/>

</action>

If we want declare an exception as a globalexception then we need to configure under <global-exception>this tag must be written <action-mapping>

Ex:- <global-exception>

<exception key=”key”& type:java.long.exception”path=”/error.jsp”>

</global-exception>

GLOBAL-FORWARDS:- when multiple action classes in a struts application are sharing same out put pages then in stated of configuring then as local ,we can of configure as global.

If we want configure as global we need <global.forward>

For an action if both local and global forward are written then the preference will be given for the <local-forwards>

Ex:- <global-forward>

<forward name=”success” path=”/success.jsp”/>

<forward name=”failure” path=”/failure.jsp”/>

</global-forward>

ValidatorFramework in Struts:

→generally,before executing business logic for a given client request,we apply validations on the user input.if validations success then the business logic will be executed.

→In order to apply input validations,we can use any one of the following two approachs.

1)programatic approach.

2)Declarative approach.

→In progamatic approach, a programmer has to override validate() in the formbean class.

→In