Spring Material: Part-5

Spring Transaction

Project Architecture

➔Before go to the Spring project Architecture first we need to know about JavaEE Project Architecture and Struts Project Architecture.

Which are called as presentation tier component?

➔The component which is the part of the application that deals with client responsibility is managing and handling the request, these classes and the component are called presentation tier component.

Why Servlet called as Controller?

➔Whenever we send the request from end user servlet container will receive the request but servlet container will never act as a controller because he never manages to send a response back to the end user.
➔Servlet container will dispatch the request to the servlet component then servlet component will perform the operation.
➔Always servlet will not do the same operation it depends on request URL, by looking at URL servlet will do the operation.
➔Servlet will decide what it will do after getting the request from end user, so it is called decision maker so it is called as servlet.

Why JSP called as View Component?

➔JSP is managing to render the output to the client, apart from that he will not do any action so it is called view component.

Why we should not write Business Logic in Servlet?

➔If we write business logic in servlet then business logic will tightly coupled with presentation tier and it will not become reusable so code will become du, Why because we are not creating the object of servlet, if we create the object then we have to pass the HttpRequest and Response object which we can't pass because it is a technology specific component, so don't write business logic inside the servlet.

What is the problem we are going to face in Spring and struts frameworks if we write Business logic in controller class?

➔Presentation tier components are build in servlet and JSP's because they are responsible for end user responsibility.
➔ If build web application using spring framework, here also presentation tier is required for handling the client request.
➔One will be managing the view another will manage the decision making process.
➔In Spring JSP Page will act as a view component and controller class will act as controller.
➔When we are using struts JSP page will act as a view component and action class will act as decision maker controller.
➔That means the presentation tier components of your application will change on technology to technology.
➔When we will migrate our application one technology to another technology then Business logic will be lost.
➔So don't write business logic inside the controller.

What is POJO class?

➔The POJO class contains attribute with business methods, and the method most of the time will not have parameters.

Why we should not use Delegate and DAO as part of typical J2EE application?

➔In Typical J2EE web application Delegate and DAO will not be there,
➔ Most of the time we write business logic and persistency logic as part of POJO class only.
➔Our Application has not contain any complex business logic these are less complicated solutions so we have to use Servlet, JSP and POJO class we should not use Delegate and DAO class.

Why we should not use Delegate and DAO as part of typical J2EE application why spring and Struts came into picture?

➔Complex solutions can't be build using Servlet and JSP's.
➔Servlet and JSP's are not being designed to build better and bigger complex system.
➔These are designed to build simple requirements of an application that means receive the request read the data and delegate the request to the POJO class.
➔If I have complex requirement of the web tier of the application, and if I have 100's of pages are there in my application then don't go for servlet and JSP, because navigation complexity will not be there when I go for servlet and JSP's.
➔The pages and the servlet will become tightly coupled.
➔And we can't easily identify which servlet will forward the request to which JSP.
➔Suppose one servlet forward request to x.JSP but I want y.JSP then I have to change it to y.JSP but I can't find the servlet easily, I have to go with each and every servlet .
➔ So managing the navigability is not so easy.
➔If I go for a framework then it will become easy because everything will configure as part of properties file.

What is difference API and Framework? When we should go for Framework?

➔**API** (Application Programming Interface) **contains Abstract classes and Interfaces**, concrete class is not there in as part of API's so it is partial and it is not complete.

➔**API's are huge in nature that** means the numbers of components are more there as part of the API. And it not easy to learn it takes more time to learn.

➔**The worst part of API is the class that are there with an API are not easy to understand because these are interlinking with each other.** That means to learn one class we need to know some other class. Due to the classes are interlinking with other classes. For example to execute simply SQL query we have to write following 4 lines of the code.

➔Because of class are interlink with each other one has to understand all the classes that are provided with in that API, and **we can't start working with that without knowing complete knowledge of that API.**

➔**API will not provide Boilerplate code** *(ex: To fulfil some operation we may have write some code, the piece of logic that I am writing to fulfil that operation not only I any one has to write the same code for such type of requirement, such type of logic that we are trying to write is seems to be common on across all the project across world. This is called boilerplate logic)* so **programmer has to write Boilerplate Logic.**

➔If Boilerplate logic is not there in API so programmer has to write more lines of the code so the **time** required for developing the application is more, the **efforts** that we will be invest in developing the application is more, the **manpower required** to develop the application is more the **cost** involved in the application will be more.

➔As we write more number of lines of code the chances of **increasing bugs** as part of our lines of code will be more.

➔ As we write more number of lines of code the maintenance **time required to maintaining the application** is more and the **cost of maintaining** the application will be high.

➔ As we write more number of lines of code the time required for testing the application will be more, the amount of time we need to spend in insure the quality code is delivered out of our application will be more.

➔Considering all the things that if the API not provides the Boilerplate Logic all the above drawbacks that we are going to face if we are working with API so API's are lots of limitations are there.

Framework

➔Framework has provided a bunch of classes, spring framework developers has provided some classes for us that are concrete classes. And the classes that are provide by spring that anyone can directly instantiate the class object and can call the function, In case of API, API is partial and it is not complete.
➔The number classes that are provided by framework are less, and they design the classes in such a way that the minimal set of classes itself we can use to get the outcomes.
For example: if we want to write JDBCprogram we need minimum 7 classes while the same thing if we want to do in spring JDBCinstead of using Java JDBC then only one class we have to use and one method we need to call. That means the no of classes will be less.
➔The classes that provided by the framework are not interlink with each other.

➔ One can directly work with the specific class that means it is easy to understand the framework, when compared with API those are less complicated

➔And the amount of time that we need to learn the framework will be less and very easy to learn because interlink between classes are less.

➔We can learn a part of a framework that is required for fulfil our requirement. That means framework will support rapid application development.

➔Framework is providing concrete class and inside the concrete class boilerplate logic has been written by the developer itself. So programmer no needs to write boilerplate logic.

➔In this case we need to write less amount of lines of code, if we are writing less no of lines of code the time required for developing the application will be less.

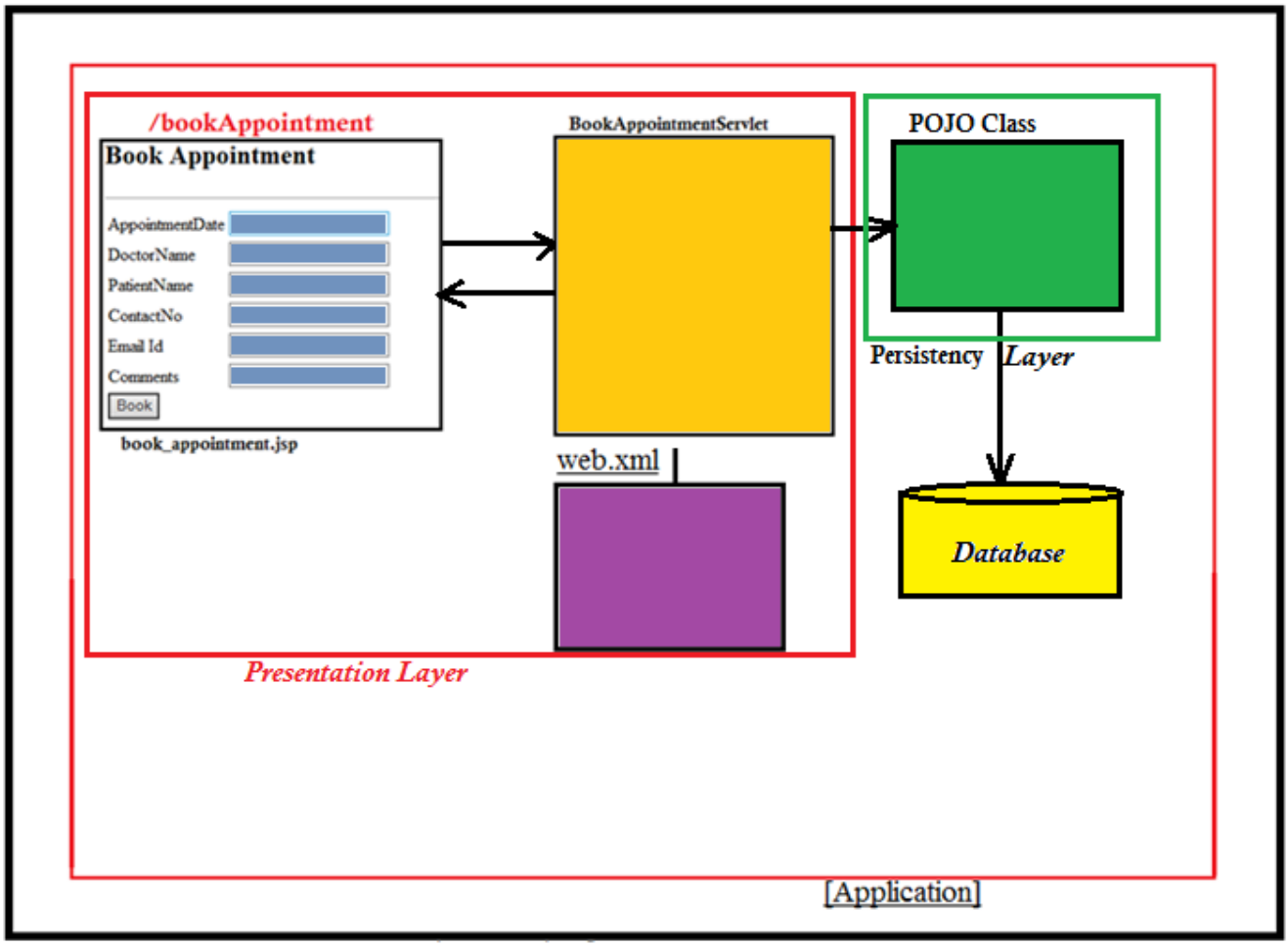
➔Cost involved in developing the application will be less, manpower required for the developing the application will become less.

➔Due to writing less no of lines of code the chances of bugs will be very very less.

➔The time required for testing the application will be less. When compared with API.

➔The logic that has been provided by framework people has pretested. That is completely free of bugs ,that means half of the application that we have developed is already pretested and many of the bugs has been avoided and quality code is delivered out of our application will be more.

J2EE Project Architecture



Servlet Container

➔Now when user send the request form book-appointment.JSP first it will be received by servlet container then servlet container goes to web.xml file to find the appropriate servlet for which request is coming for. Now first servlet container goes to <servlet-mapping> section in web.xml match the incoming request url with the configured url if matches then get the servlet-name from there and comes to <servlet> section match the servlet-name taken from servlet-mapping section to the servlet-name configured in the servlet section if matches then it get the servlet class to which container has to forward the request. In this way request to the appropriate servlet is mapped by servlet container.

➔We should never write the business operational logic or persistency logic in servlet. Let's see why we should not write persistency logic or business logic in presentation tyre component.

➔Let's suppose in above application we have used oracle database to persist the data and written the logic for doing this in our servlet itself.

➔After some time due to some reason if we want to use MS SQL Server or DB2 in place of oracle then to perform the persistent operation again we have to modify the logic in servlet according to database vendor which will be costlier.

➔Now we can see that changes made in persistent layer vendors is forcing us to make changes in presentation layer.

➔Similarly if we have to move from JDBCtechnology to Hibernate again we have to modify the code in servlet.

➔In this way we can say that writing the persistence logic in presentation tire component (Servlet) is very bad approach to develop an application.

➔It will make presentation tire tightly coupled with persistent tire.

➔Now let's suppose we are writing the business logic in servlet to develop the application.

➔After some period of time, we want to add some more business functionality in our application then again we have to modify the servlet.

➔In this way whenever business requirement change we have to modify the code in servlet. Here, also change required for business layer is affecting presentation layer.

➔So, we should not write the business logic in presentation tire component. It will also make presentation tire tightly coupled with business tire logic.

What are the drawbacks in developing the application using servlet architecture in a web application?

➔There will be multiple servlet and JSP. From each JSP whenever request comes to its respective servlet it brings some input data to the servlet then in servlet programmer has to write the logic for wrapping the data.

➔Wrapping of data means collecting the input data fromJSP and map it to an object which will be passed to business layer.

➔In this way we can see that for each and every servlet programmer has to write data wrapping logic which seems to be redundant process.

➔This is the biggest problem with servlet based application. To solve this problem we need some front controller which will perform the redundant process.

➔In case of servlet based application programmer has two choice one make one servlet as front controller and other use filter as front controller.

➔But, the problem is still programmer has to write the complicated wrapping logic / Form handling logic . To solve this major problem web application frameworks came into picture like Struts, JSF, and Spring MVC etc.

Request Wrapping BoilerPlate Logic & Form Handling Boiler Plate Logic

web.xml

```
<servlet>
	<servlet-name>BookAppointmentServlet</servlet-name>
	<servlet-class>com.bpl.pack1.BookAppointmentServlet</servlet-class>
	<init-param>
		<param-name>className</param-name>
		<param-value>com.bpl.pack1.Appointment</param-value>
	</init-param>
</servlet>
<servlet-mapping>
	<servlet-name>BookAppointmentServlet</servlet-name>
	<url-pattern>/bookApp</url-pattern>
</servlet-mapping>
```

```
public class Appointment {
	private Date appointmentDate;
	private String doctorName;
	private String patientName;
	private String contactNo;
	private String emailId;
	private String comments;

	//Setters & Getters
}
```

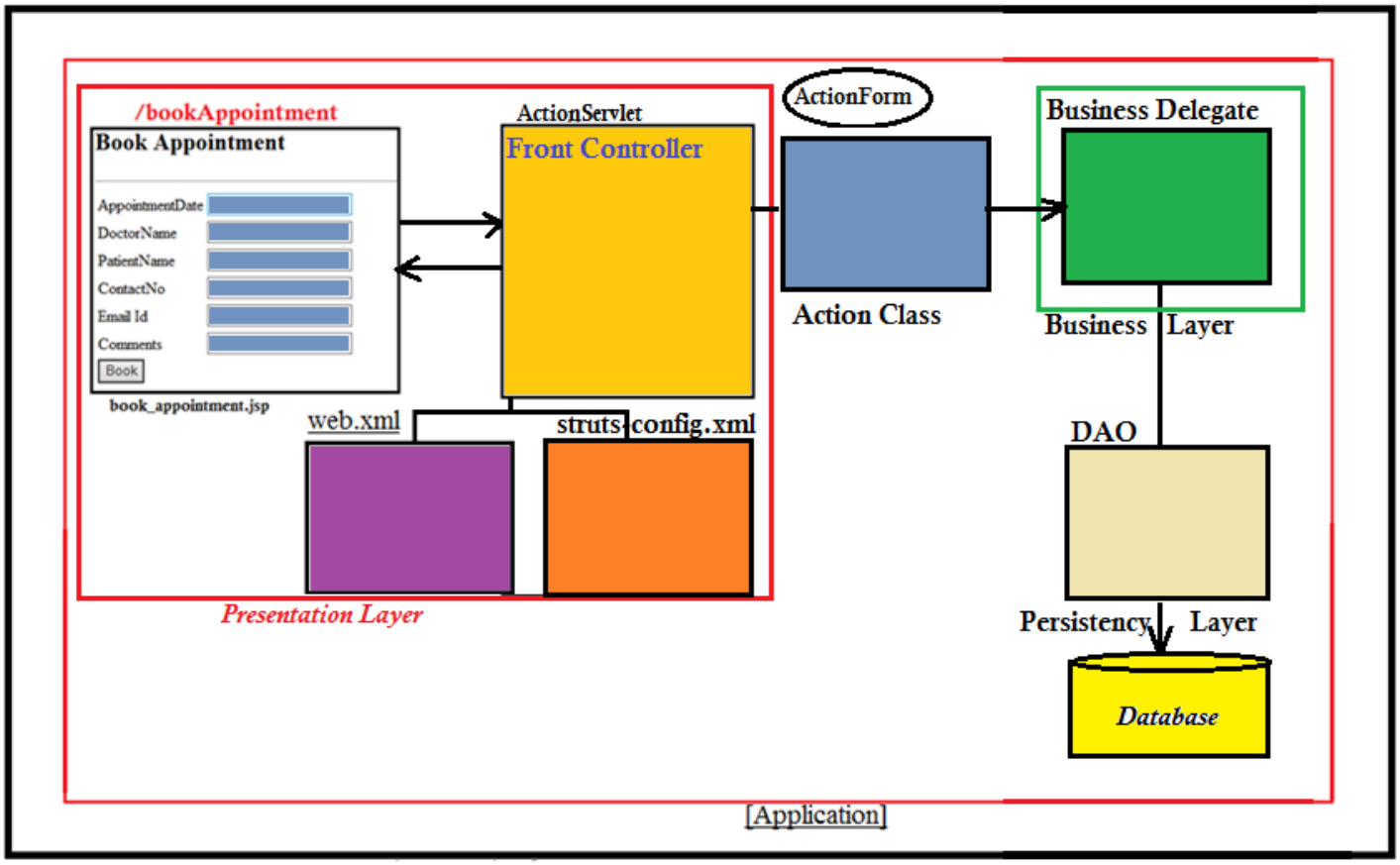

```
public class BookAppointmentServlet extends BaseServlet {
	private static final long serialVersionUID = 1L;

	protected void service(HttpServletRequest request,HttpServletResponse response)
					throws ServletException, IOException {
		Appointment appointment = (Appointment) getWrapRequestParam(request);
		boolean valid=validate(request);
		String page=null;
		if (valid) {
			page="SumbitedData.jsp";
		}else{
			page="bookApp.jsp";
		}
		request.setAttribute("data", appointment);
		request.getRequestDispatcher(page).forward(request,response);
	}
}
```

```
public abstract class BaseServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;
	private String className;
	public void init(ServletConfig config) throws ServletException {
		className = config.getInitParameter("className");
	}

	public Object getWrapRequestParam(HttpServletRequest request) {
		Object obj = null;
		try {
			Map<String, String[]> requestDataMap = request.getParameterMap();
			Class clazz = Class.forName(className);
			obj = clazz.newInstance();
			for (String key : requestDataMap.keySet()) {
				Method[] methods = clazz.getDeclaredMethods();
				for (Method method : methods) {
					if (method.getName().equalsIgnoreCase("set" + key)) {
						method.invoke(obj, requestDataMap.get(key)[0]);
					}
				}
			}
		} catch (Exception e) {
		}
		return obj;
	}

	public boolean validate(HttpServletRequest request) {
		Object obj = null;
		boolean flag = true;
		Map<String, String[]> requestDataMap = request.getParameterMap();
		List<String> errors = new ArrayList<String>();
		for (String key : requestDataMap.keySet()) {
			if (requestDataMap.get(key)[0].equals("")
					|| requestDataMap.get(key)[0] == null
					&& requestDataMap.get(key)[0].trim().length() <= 0) {
				errors.add("" + key + " is not Valid");
				request.setAttribute("errorData", errors);
				flag = false;
			}
		}
		return flag;
	}
}
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<html>
	<body>
		<p style="color: red">
		<c:forEach items="${errorData}" var="obj">
				${obj} <br>
		</c:forEach>
		</p>
		<form action="bookApp" method="get">
			<fieldset>
				<h2>Book Appointment</h2><hr/>
				<table >
					<tr>
						<td>AppointmentDate</td>
						<td><input type="text" name="appointmentDate" value="${data.getAppointmentDate()}"></td>
					</tr>
					<tr>
						<td>DoctorName</td>
						<td><input type="text" name="doctorName" value="${data.getDoctorName()}"></td>
					</tr>
					<tr>
						<td>PatientName</td>
						<td><input type="text" name="patientName" value="${data.getPatientName()}"></td>
					</tr>
					<tr>
						<td>ContactNo</td>
						<td><input type="text" name="contactNo" value="${data.getContactNo()}"></td>
					</tr>
					<tr>
						<td>Email Id</td>
						<td><input type="text" name="emailId" value="${data.getEmailId()}"></td>
					</tr>
					<tr>
						<td>Comments</td>
						<td><input type="text" name="comments" value="${data.getComments()}"></td>
					</tr>
					<tr >
						<td colspan="2"><input type="submit"  value="Book"></td>
					</tr>
				</table>

			</fieldset>
		</form>
	</body>
</html>
```

Let's understand the struts framework architecture

Struts Project Architecture



What is Front Controller? What is the purpose of front controller?

➔**Front controller is a design pattern.**

➔**Front controller acts as one entry point of our application.**

➔If I want have controller over the request that is coming into my application ,such type of controlling request can be happen if I have one single getaway.

➔ If we have front controller we can have better controller over managing the request.

➔If the entire request coming to one getaway then it is easy to apply any kind of services.

➔Like a validating the request can be applied at one place because all the request coming to the single entry point whether the request can be allow or should not be allow such kind of check or such kind of validate the easily acquired.

➔**Apart from that there could be some common mandatory system services will be there like form validating, request mapping etc… and plumbing services (optional) like audit will be managed by front controller.**

➔Rather than writing these logic in all the servlet Struts framework has provided a front controller to manage all the request.

Struts Application

Book Appointment

AppointmentDate
DoctorName
PatientName
ContactNo
Email Id
Comments
Book

"/book-appointment"

- Appointment Date is Blank
- Doctor Name is Blank
- Patient Name is Blank
- Contact No is Blank
- Email Id is Blank
- Comment Is Blank

Book Appointment

AppointmentDate
DoctorName
PatientName
ContactNo
Email Id
Comments
Book

Data Received

AppointmentDate :	12/05/2017
DoctorName :	Sriman
PatientName :	Dhananjaya
ContactNo :	9437215211
Email Id :	javatech.dj@gmail.com
Comments :	Heart problem

ActionClass

```
public class BookAppointmentAction  extends org.apache.struts.action.Action{
        private final static String APP_RESERVE= "success";
        @Override
        public ActionForward execute(ActionMapping mapping, ActionForm form,HttpServletRequest request,
                                        HttpServletResponse response) throws Exception {

                        return mapping.findForward(APP_RESERVE);
        }
}
```

```
public class AppointmentActionForm  extends org.apache.struts.action.ActionForm {
        private static final long serialVersionUID = -1150580272534044855L;
        private String appointmentDate;
        private String doctorName;
        private String patientName;
        private String contactNo;
        private String emailId;
        private String comments;

        //setters & getters

    public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
      ActionErrors errors = new ActionErrors();
      if (appointmentDate== null || appointmentDate.trim().equals("")) {
        errors.add("appointmentDate", new ActionMessage("error.appointmentDate.required"));
      }
      if (doctorName == null || doctorName.trim().equals("")) {
        errors.add("doctorName", new ActionMessage("error.doctorName.required"));
      }
      if (patientName == null || patientName.trim().equals("")) {
        errors.add("patientName", new ActionMessage("error.patientName.required"));
      }
      if (contactNo == null || contactNo.trim().equals("")) {
        errors.add("contactNo", new ActionMessage("error.contactNo.required"));
      }
      if (emailId == null || emailId.trim().equals("")) {
        errors.add("emailId", new ActionMessage("error.emailId.required"));
      }
      if (comments == null || comments.trim().equals("")) {
        errors.add("comments", new ActionMessage("error.comments.required"));
      }
      return errors;
    }
}
```

ApplicationResource.properties

```
# -- standard errors --
errors.header=<UL>
errors.prefix=<LI>
errors.suffix=</LI>
errors.footer=</UL>
# -- validator --
errors.invalid={0} is invalid.
errors.maxlength={0} can not be greater than {1} characters.
errors.minlength={0} can not be less than {1} characters.
errors.range={0} is not in the range {1} through {2}.
errors.required={0} is required.
errors.byte={0} must be an byte.
errors.date={0} is not a date.
errors.double={0} must be an double.
errors.float={0} must be an float.
errors.integer={0} must be an integer.
errors.long={0} must be an long.
errors.short={0} must be an short.
errors.creditcard={0} is not a valid credit card number.
errors.email={0} is an invalid e-mail address.
# -- other --
errors.cancel=Operation cancelled.
errors.detail={0}
errors.general=The process did not complete. Details should follow.
errors.token=Request could not be completed. Operation is not in sequence.
# -- welcome --
welcome.title=Struts Application
welcome.heading=Struts Applications in Netbeans!
welcome.message=It's easy to create Struts applications with NetBeans.

error.appointmentDate.required=Appointment Date is Blank
error.doctorName.required=Doctor Name is Blank
error.patientName.required=Patient Name is Blank
error.contactNo.required=Contact No is Blank
error.emailId.required=Email Id is Blank
error.comments.required=Comment Is Blank
```

struts-config.xml

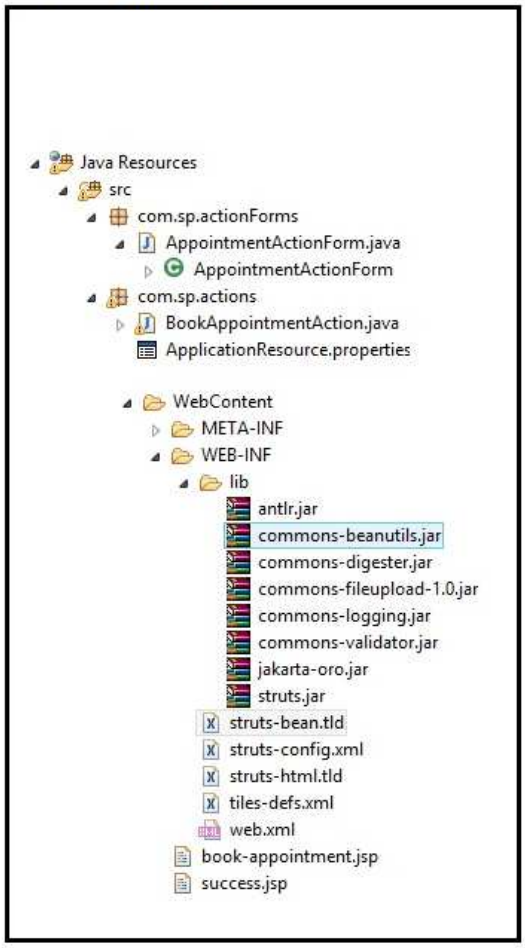
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts-config PUBLIC
      "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
      "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">
<struts-config>
       <form-beans>
                  <form-bean name="AppointmentActionForm" type="com.sp.actionForms.AppointmentActionForm" />
       </form-beans>

       <action-mappings>
              <action input="/book-appointment.jsp" name="AppointmentActionForm" path="/book-appointment"
                          scope="request" validate="true" type="com.sp.actions.BookAppointmentAction" >
                          <forward name="success" path="/success.jsp" />
              </action>
       </action-mappings>

   <controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>
       <message-resources parameter="com/sp/actions/ApplicationResource"/>


       <plug-in className="org.apache.struts.tiles.TilesPlugin" >
     <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />
     <set-property property="moduleAware" value="true" />
   </plug-in>

</struts-config>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://java.sun.com/xml/ns/javaee"
              xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
              id="WebApp_ID" version="3.0">
 <display-name>StrutsProject</display-name>
       <servlet>
          <servlet-name>action</servlet-name>
          <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
          <init-param>
                    <param-name>config</param-name>
                    <param-value>/WEB-INF/struts-config.xml</param-value>
          </init-param>
           <load-on-startup>1</load-on-startup>
        </servlet>
        <servlet-mapping>
            <servlet-name>action</servlet-name>
            <url-pattern>*.do</url-pattern>
        </servlet-mapping>

         <jsp-config>
    <taglib>
      <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
    </taglib>
    </jsp-config>
</web-app>
```

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
<head>
<title>Login Page</title>
</head>
<body>
  <div style="color:red">
       <html:errors/>
  </div>
 <html:form action="/book-appointment.do" method="get" >
   <fieldset>
     <h2>Book Appointment</h2><hr/>
       <table >
         <tr>
           <td>AppointmentDate</td>
           <td><html:text name="AppointmentActionForm" property="appointmentDate" /></td>
         </tr>
         <tr>
           <td>DoctorName</td>
           <td><html:text name="AppointmentActionForm" property="doctorName" /></td>
         </tr>
         <tr>
           <td>PatientName</td>
           <td><html:text name="AppointmentActionForm" property="patientName" /></td>
        </tr>
        <tr>
           <td>ContactNo</td>
           <td><html:text name="AppointmentActionForm" property="contactNo" /></td>
        </tr>
        <tr>
           <td>Email Id</td>
           <td><html:text name="AppointmentActionForm" property="emailId" /></td>
         </tr>
         <tr>
           <td>Comments</td>
           <td><html:text name="AppointmentActionForm" property="comments" /></td>
          </tr>
          <tr >
            <td colspan="2"><html:submit value="Book" /></td>
          </tr>
       </table>
    </fieldset>
  </html:form>
</body>
</html>
```

Project Structure



➔In case of struts view (client) will be JSP or Html whenever user send a request it comes to servlet container then servlet container goes to deployment descriptor file (web.xml) and match the incoming url to the configured url-pattern.

➔In web.xml url pattern for ActionServlet is *.do which means for any url request ends with .do will match with the url of ActionServlet. So, in this case for every request ending with .do will be forwarded to ActionServlet.

➔Now ActionServlet will perform the request wrapping process and forward the request to Action class by calling execute (req, resp, actionForm, mappings) method. Let's understand how ActionServlet perform this operation.

➔ActionServlet is struts framework in-built class which acts as front controller. In ActionServlet logic for wrapping the data coming from JSP to ActionForm is written.

➔The wrapping logic is generic it means it is not specific to the ActionForm or any input control. When ActionServlet receives a request it collects the input data and goes to struts-config.xml file search for the Action class to which request is coming based on the path configured for the Action class by matching path and URL inside <action-mappings> Section.

➔After getting the Action class it search for the name attributes in <action-mappings> section if there is any name it gets it and go to <form-beans> section find the actual ActionForm which will be sent to the Action class based on that name.

➔Now after getting the ActionForm ActionServlet will create the object of ActionForm and populate the collected input data from JSP to the ActionForm class. If any validation over data is required then ActionServlet will call the validate method of ActionForm. After performing the

validation ActionServlet will call execute method of Action class by passing request, response, actionForm & mappings as input to perform the requested operation.

➔From the above explanation we come to know how ActionServlet Identify the appropriate Action class and ActionForm. But, the point is how ActionServlet knows to validate the data it has to call validate() method and to pass the actionForm it has to call execute(-,-,-,-) method of Action class.

➔To solve this problem struts has recommended the programmer to not write any class as Action class or ActionForm. To make a class as ActionForm programmer must have to extend from ActionForm class and to make a class as Action class they must have to extend the class from Action class.

➔ When we write BookAppointmentAction and AppointmentActionForm by extending from Action and ActionForm class we are overriding the execute() method and validate() method.

➔Which is coming from Struts components that's why both execute() and validate() are known to ActionServlet. In this way ActionServlet performs common system processing logic(Data wrapping) and plumbing logic.

What we go for DAO? When can be called a class as a DAO?

➔If a class only include data access specific logic of the persistency service that's the regions the class is been called as DAO access object.

➔This is the class of the object of the class will helps you in accessing the data of the database that's why this is been called as DAO class.

What's as a DAO will help you, what is the role of a DAO, how is it going to helps you, what's the participating of DAO within the Application?

➔There are two regions that are why we are using DAO in our applications.

➔DAO acts as an Abstraction layer of the persistency tire of our application, none of the classes with in our application will really bother about persistency requirement of the application. Only just bother about talking to the DAO .The rest of the information will provided the DAO .So it is providing the abstraction layer of the persistency information.

➔DAO acts as protect the rest of the classes with in your application .Any changes happen in the persistency tire of the application only has to reflected as part of the DAO. Most of your classes with in the application will not affected.

Benefits go for DAO

➔DAO will help you to managing in switching from one persistency technology to another persistency technology.

➔If you switch to one persistency vender to another persistency vender also the logically DAO only get impacted the rest of the class are not impacted.

➔If you are using DAO none of the classes know where is the source system is there the data is coming from, none of the classes know the representation of the underline source system. So we are completely decoupled to the source system. So write your persistency logic DAO layer only.

Why we use Delegate?

1st reasons to go for delegate

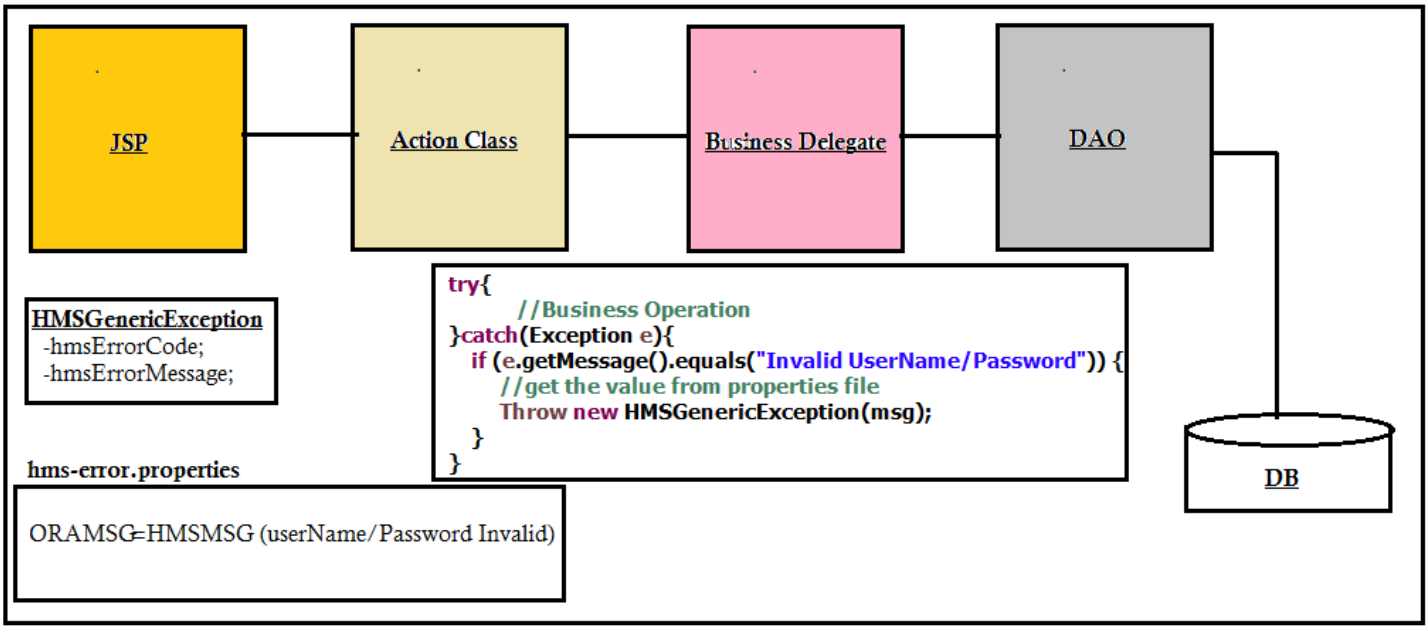
➔Business Delegate helps us to manage the Exception within of our application, let's see the how we can manage the Exception in our application.

➔There are two ways are there because it is related to how granular we want to represent error details to the end user.

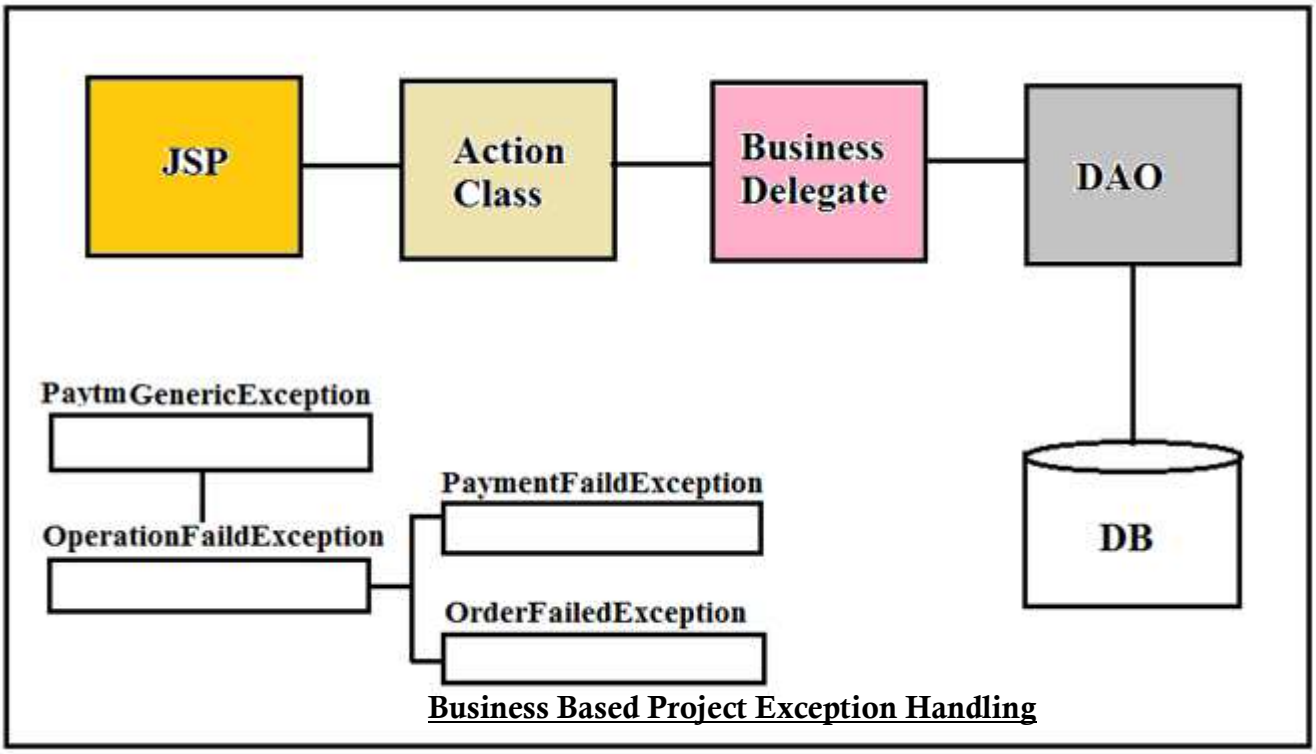
1st Way

➔We want to tell the root cause of the failure reported to the end user.

➔To identify the root cause failure that I want to report the exact reason for the failure then we need to grab the technical error information about that failure as specially we do such kind of things when we doing when the end user is it specialist who aware about the technology.



2nd Way Product Based project Exception Handling



Business Based Project Exception Handling

How you validate the Application?

➔When we submitted the request from JSP page Action class will get the request then Action class pass the data to business delegate, before passing the data to business delegate we have to ensure every piece of information that is being sent as a input to the business tier of our application is seems to be appropriate.
➔If we send the invalid data to the business delegate then there is a chance that incorrect details being passed as an input to the business delegate so first we have to validate the data.

➔There are two ways we can validate the data that is being received from the client.
i) Server side validation **ii)** Client side validation

➔ It is depends on project requirement which side validation is to be used.

What are the benefits you are going to get if you go for client side validation?

➔Unless until the data that the user is entered is valid, request will not send to the server.

➔Assume I did a client side validation. I wrote java script validation as part of the JSP page. Then the user had enter a wrong data and click on submit then the request will not goes to the server because the data is validate in the client side.

➔ As we validate so the bandwidth space will not waste and network traffic will not increase upon resending the request to the server. That's why we go for java script client side validation.
When to go for Server side validation?

➔Assume I entered the data in the Rest Easy Client or in Postman or I write manually the http program which will directly send the request to the server then there is no client side validation.

➔As we didn't write any server side validation we will get exception as output. To bring our server down a hacker can write a program which is spans 1000 of threads in sending the parallel http request to your server then our server will brought down.

➔ So if your application is design and is being used by the whole world then instead of bandwidth and network traffic the security is more important.

➔So in this scenario I need to use server side validation. I should not use client side validations.

➔In an internet site where the people are going to be the end user then having the client side validation is not safe so here we use server side validation. That's why struts has provided validate class.

➔If it is an intranet network application then the employees of the organization is going to use it. It is hosted on intranet.

➔ So there is no chance that outside people will send the request. So here we don't want to use server side validation. Because no one wants to break their application because they have only to suffer for this.

➔So they looks for properly using the application, In such a case we will go and use the client side validation rather than going for server side validation.

How are you handling the exceptions as part of your project?

➔While designing the application itself whatever the data that flows as an input into the application from to the presentation tier to the business tier we ensure that the data is much validate at the presentation tier .

➔We are using server side validation as part of our application to ensure what makes the data in which way it is passed as input to our application rather user always such data is being filtered and such data is being handled and such data will be validated to ensure that every data that comes as an input to your application is validated.

➔Unless until the data is valid it will not enter into the application.

➔If the data is invalid we report the data in an appropriate way as a converting them into a validation error to the end user.

➔SO that there will not be any chance that my application is going to run into exception due to the incorrect data is handled as part of my application.

➔Once the data will received by the action class then the action class is validate the data before it will pass to the business delegate of my application so that we can avoid the exception as much as possible.

➔ But still there may be a chance of arise the exception like payment gateway exception or while parsing the xml which is checked exceptions. So we must have to handle them.

➔So therefore we have hierarchy of Exception being built as part our application to report appropriate failure information to the end user.

➔ **There is a base Exception class so all the Exception that are there with in our application that we used to hold for reporting the exception**

➔And for every module we are going to break up into separate Exception, where the module specific error information can be reported.

➔Within the module there are several different operations we are going to perform so representing each and every operation we are creating separate exception class these are called module specific exception.

➔And based on the exception that we will get we are translated in to friendly error information.

➔ Every other class within our application which ever reporting the exception will be caught at the business delegate & business delegate will translate it from technology specific exception into application specific exception and are reported to my presentation tier components.

➔ Now my action class will handle that application specific exception & will converts into friendly error message & display it on error page to the end-user.
➔If any Runtime Exception came the we have to handle them through by configuring Error page within our web.xml file

2nd reasons to go for delegate

➔When the client submit the request from the JSP page then the data is received by action class. The data submitted by the JSP page is received by the action class in the terms of action form.

What is action form & what is the purpose of action form?

➔Action form will be act as form carrying object. Action form contains attributes & it depends on number of input data we are sending from the JSP page.

➔Every attributes that we are declared in the action form has to be in String form.

➔Because when we submitting the request from the JSP page the action servlet always received the request parameter values in the form of String. Whatever the primitive data type we are sending from the JSP page it is always received in the form of String.

➔Then after receiving the primitive data it converts these data into their corresponding data types. While converting the data from string to the corresponding data type of the attribute there may be a chance of occurring the exception.

➔Assume while converting the string data into integer data type you face the number format exception. Because we can't convert the string value like ("abcd") into integer.

➔Then the action servlet will get the exception & it will not do anything it simply throw as an output to the user.

➔That means struts will not gracefully handle such kind of failure in converting the string represented value into the attribute types of my action form. So there is a chance of my application will get runtime exception.

➔So to ensure that you always be safe in receiving the data always it recommended to declare the attribute type as String only.So every attribute you declare in the action form has to be string only so that even the incorrect data is being submitted also can be received as part of the string attributes.

➔Once the data is mapped or wrapped in the action form immediately the action Servlet will call the validate () method.

➔Validate() method will validate the data means is the data is rely an integer type or not if it is not an integer add it as an error & return the error saying the input data is not valid.

➔So now when the request submitted from the JSP page the action servlet class will receive the request in the terms of action form. But my action class didn't not perform business operation or persistency operation because if my action class will do this operation then my business logic & persistency logic will tightly coupled with presentation tier technology.

➔So the business logic will be there in business delegate. So whatever the data is received by my action class has to pass the data as input to the business delegate .

➔But the data is with there in my action form should not be passed as input to business delegate. Because the action form is a struts specific component.

➔Every action form should extend from struts action form. If you directly pass the action form as input to the business delegate then the logic has been written inside the business delegate has to be modified when we switch from struts to spring. Because if we remove the jars these classes will not work or compiled.

➔So my action class should never pass the action form as an input because my business tier components will get tightly coupled with my presentation tire components technology. That's why Value object comes into picture.

Why we use value object ? What is the purpose of the value object?

➔The value object is just like java bean. It is a pojo class .
➔It is used for carrying the values between presentation tier to business tier components of my application.

What is there inside the value object ? What does the value object contains ?

➔Value object will contains the attributes in which we wanted to populate the data that is being received by the client and we want to send it as input to the business deligate.

➔The number of attributes that are there inside the value object is exactly same as the attributes present inside the action form which is send by the client to the action form.

➔The attributes that are there inside the value object and the data types of these attributes will be again string only.

➔So action class is receiving the data in the terms of action form. Action class should not convert the same data in the format of which my business deligate requires.

➔Assume tomorrow my business deligate wants long data type instead of int. Then if there is a change in the way business deligate wants the data to perform the business logic then my application is going to impacted.

➔If there is a change in presentation tier then I can change the action class . But the change is there in business tier even then also I will be impacted.

➔So my action class should not convert the presentation tier specific format of the values in the business tier to perform the business logic .

➔It should pass the same presentation tier values as input only. That means the value object is presentation tier values represented component.

➔Suppose tomorrow my database column has been changed from integer to float then I have to change my action class, because am converting into your format.

➔ Whets comes as an input the same value is going to pass as an input to business deligate otherwise if there is any change in data in business deligate then I have to change it in action class because I am taking the responsibility to convert. So I should not convert.

➔So value object will contain the same values only that is being pass as an input to the business deligate.

➔Then the business deligate has to pass this data as input to the DAO. Then the DAO will stores the data into the data base.

➔But the data inside the value object which is passed by the business deligate to DAO to perform persistency operation is not in form of persistency tire representation format. The values that are there inside the value object is in presentation tier format.

➔All the values are inside the value object is in the string form. If the database wants util.date then DAO will not convert it.

➔If the business deligate wants the sSQL date() then he has to send the data in the form of SQL date ().

➔SO the business deligate should not pass the value object to the DAO. Because the DAO didn't know how to convert the presentation tier values to the persistency tire values.

➔That's why the business deligate has to take the responsibility to converting the value object data into business object that should pass as an input to the DAO. That's why the business object is come into picture.

3rd reasons to go for delegate

➔When request is coming the request is received by Action class,Action class will delegate the request to Business Delegate then Business Delegate will perform the business operations .Upon completing the operation it persist the data from DAO.

➔Does the Business delegate will talk to one DAO or chance that some specific use cases a business delegate may have to talk to Multiple DAO also?

➔Always a delegate will not talk to one dao rather delegate will talk to multiple dao also.

➔If my delegate is trying to work with multiple dao then how to i need to manage the transactionality as part my application.

➔DAO will not create a connection or commit or roll back the connection.
Persistency operations that are there in several DAO with the same business functionality will not committed in one shot.

➔Then who has to managing the transactionality, Who has to create the connection ?Who has to pass the connection as a input ? Delegate Who has to use the connection that has pass ?
➔Dao,so every DAO should to create a connection .DAO use the connection thats is pass by business delegate .

➔Once both the DAO successively completed the operation and return the control to the business delegate the delegate will commit the connection and close the connection.

➔If any of the DAO are not able to perform the operation then what are they going to do?

➔Then delegate throws an exception then delegate will ask to to the transaction to roll back And translate the technology specific exception to business application specific exception and throws to the Action class. So the business Delegate managing the transactinality .This is the 3rd resigns to go with business delegate.

How many DAO delegate are there in your application?

➔In my application there are multiple business delegate are there.

➔In every operation we are not creating one one business delegate.

➔AddDoctor business delegate deleteDoctor business delegate update business delegate for every operation we are not creating one one business delegate.

➔In my application multiple DAO also there .For every table we are not creating one one DAO.

➔The related group of table that are connecting with a relationship will be created with one table . User profile and logging credentials are related User table that's why for these table we are creating one DAO to handle the database operation for the related group of table .Thats means within our application we never create one Dao with one table with group of related table with in the database we are creating one DAO.

Then How many delegate are there with in the Application?

➔There may be multiple delegate are there .Depending upon the use case we are dividing the delegate.

➔One high level use case we break then into multiple use cases that is called low level use case document.

➔For every use case in our application we are come with one delegate .Because delegate is going to handle the business functionality.

➔For related business functionality we are come with one delegate .

➔So never create the delegate like addDocuerDeligate delete Doctor Delegate for the whole doctor related functionality in our application .

➔Just write DoctorDelgate is the delegate .Inside it addDocuer, deleteDoctor, UpdateDocter in such a way multiple method will represent each and every functionality within the delegate .

➔So delegate will identify base on use cases DAO will indentified with the related operation .Thats why we break in to the delegate and DAO in our application.

Why Action Class should not talk to the DAO?

➔In our application JSP always act as a view component.

➔When we submit the request from JSP page Action class will receive the request.

➔Action class act as a command class it delegates the request to business delegate then delegate has to call the DAO because business delegate will not perform the persistency operation that's why DAO comes into picture.

➔Our Action class should not talk to the DAO.

➔Because if our Action directly talk to the DAO, then DAO performing the persistency operation it might run into Exception that will be reported the DAO.

➔Then DAO has to throw the same Exception back to the action class.

➔Then the action class will receive the SQLException.

➔SQLException will not carry the error information neutral to the database rather it is specific to the database.
➔If ActionClass has to identify the root cause of the failure by looking at the error information which is database specific error.

➔Our Action Class directly being expose to the SQLException ,if it is directly reading the SQLException where upon changing the database oracle to mysql then the Exception information that directly expose to Action class will get change as it is database specific.
➔So the Logic that is written in Action Class has to be modified.

➔JSP and Action class is presentation tier component, if the change in persistency tier will impact presentation tier component which ends up with a saviour maintenance cost.

➔Almost every class within our application has to be rewritten, maintenance cost of maintaining the application will be high.

➔Such changes that are being happened in the application will makes our classes fragile difficult to manage, difficult to debug.

➔So Action Class should talk to DAO.

Application Flow

➔In our application JSP always act as a view component.
➔When we submit the request from JSP page Action class will receive the request as part of Application Form.
➔When Action class will receive the request and going to wrap into value object and passes the value object to the business delegate.
➔Then Business delegate will populate the data into the business object compute the data then perform business logic and passes the data as an input to the DAO.
➔None of the DAO will participate under the transaction. Managing the transaction and rollback the transaction will be manage by business delegate.
➔Because any of the DAO are throwing an Exception that will be handle and converting in to Application specific Exception and return back to Action class which is presentation tier component.
➔Action Class convert the Application specific Exception into end-user friendly message and displayed the output to the end-user.

About Business Delegate

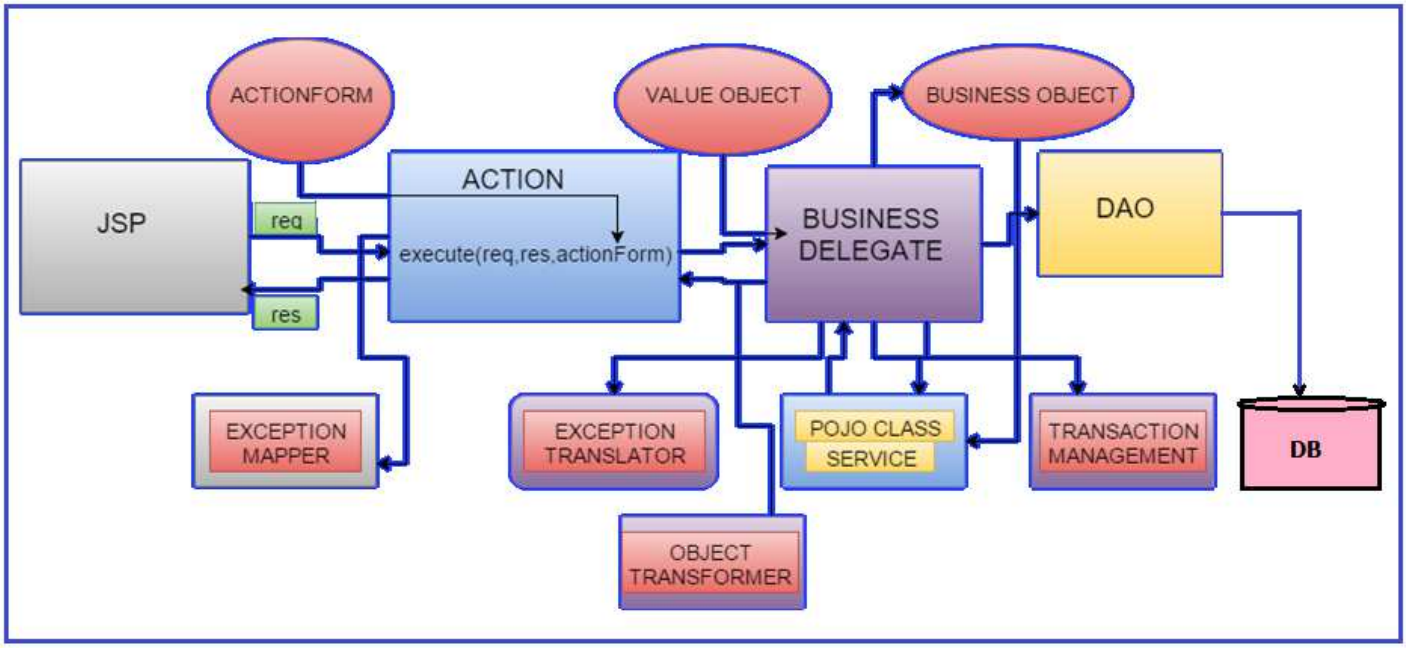
➔Business delegate should never carry any operation.
➔Business delegate always should delegate only.
➔But in our Application Business delegate carry the operation also.
➔Which is not seems to be appropriate ,so we should not call it as a business delegate.
➔ Because it is not performing the business delegate responsibility rather it is performing the operation also.

If interviewer will ask you it is not performing the business delegate responsibility rather it is performing the operation why you call it as business delegate?

.
➔Then the architectural point of view even though we are taking it as delegate but design pattern perspective is no more being called as delegate.

➔As per our architect decision we design our application in such kind of design may be the business logic that we are writing as part of the application to be less complicated or could be manageable even though we don't separate it separately.
➔That could be one reason why my architect has chosen to have business delegate to carry the business operation also.
➔But definitely I could not encourage them to recommend to have the architecture in this way. I do understand.

Architecture of Struts Framework



➔In the above project JSP is acting as a view. When user sends a request it is received by Action class. All the input control sent by user through JSP comes to Action class in form of action Form.

➔In Action Form class there will be attributes and the number of attributes is same as the number of input controls coming from JSP. Generally the types of attributes in Action Form will be String only.

➔ Action Form class attributes may be of any type but if user send String type value for Integer type like "abc" for int age then during type casting ActionServlet will not handle this request gracefully.

➔It will send an ugly error page to the end user because of this it is recommended to take attribute type as String only.

➔ Now after receiving the request by Action class it will populate the action Form data to Value Object and then call the Business Delegate by passing value object. Value Object is a pojo class which will contain same number of attribute as there in Action Form and types of the attribute is String only.

➔ VO is containing same data as there in Action Form then why Action class is populating the ActionForm data to Value Object and call the Business Delegate by passing Value Object. Why Action class does not pass action Form directly to the Business Delegate.

➔To find the answer let's suppose Action class calls the Business delegate by passing action Form. After some period of time we do not want to use struts as presentation layer then to get Business layer work properly we have to modify the business logic which are using action Form because Action Form is struts specific component.

➔This is the reason we should not directly pass action class instead of it we should wrap it in Value Object and pass it to Business Delegate because even if we want to remove struts as presentation layer then

also our business layer will work properly. It means if we pass action Form to Business Delegate then business layer will get tightly coupled with presentation layer.

➔ Value Object contain the data in presentation tire format.

➔So, Business delegate will call the Object Transformer to convert the value Object into Business object. Business Delegate itself will not convert Value Object to Business Object because delegators are not meant for doing. It just delegates the request to appropriate component to perform the operation.

➔After getting the Business Object Business Delegate will call the Service class by passing Business Object as input to perform the business operation.

➔Service class is just a pojo class which will contain the logic for performing the complex business operation.

➔After performing the business operation Business Delegate will call the DAO to perform the persistent operation by passing Business object or entity.

➔DAO is a class which will contain only persistent logic only.

➔Some time while performing the persistent operation DAO will raise an exception then Business Delegate will catch it and pass this exception to Exception Translator which will translate the exception to Application specific exception or End-User friendly Exception.

➔Application Specific Exception are tied with error code and error description which will be understand by technical team who are using this application.

➔Now after translating the Exception Business Delegate will give it to Action class now Action class will pass this to Exception Mapper to map it to appropriate Exception based on that Action class will give response back to the end user as friendly error message.

➔Let's suppose there are two DAO1 and DAO2. Now DAO1 is performing the operation of Saving the Employee in data base with data as Emp_ID= "E1", fName="Dhananjaya" after performing the operation DAO1 has commit it.

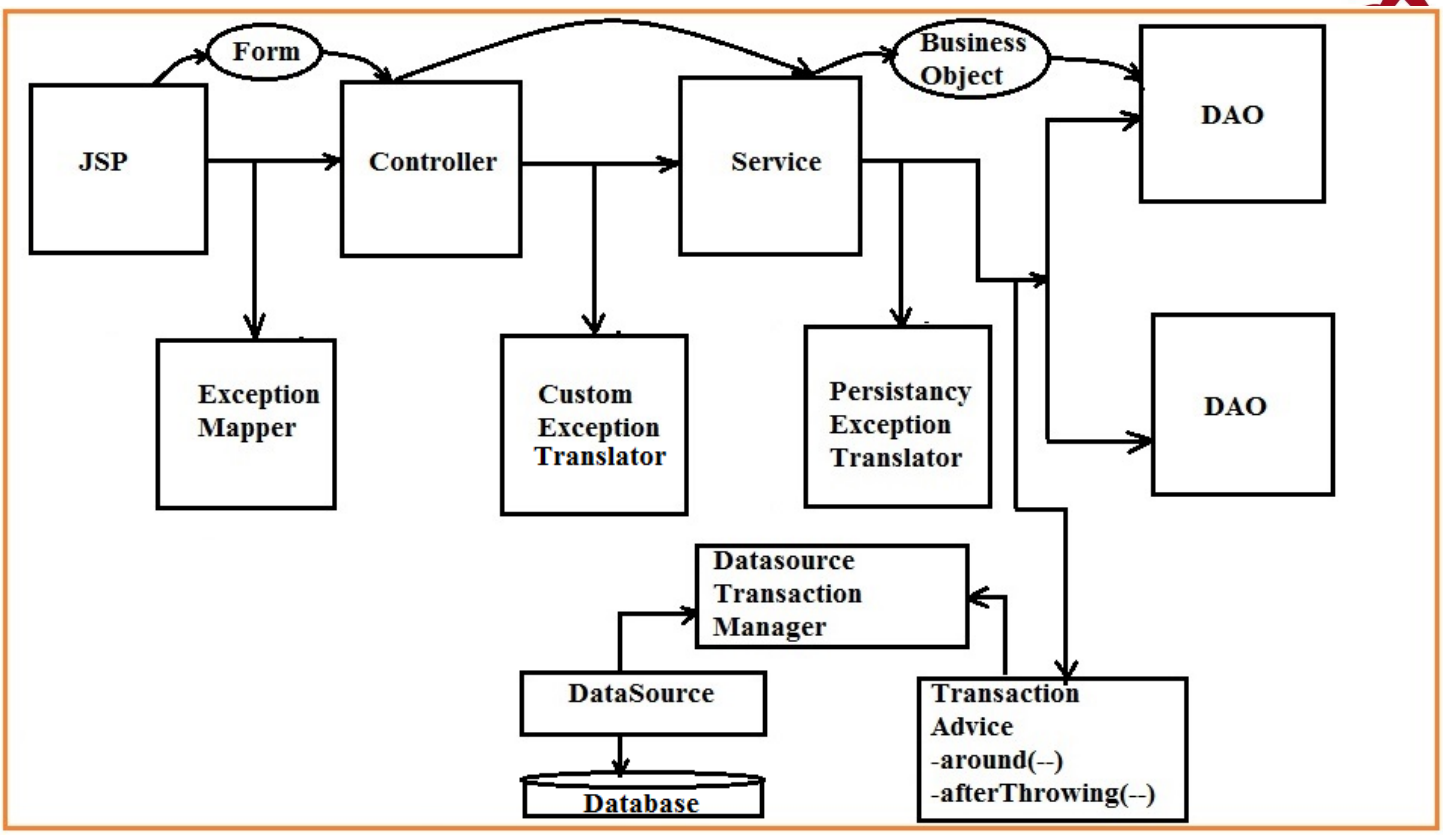
➔Now DAO2 starts performing the operation of saving the Employee to the Project based on their id now suppose there is already an employee with Emp_Id=E1 on Project_ID=P1 in this case DAO2 will raise an exception and it will rollback but this rollback will not applied on DAO1 so Emp_ID=E1 will be there in Data base.

➔In this situation we can see that if we perform transaction on DAO level then it will become individual to the DAO it will not become an application level transaction. That's why instead of doing the commit or roll back on individual DAO do it in Transaction Manager component which will be called by Business Delegate after Performing each and every DAO operation so commit and roll back will be applied at once.

➔If everything goes fine then Business Delegate will return appropriate response to Action and Action will give expected response back to the user.

➔In this way all the components in Struts based project going to communicate with each other and perform the operation while receiving a request. All the component shown in the above diagram need not to be in each and every strut based project. It depends on the requirement but a standard struts project should contain all those component.

Spring Architecture



.

Why we have a service why we don't have a business delegate?

➔When I submitted the request from the JSP page. Controller will receive the request. Then controller will call service, then business delegate will call DAO & DAO is performing the persistency operation.

➔DAO is performing the operation using Spring JDBC. Instead of using Spring JDBCI use Hibernate. Hibernate will talk to the database. Now the database is run in to an error. Then the data base is report that error to hibernate.

➔Hibernate will throws the exception to DAO. My DAO will get hibernate exception. So the exceptions which are reported by the underlying database is depends upon the technology we are using to performing the persistency operation.

➔If DAO is use spring JDBC to persist the data then it will get Data Access Exception. If DAO is using hibernate then it will get Hibernate Exception.

➔So the logic we have written in the exception mapper to mapping the exception is depends upon persistency tier technology.

➔So if persistency tier technology is throws different exception then the logic we have write to mapping the exception has to be modified. So my persistency tier will be impacted.

➔So my service should not throw the exception. It has to catch the exception & should convert into application specific exception and should throw it to the controller.

➔Now my controller has to identify the exception that means the application specific exception is come to exception mapper.

➔The mapper will identify the exception and will see which controller is throwing it. Based on it mapper will converts into business friendly error message and send it to the end user.

What is Exception Mapper?

➔Exception mapper converts the end-user friendly error message.
➔Data AccessException or Application specific exception and controller are the input to the Exception Mapper.
➔Controller is required because exception mapper has to know under which controller operation the exception has been raised.
➔Exception mapper has to know it then only it map in to appropriate friendly error message.

What is persistency exception translator?

➔The exception which is reported by the DAO is always depends on the technology which we are using to persist the data.

➔If we using spring JDBC then we will get Data Access Exception. If we use Hibernate we will get Hibernate Exception, If we use java JDBC we will get SQL Exception.

➔So in persistency exception translator we will configure the exceptions which are reported by the DAO to Data Access Exception.

➔ So that all the technology specific exception is thrown by the DAO is converted into DataAccessException or whatever the persistency technology we use to persist we always get the same Data Access exception so we never need to modify our logic.

➔So ultimately the exception has been reported we will get DataAccessException which is unchecked exception.

➔ Non of the classes will catch , none of the classes will declare thrown so none of the classes of my application will tightly coupled with the technology and always the exception is reported is same irrespective of which technology we are using for performing the persistency operation .

➔SO always the technology specific exception will neutralized into spring JDBC exception so always my presentation tier component will get DataAccessException so that I don't need to modify the logic if the exception is always same.

Why we need Custom exception in spring?

➔While upon receiving the request from controller, service should not pass the request to the DAO rather it perform internally some business operation.

➔During the business operation it has some piece of code to execute. There is chance that the piece of code might be report an exception.

➔Like to receive the data from the file system or it has to talk to an external component provided by framework or to con the data type from float to long.

➔So there is a chance that while performing the business operation it might be encountered an exception (Technology specific Exception). SO it could be a checked or it could be an unchecked exception.

➔After getting the exception service should not throws it to controller. Because if there is a change in exception aging I have to modify the logic inside my controller.

➔ So my controller is being tightly coupled with the technology specific exception or external component exception.

➔So to avoid this problem my service has to catch the exception and convert the technology specific exception or external component exception into application specific exception so that the every exception that is being come into my service will be same , so I still need to have application specific exception(Custom Exception).

How to design Custom Exception (Application Specific exception)?

➔These are the exceptions that are extending from Throwable & design them as RuntimeException. We never design them as checked exceptions.

➔We always need to design them as unchecked exceptions.

➔ Because unchecked exceptions can be handle if you really required.

➔and we don't need to have any alternate path of execution to continue the execution, rather we just need to display an error page so we want that to be runtime exception.

➔So I will have my own bunch of exception classes being defined as part of my application. Inside these classes module specific exception will be there inside it all the exception will be there.

➔So application specific exceptions are Runtime Exceptions.

➔Now while performing the business operation my business delegate will get some business exceptions. So all this exception will come into service.

➔Now service has to catch the exceptions and converted into application specific exception.

Why should I throw application specific exception only?

➔Because my presentation tier components will decoupled from by business tier functionality.

➔So service throws the application specific exception to controller.

➔Then controller will throw it to mapper then exception mapper will converts them into business friendly error message and send it to the end-user.

➔But actually while performing the business operation when service encounter an exception he will not converted the technology specific exception into application specific exception.

Who will convert the business specific or technology specific exception into application specific exception during the business operation?

➔When service ran into exception it will throws the exception it will not catch. It will not translate let it throw. The CustomExceptionTranslator will get it.

➔You write your own class which is exception translator class, to him the exception that has been reported & who is reporting the exception both will be passed as an input.

➔This class will make the best effects on translating the technology into application specific exception and will throw to controller.

➔Now the exception translation will be taken care by custom exception translator & service will just have to throw the exception & that will received by custom translator which is always translate it to application specific exception (AOP throws advice concept.).

How Application Form data manage by spring?

➔The controller is read the data send by the JSP page as part of command or form. Now controller is pass the control to the service to perform business operation then service will pass the control to the DAO to perform persistency operation.

➔The service needs the input from the controller to perform business operation. So controller passes the command or form as an input to the service.

➔**The command class or the form class is a POJO class. It is not a spring specific component. So we don't need to create one more component to decouple my business tier from presentation tier technology**.

➔If it is an action form it is a struts specific component. If it has to pass to the business tier then business tier will tightly coupled with struts.

➔ Just to avoiding this problem we need to create one more component which is called as value object which is a duplicate transform of action form or struts form.

➔So in the above case we are unnecessarily creating one more component which contains the same data & we are trying to write more logic for mapping the data which is waste of logic & waste of development time.

➔So to ensure that to avoid creating one more set of class called value object the spring people is make it as POJO class .

➔This POJO class contains attributes. The attribute is exactly same as JSP page field name and the number of attribute is same as the inputs that are sent by the JSP page.

➔The data type of these attributes can be any type. It could be any object type or any attribute primitive type can be the attribute.

How the string type attribute is converted into my object type?

➔By using property editor (Spring Core) we can achieve this functionality.

➔Writing anything as an attribute , while populating the data into these attribute if I has resulted in an exception due to unable to convert then spring will gracefully handle it .

➔Suppose if my business logic wants some different data like int to long I don't need to modify the data inside the controller. I don't need to do anything just modify the attribute so that spring will takes care of binding into long. So that I can give anything still my presentation tier will not impacted because he never populates the data, spring takes care of populating the data.

➔If there is any error will occurs while populating the data can be manage to be display as an validation error with repopulating the data .

➔Now the controller will pass form as an input. Form will contains business representation value only.

What is Transaction?

➔Every operation should be atomic in nature either do or die, complete the operation completely commit or rollback completely is called as transaction.

➔Now all the step with in that operation to accomplish either has to be completed or has to be rollback completely. Otherwise the system will be left in inconsistence stage and we are not applying transactionality. Either do or die means transactionality.

There are two types Transactions.

➔Local Transaction.
➔Global/XA Transaction.

➔The resources that whom we are applying the transaction can be locally located or can be separated over the network and can be distributed located also.
➔So XA Transactions are can be called as distributed transactions.

Transactional Boundary Means What?

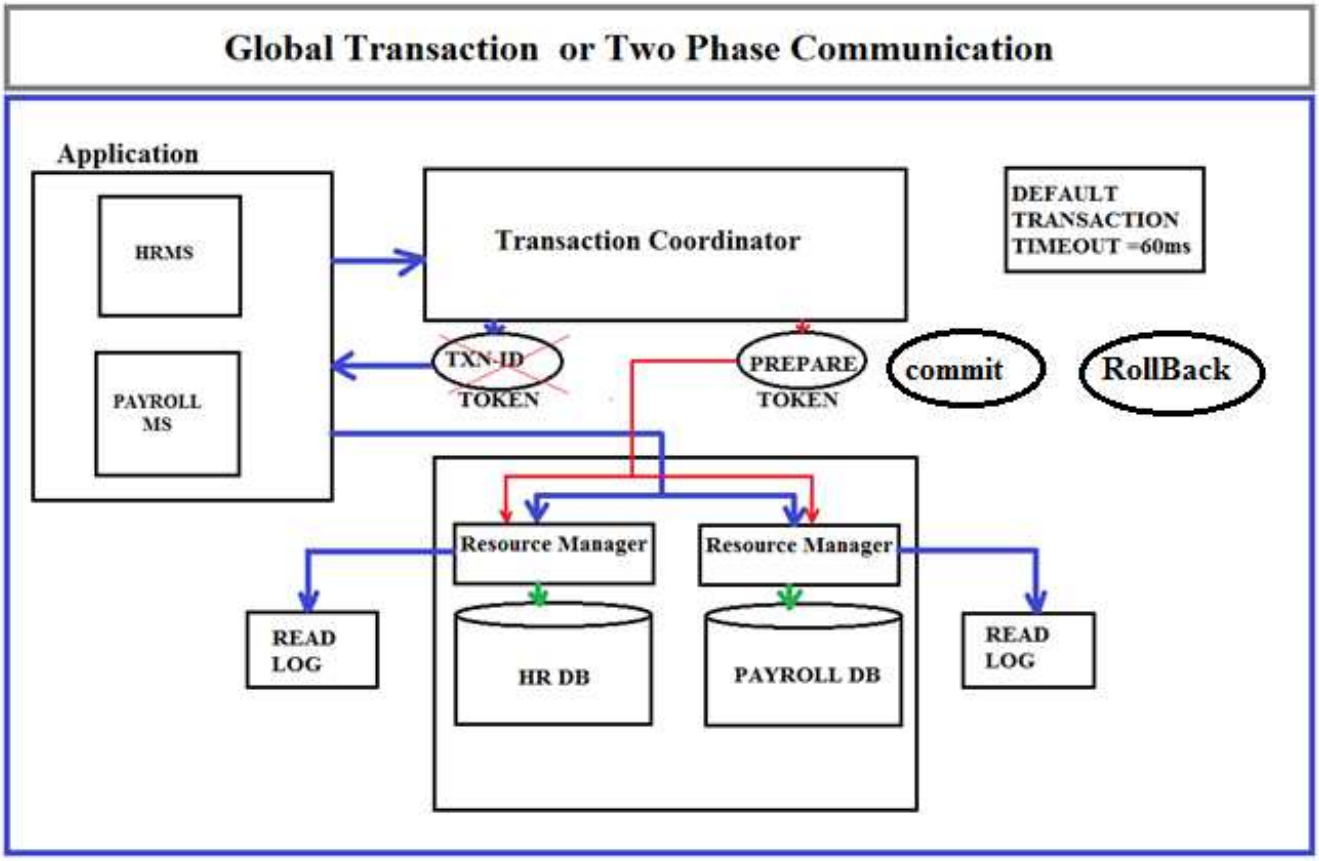
➔I wanted all the operation to be committed or rollback, bunch of operation we combing to build is called boundary

What is called as Local Transaction

➔When a transaction boundary having one transaction resource with whom we can commit or rollback then it is called local transaction.
➔In Local Transaction we are issuing the commit on resource level that means on the connection we are committing or roll backing.

What is called as Global transaction

➔In Global transaction we are issuing commit or rollback in multiple resources.
➔The commit/rollback we are issuing across all the resources at one shot
➔In Global transaction connection will not commit and rollback because there will be transaction coordinator (User Transaction Manager).



➔JDBC API doesn't support local transaction
➔It has never being designed to supporting Global transaction, it only can support local transaction.

What is Resource manager(Data Source)?

➔Resource Manager will help us to indentifying who are all the resources that are participating under the transaction.

How do I need to work with Transaction? Is JavaEE is has provided the support for Transaction or not?

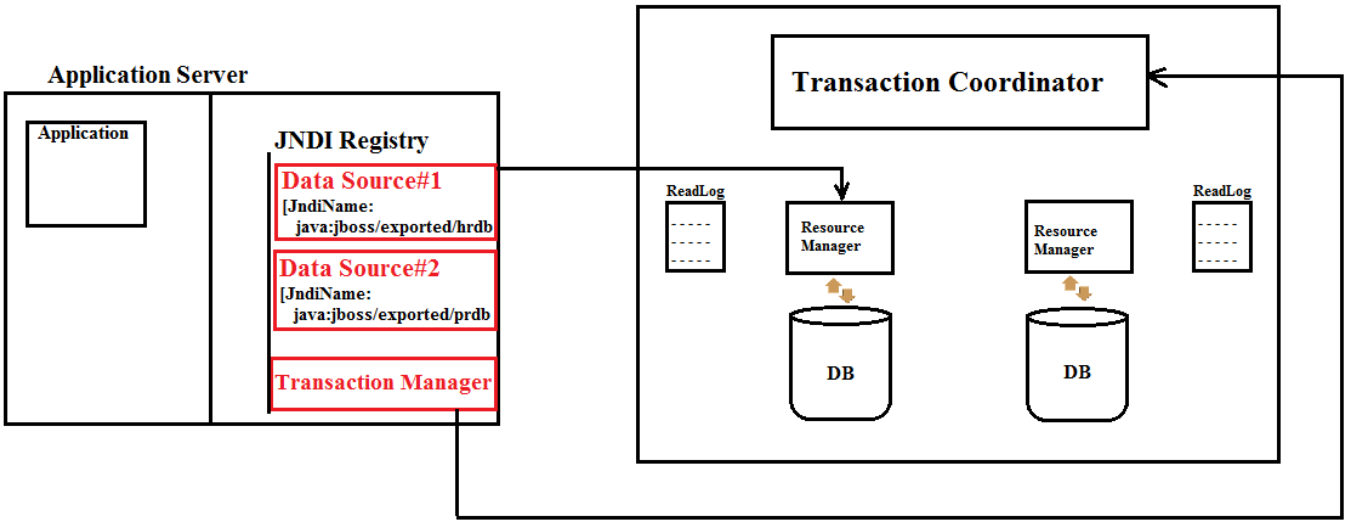
➔Yes .There are basically 2-types of Transaction are there .
Local Transaction Global Transaction/XA Transaction/ Distributed Transaction

Is the two type of Transaction are related to J2EE only or there is a global mechanism of working with Transaction?

➔It is not specific to java the transactionality of the world across the system can exist in 2-forms one is local another is global.
➔For .net application also local Transaction is there Global or XA Transaction also there, Every programming language support local and global Transaction. Depends on the capability of programming language .
➔Few programming language are does not support enterprises application development then we can't address global Transaction then only support for local Transaction. So deepens on the language capability they are supporting transactionality.

How to work with Transaction? What do you mean by local transaction? What is Global transaction?

➔I wanted all the operation to be committed or rollback ,bunch of operation we combing to build is called boundary .
➔Within the Transactional Boundary i have transactional participating resources.
➔Data Base can called transactional participating resources. Data base only support transactionality .
➔The data we have written will not be written permanently rather it will store and gate committed and will roll back base on the instruction .That is called transactional participating resources.
➔We can't say a file is transactional participating resources because when we write the data in a file data store permanently that is nothing called roll back.
➔Files are not participating transactional resources because if you commit or rollback there is no use. That's why these resources are not called transactional resources.
➔A JMS queue is a transactional resources ,other than database JMS also participating in transactional resources.
➔Only one API in java is JMS who provide support in Transaction, when we commit the Transaction then only message can connect to the Queue.
➔When receiver acknowledges receiving the message then message will be remove from the queue.



➔This is my Transactional Boundary, within the Transactional Boundary there are multiple Transactional participated resources .and on this resource we are applying changes.

➔I wanted to commit ,when ever I am committing am I committing on two databases or two Transactional participating resources or multiple Transactional participating resources or within my Transactional boundary when I issues a commit ,I am committing one single resource only ?
➔Committing one single Resource only. Within my Transactional boundary if there is only one transactional resource on whom we are applying commit or on rollback then it is called local Transaction.

➔If the Transaction boundary there are multiple Transaction participating resources.

When we use Global Transaction?

➔Suppose there is a organisation in which thousands of thousand employee or lacks of employees are there I wanted to maintain the role information the employee information within my application .

➔We need multiple resources because if the system will down the whole application will be effected the total business will be loss.
➔That's why peoples are buying several several software like peoplesoft,OracleAppls,If one resource is down the total application is not stopped .If we are using one Resources then total business will be loss.
➔So now We have two application .One is HRM project another is Financial Project. HRM application are using HR database and Financial application is using on Payroll Database .
➔They splited the business application into two different application and database also different.
➔The application may be different .because the HRM application can access the employee of the organization but Financial application is not access by Employee of the Organization only payroll team only will use it .
➔So HRM is a global visibility but Financial is not global visibility that's why they splited the application two part .so this is a example of need of Global Transaction.
➔Both are related to employee information why we can't store in one single database itself.
➔Because the way we are using HRM is different and the way we are using Financial is different.
➔Financial information is more secure .That's why they are using two databases.
➔Every data base will not support Global Transaction oracle, mysql are support Global Transaction but some of database are not support global Transaction.
➔In my transaction Boundary I have Global Resources. So that in one shot commit or rollback should be happen.
➔So multiple Transaction Resources Manager are there on whom I am committing or rollback that is called Global Transaction.

How To work with JDBC Transactions?

➔JDBC API only support Local Transactions.
➔JDBC API is not support Global Transaction .
➔We can write **connection.autoCommit(false)** and do the operation and after complete all the operation we have to commit but at the time of committing if exception come we are not rollback other committed resources. So Global Transaction is not possible in JDBC API

How to work with Multiple database and what is Global Transaction ?

➔If you have two database then you need Two connection. two preparedStatement .
➔Then we have to restrict the autocommit as false of two resource
➔Set the flag variable if flag is true then commit. But in this approach so many problems are there.
So don't directly tell to resources talk to Transactional Coordinator tell to begin transactional convocation state so that i can talk to multiple Resources.

➔The Transactional Coordinator will create Transactional Id and send to the client and tell give the id to the Resources who want to talk otherwise he don't know which resources you are talking and which resources will be committing or roll back.

Internals of Global Transaction

➔In Global transaction we are working with both the resources, I have done some changes into the HR database and I have done some changes in the payroll database as well.

➔Bothe the resources has been used ,once I done with all my changes I will go to transaction coordinator asking the transaction to commit, then transaction coordinator will create a token (Prepare Token) and the it will passes that token to 1st resource and 2nd resource .

➔One after the another one , It passes the token to resource manager 1 and asking to resource Manager to check his underlying resource is ready to commit or not then let him prepare it to the commit state , then immediately the resource manager will goes to the resource and ask whether the database or current snap information or whether the current state of transaction is existed in the snap shot of the database or not .

➔Then the resource will perform several checks like an integrity check foreign key constraint check, name key constraints check and then primary key constraints check, all certain checks that he has enforced on the underlying table to be validated and the verify whether the data changes that we have made are consistent with the database or not if it is consistent immediately he will say yes boss I am ready to commit.

➔Then the resource manager will say lock your self don't allow anyone to perform the operation during this time then the Resource 1 will be locked.

➔Then the resource manager will maintain the changes in the read logs.

➔Then the Transaction token return to the transaction coordinator and again the transaction coordinator passes that token to second resource manager 2 then the resource manager will say lock your self don't allow anyone to perform the operation during this time then the Resource 1 will be locked . Then the resource manager will maintain the changes in the read logs.

➔Now both the resources are ready to commit the transaction coordinator will issue a commit token and the commit token will pass to one resource after another resource.

➔Resource 1 has been committed and the commit token will be return to the transaction coordinator then the transactional coordinator again goes to the resource manager two asking him also to commit he goes to the underlying resource and commit. Then everything is success.

But in failure case...

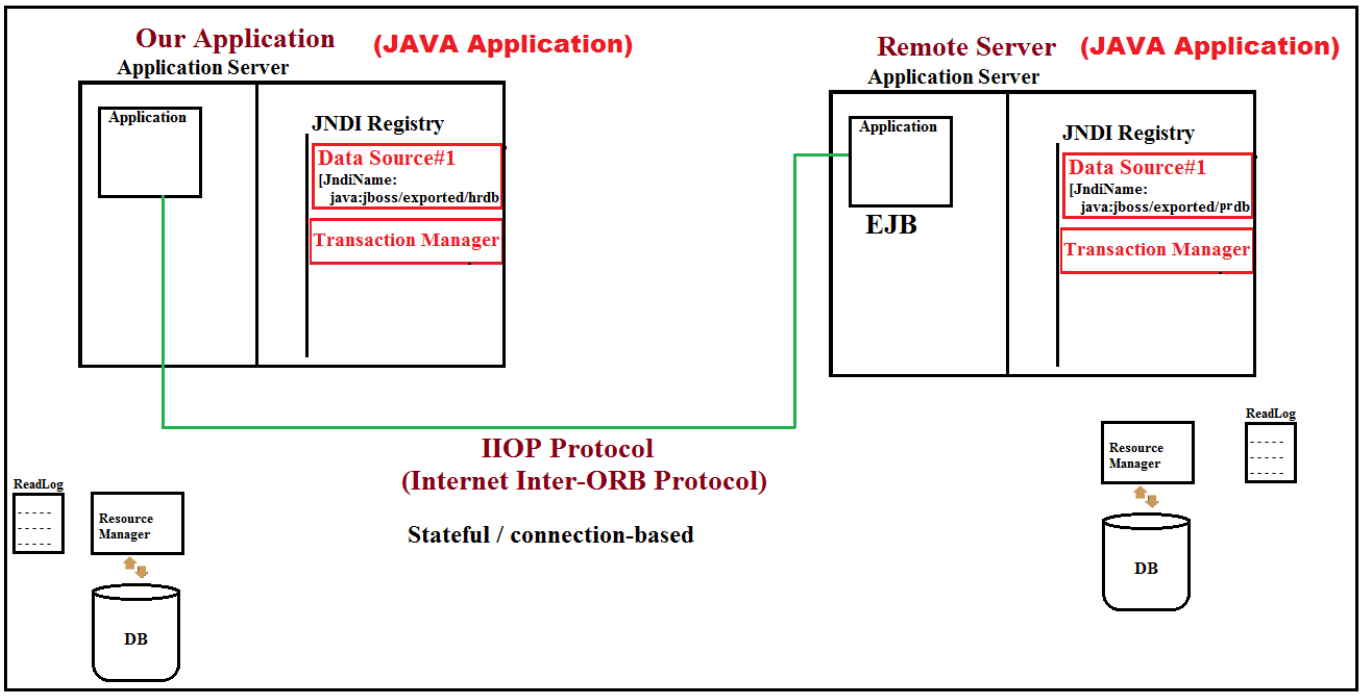
➔Commit has been issued to the resource manager 1 he goes to the resource ask him to commit. His commit is completed the commit token will return back to the resource manager 2 asking him to commit, suppose database went down

➔Now the resource 2 is not available Then the commit token will not return, so that the transactional coordinator will wait for certain amount of time i.e. transactional time out (default is 60 ms).

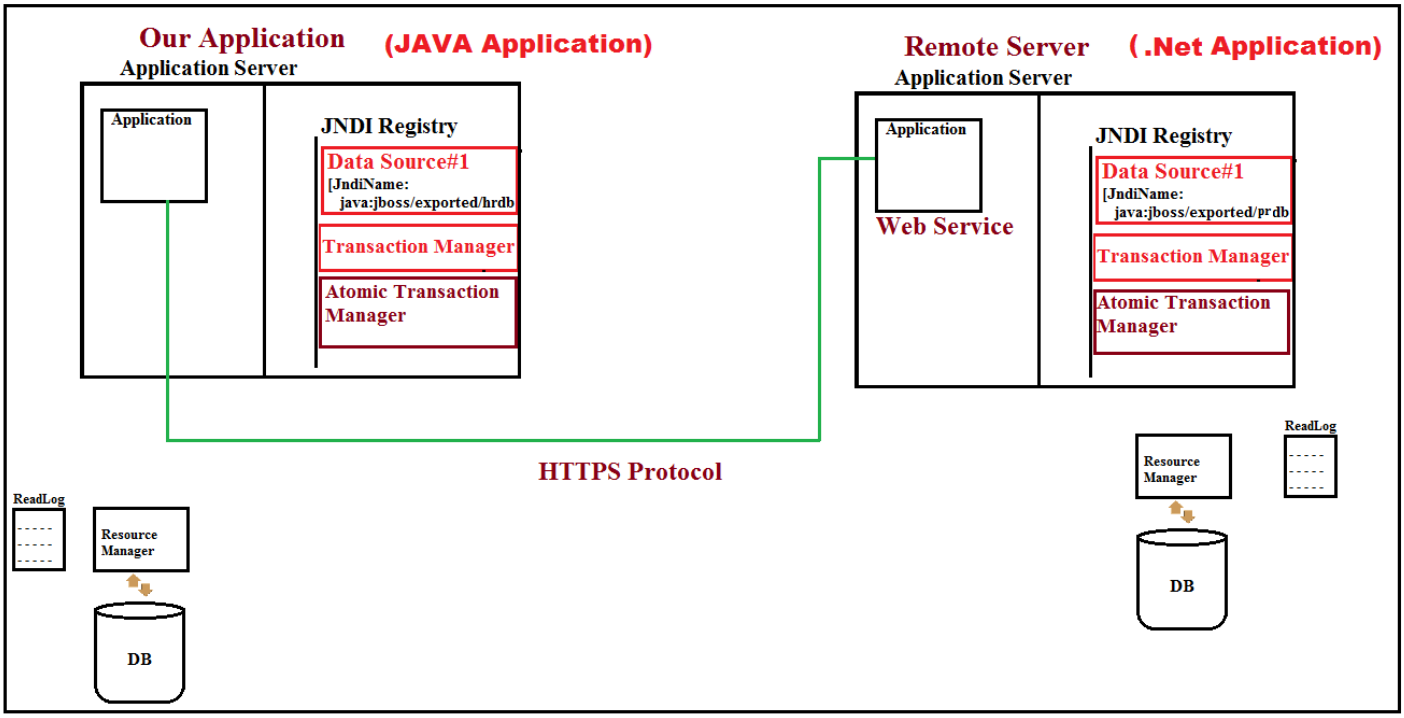
➔Then the immediately transaction coordinator will issue one more token called roll back token send the roll back token to resource manager 1 .

➔Now resource manager will issue a roll back on database but the database is already committed there is nothing called rollback.

➔So the resource manager will goes to read log if the 1st statement is insert then it will delete that record and then second operation is update then he will go and then modifies to the old state and then reverts the changes to the old state and will roll back then the token returns to the transactional coordinator and communicate the same.



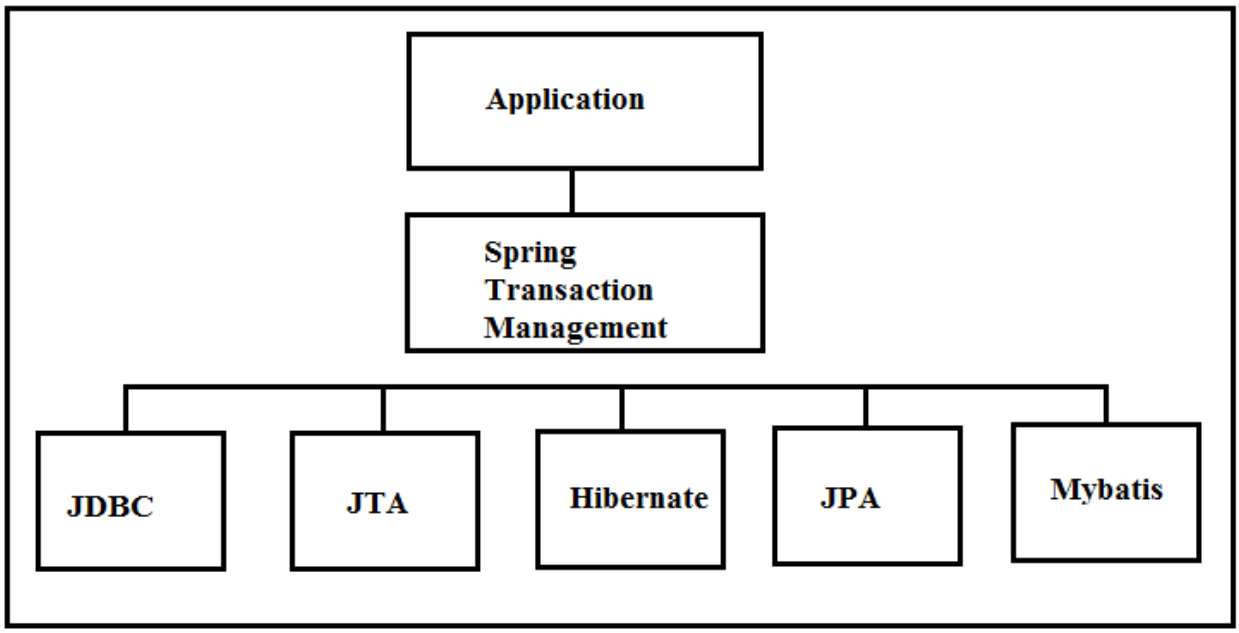
➔To perform distributed transaction management we need either application server(JEE server) or third party API like atomokis that's gives user transaction object as transaction manager.
➔We also need XA JDBC driver , XA Datasource manager to get Transaction manager required for distributed Transaction management.
➔Here are never directly call connection.commit() or connection.rollback().
➔We ask transaction Manager to perform these operation by getting connection through Resource Manager (DataSource).
➔The Server like JBoss supply "userTransaction" object in its JNDI registry having fixed JNDI name lije "java:jboss/exportyed/userTransaction".
➔We use JTA support to perform global transaction management in programmatic approach because JTA only support programmatic Approach.
➔We use EJB it supports both programmatic and configuration approach to perform Global transaction management.
➔ Based on XML configuration the EJB container applies Transaction management services on to EJB Components.
➔EJB transaction management service is really nice but it is Heavy weight.



➔The above Global transaction can be happened if the technology is same.
➔In case of different technology we need web service transaction provided by ws* specification.
➔There will be an atomic transaction manager will be there who manage the global transaction.

Spring Transaction

➔We used different java,JEE API's or 3rd party API like JPA,Hibernate, and etc… to perform different models of Transaction management.
➔But If I want to switching from local transaction management to global transaction management or one transaction management technology to another transaction management technology then it is very complex for the programmer.
➔Spring provides unified environment to work with ant transaction management because it is a framework providing abstraction layer on multiple technologies which supports all the technologies ,The same piece of code will works with any persistency technology framework.
➔Due to this we can make spring to use internally any implementation of transaction management.
➔We need to tell begin the transaction, commit the transaction, or roll back the transaction on spring transaction and even we need to tell which transaction we are using (JDBC/Hibernate/JTA…)
➔If you tell the Spring transaction will takes care of corresponding transaction management API is being provided technology for applying the transactionality.



The process of combining related operations into single unit and executing then by applying do everything or nothing principle called 'Transaction Management'.

Declarative Approach Example

```
public class AppointmentForm {
        private int appointmentNo;
        private int patientNo;
        private String patientName;
        private int age;
        private String gender;
        //Setters & Getters
}
```

```
public class AppointmentBO {
        private int appoinmentNo;
        private Date appointmentDate;
        private int doctorNo;
        private int patientNo;
        //Setters & Getters
}
```

```
public class PatientBo {
        private int patientNo;
        private String patientName;
        private int age;
        private String gender;
        private String contactNo;
        //Setters & Getters
}
```

Test.java

```
public class Test {
        public static void main(String[] args) {
                ApplicationContext context=new ClassPathXmlApplicationContext("com/st/common/application-context.xml");
                AppointmentForm form=new AppointmentForm();
                form.setAppointmentNo(3);
                form.setAppointmentDate(new Date());
                form.setDoctorNo(1);
                form.setPatientNo(4);
                form.setAge(27);
                form.setContactNo("985399001");
                form.setGender("M");
                form.setPatientName("Bhima");
                AppointmentController controller=context.getBean("appointmentController",AppointmentController.class);
                controller.newAppointment(form);
        }
}
```

application-context.xml

```
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
        <import resource="persistancy-bean.xml"/>
        <import resource="service-bean.xml"/>
        <import resource="aop-bean.xml"/>
        <import resource="contorller-bean.xml"/>
</beans>
```

ApplicationController.java

```
public class AppointmentController {
        private ManageAppointmentService manageAppointmentService;

        public void setManageAppointmentService(ManageAppointmentService manageAppointmentService) {
                this.manageAppointmentService=manageAppointmentService;
        }

        public void newAppointment(AppointmentForm form){
                manageAppointmentService.newAppointment(form);
                System.out.println("App Data Inserted Successfully");
        }
}
```

controller-bean.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans>
        <bean id="appointmentController" class="com.st.controller.AppointmentController">
                <property name="manageAppointmentService" ref="manageAppointmentService"/>
        </bean>
</beans>
```

ManageAppointmentService.java

```
public class ManageAppointmentService {
	private PatientDao patientDao;
	private AppointmentDao appointmentDao;

	public void setPatientDao(PatientDao patientDao) {
		this.patientDao = patientDao;
	}
	public void setAppointmentDao(AppointmentDao appointmentDao) {
		this.appointmentDao = appointmentDao;
	}
	//@Transactional(readOnly=false)
	public void newAppointment(AppointmentForm form){

		PatientBo patientBo=new PatientBo();
		patientBo.setPatientNo(form.getPatientNo());
		patientBo.setPatientName(form.getPatientName());
		patientBo.setAge(form.getAge());
		patientBo.setGender(form.getGender());
		patientBo.setContactNo(form.getContactNo());
		patientDao.savePatient(patientBo);

		AppointmentBO appBo=new AppointmentBO();
		appBo.setAppoinmentNo(form.getAppointmentNo());
		appBo.setAppointmentDate(form.getAppointmentDate());
		appBo.setDoctorNo(form.getDoctorNo());
		appBo.setPatientNo(form.getPatientNo());
		appointmentDao.saveAppointment(appBo);
		System.out.println("Commited");
	}
}
```

service-bean.xml

```
<beans >
	<bean id="manageAppointmentService" class="com.st.service.ManageAppointmentService">
		<property name="patientDao" ref="patientDao"/>
		<property name="appointmentDao"  ref="appointmentDao"/>
	</bean>
</beans>
```

aop-bean.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns:aop="http://www.springframework.org/schema/aop"
	xmlns:tx="http://www.springframework.org/schema/tx"
	xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
		http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
		http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">
		<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
			<property name="dataSource" ref="dataSource"/>
		</bean>
		<tx:advice id="txAdvice" transaction-manager="transactionManager" >
			<tx:attributes>
				<tx:method name="newAppointment" read-only="false" />
			</tx:attributes>
		</tx:advice>
		<aop:config>
			<aop:pointcut expression="within(com.st.service.*)"  id="pc1"/>
			<aop:advisor advice-ref="txAdvice" pointcut-ref="pc1"/>
		</aop:config>
	<!-- <tx:annotation-driven transaction-manager="transactionManager"/> -->
</beans>
```

AppointmentDao

```
public interface AppointmentDao {
	int saveAppointment(AppointmentBO Appbo);
}
```

AppointmentDaoImpl

```
public class AppointmentDaoImpl implements AppointmentDao {
	private JdbcTemplate jdbcTemplate;

	public AppointmentDaoImpl(JdbcTemplate jdbcTemplate) {
		this.jdbcTemplate = jdbcTemplate;
	}

	private final String SQL_FOR_APPOINTMENT = "insert into appointemnt (APPOINTMENT_NO,"
					+ "APPOINTMENT_DATE,DOCTOR_NO,PATIENT_NO) values(?,?,?,?)";

	@Override
	public int saveAppointment(AppointmentBO appBo) {
		int result = jdbcTemplate.update(SQL_FOR_APPOINTMENT,
				appBo.getAppoinmentNo(), appBo.getAppointmentDate(),
				appBo.getDoctorNo(), appBo.getPatientNo());
		return result;
	}
}
```

PatientDao

```
public interface PatientDao {
	int savePatient(PatientBo patientBo);
}
```

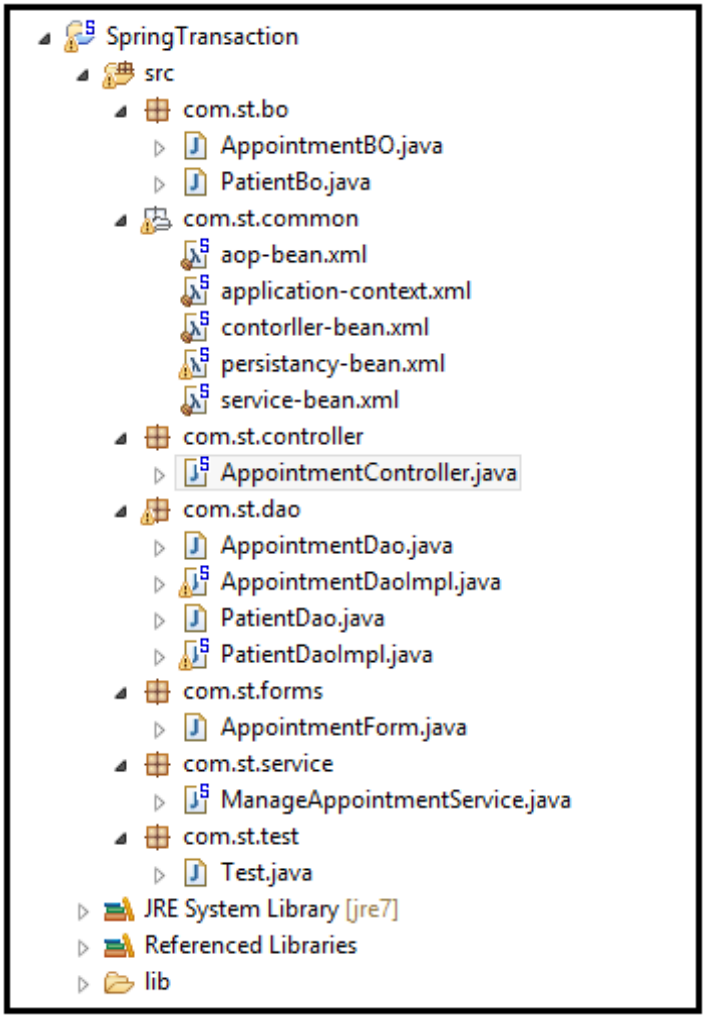
PatientDaoImpl

```
public class PatientDaoImpl implements PatientDao {
	private JdbcTemplate jdbcTemplate;
	private final String SQL_FOR_PATIENT = "insert into patient (patient_no,patient_nm,age,"
			+ "gender,contact_no) values(?,?,?,?,?)";

	public PatientDaoImpl(JdbcTemplate jdbcTemplate) {
		this.jdbcTemplate = jdbcTemplate;
	}

	@Override
	public int savePatient(PatientBo patientBo) {
		int result = jdbcTemplate.update(SQL_FOR_PATIENT,
				patientBo.getPatientNo(), patientBo.getPatientName(),
				patientBo.getAge(), patientBo.getGender(),
				patientBo.getContactNo());
		return result;
	}
}
```

persistancy-bean.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">

		<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
			<property name="driverClassName"  value="oracle.jdbc.driver.OracleDriver"/>
			<property name="url" value="jdbc:oracle:thin:@localhost:1521:xe"/>
			<property name="username" value="spr_usr"/>
			<property name="password" value="welcome1"/>
		</bean>

	<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
		<constructor-arg ref="dataSource"/>
	</bean>
	<bean id="patientDao" class="com.st.dao.PatientDaoImpl">
		<constructor-arg ref="jdbcTemplate"/>
	</bean>
	<bean id="appointmentDao"  class="com.st.dao.AppointmentDaoImpl">
		<constructor-arg ref="jdbcTemplate"/>
	</bean>
</beans>
```

Application Structure



Annotation Approach Example

Test.Java

```
public class Test {
	public static void main(String[] args) {
		ApplicationContext context=new ClassPathXmlApplicationContext("com/st/common/application-context.xml");
		AppointmentForm form=new AppointmentForm();
		form.setAppointmentNo(6);
		form.setAppointmentDate(new Date());
		form.setDoctorNo(1);
		form.setPatientNo(6);
		form.setAge(27);
		form.setContactNo("985399001");
		form.setGender("M");
		form.setPatientName("Bhima");
		AppointmentController controller=context.getBean("appointmentController",AppointmentController.class);
		controller.newAppointment(form);
	}
}
```

AppointmentForm

```
public class AppointmentForm {
	private int appointmentNo;
	private int patientNo;
	private String patientName;
	private int age;
	private String gender;
	//Setters & Getters
}
```

AppointmentBo

```
public class AppointmentBO {
	private int appoinmentNo;
	private Date appointmentDate;
	private int doctorNo;
	private int patientNo;
	//Setters & Getters
}
```

PatientBo

```
public class PatientBo {
	private int patientNo;
	private String patientName;
	private int age;
	private String gender;
	private String contactNo;
	//Setters & Getters
}
```

application-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://www.springframework.org/schema/beans
	 http://www.springframework.org/schema/beans/spring-beans.xsd">
	<import resource="persistancy-bean.xml"/>
	<import resource="aop-bean.xml"/>
</beans>
```

AppointmentController

```
@Controller("appointmentController")
public class AppointmentController {
	@Autowired
	private ManageAppointmentService manageAppointmentService;

	public void newAppointment(AppointmentForm form){
		manageAppointmentService.newAppointment(form);
		System.out.println("App Data Inserted Successfully");
	}
}
```

ManageAppointmentService

```
@Service("manageAppointmentService")
public class ManageAppointmentService {

	@Autowired
	private PatientDao patientDao;
	@Autowired
	private AppointmentDao appointmentDao;

	@Transactional(readOnly=false)
	public void newAppointment(AppointmentForm form){

		PatientBo patientBo=new PatientBo();
		patientBo.setPatientNo(form.getPatientNo());
		patientBo.setPatientName(form.getPatientName());
		patientBo.setAge(form.getAge());
		patientBo.setGender(form.getGender());
		patientBo.setContactNo(form.getContactNo());
		patientDao.savePatient(patientBo);

		AppointmentBO appBo=new AppointmentBO();
		appBo.setAppoinmentNo(form.getAppointmentNo());
		appBo.setAppointmentDate(form.getAppointmentDate());
		appBo.setDoctorNo(form.getDoctorNo());
		appBo.setPatientNo(form.getPatientNo());
		appointmentDao.saveAppointment(appBo);
		System.out.println("Commited");
	}
}
```

aop-bean.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns:aop="http://www.springframework.org/schema/aop"
	xmlns:tx="http://www.springframework.org/schema/tx"
	xmlns:context="http://www.springframework.org/schema/context"
	xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
		http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd
		http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
		http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">
		<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
			<property name="driverClassName"  value="oracle.jdbc.driver.OracleDriver"/>
			<property name="url" value="jdbc:oracle:thin:@localhost:1521:xe"/>
			<property name="username" value="spr_usr"/>
			<property name="password" value="welcome1"/>
		</bean>

		<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
			<property name="dataSource" ref="dataSource"/>
		</bean>

		<tx:advice id="txAdvice" transaction-manager="transactionManager" >
			<tx:attributes>
				<tx:method name="newAppointment" read-only="false" />
			</tx:attributes>
		</tx:advice>
		<aop:config>
			<aop:pointcut expression="within(com.st.service.*)"  id="pc1"/>
			<aop:advisor advice-ref="txAdvice" pointcut-ref="pc1"/>
		</aop:config>
	<tx:annotation-driven transaction-manager="transactionManager"/>
	<context:component-scan base-package="com.st.controller,com.st.service,com.st.dao,"/>
</beans>
```

AppointmentDao

```
public interface AppointmentDao {
	int saveAppointment(AppointmentBO Appbo);
}
```

AppointmentDaoImpl

```
@Repository("appointmentDao")
public class AppointmentDaoImpl implements AppointmentDao {
	@Autowired
	private JdbcTemplate jdbcTemplate;
	private final String SQL_FOR_APPOINTMENT = "insert into appointemnt (APPOINTMENT_NO,"
			+ "APPOINTMENT_DATE,DOCTOR_NO,PATIENT_NO) values(?,?,?,?)";

	@Override
	public int saveAppointment(AppointmentBO appBo) {
		int result = jdbcTemplate.update(SQL_FOR_APPOINTMENT,
				appBo.getAppoinmentNo(), appBo.getAppointmentDate(),
				appBo.getDoctorNo(), appBo.getPatientNo());
		return result;
	}
}
```

PatientDao

```
public interface PatientDao {
	int savePatient(PatientBo patientBo);
}
```

PatientDaoImpl

```
@Repository("patientDao")
public class PatientDaoImpl implements PatientDao {
	@Autowired
	private JdbcTemplate jdbcTemplate;
	private final String SQL_FOR_PATIENT = "insert into patient (patient_no,patient_nm,age,"
			+ "gender,contact_no) values(?,?,?,?,?)";
	@Override
	public int savePatient(PatientBo patientBo) {
		int result = jdbcTemplate.update(SQL_FOR_PATIENT,
				patientBo.getPatientNo(), patientBo.getPatientName(),
				patientBo.getAge(), patientBo.getGender(),
				patientBo.getContactNo());
		return result;
	}
}
```

persistancy-bean.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://www.springframework.org/schema/beans
	http://www.springframework.org/schema/beans/spring-beans.xsd">

	<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
		<constructor-arg ref="dataSource"/>
	</bean>
</beans>
```

While performing the database operation on databases we fall into several problems. These are Problems are 3 types.

➔Dirty Read Problem
➔Non repeatable Problem
➔Phantom Read Problem

What is Dirty Read Problem?

➔Suppose there are two transaction t1 and t2.

➔ When transaction1 trying to perform the data base operation within a time time1.

➔Transaction2 trying to read the records of data from the data base in same time time1.

➔While transaction t2 reading the data from the database. Due to some problem transaction t1 roll backed the problem is transaction t1 changes are not yet committed.

➔Now whatever the data is being read by transaction t2 got changed.

➔And these are not scale data these are dirty data.

➔Because these changes are not permanent these are temporary changes that are not yet being committed.

➔Before transaction t1 leading to commit itself, transaction t2 going to read it, i.e. if transaction t2 read the uncommitted changes of the transaction t1.

➔So the read of transaction t2 is called dirty read.

➔In case the transaction t1 is able to rollback the changes, the changes of data is being called dirty data

What is Non repeatable Problem?

➔There is an active transaction, under this active transaction time 1ms, we executed one select query it return record r1.

➔Now under the same active transaction at time 2nd ms, when again I am executing the same select query I may not get the same record r1.

➔Even the transaction is active the same record r1 may not come.

➔Because there is no guaranty same data will come (Like Movie ticket booking) this is called non repeatable read.

➔In active transaction when we executed a select query it fetch some records of information.

➔I expect the same records of information will be available until I close the transaction,

➔But even the transaction is active the data is not available why because while transaction t1 is fetch the data parallely other transaction t2 is trying to modify the data.

➔That means querying the data 3 records are matching when 3 records are matching what am I querying my transaction have not locked

➔So that other people can update the data or can delete the data from database so in such case non repeatable problem will occur.

➔So we have to lock the record by my transaction, so that when the other people trying to performing an update operation or the delete operation on this record those will not updated until I release that record.

➔So to make this 2nd ms also the same data to be query record, we need to lock the record by specifying the Isolation Level.

What is Isolation Level?

➔To control the way we wanted to use the data within you transaction by default every record that we read from the database will not be locked.

➔When we performing the read-only operation the data base will not require any lock of that record that we have queried.

➔So that there is no guaranty the same record will come even though transaction is under active.

➔But my requirement is until I completed my operation I don't want anyone to permit the same records of information to modify parallely by other people,

➔We can get the control by locking, so from application we have to specify our isolation level to lock the record.

What is Phantom Read Problem

➔The transaction is under active.

➔Under the active transactionat 1ms I executed one select query.

➔The select query has selected bunch of records (like select * from product).

➔Now again I am execute the same query under the same active transaction at 2nd ms.

➔There may be a chance query will fetch more number of records than pervious.

➔Because the we have not locked the table some other transaction may insert the new record.

➔So Sometimes we want the table level lock instead of record level lock.

➔This is called as Phantom read problem

Note:
Every time when you open the transaction you have to specify the isolation level,

What are the Isolation levels supported by Databases?

➔**READ_UNCOMMITTED**

➔**READ_COMMITTED**

Don't read the uncommitted transactions that are made by the other transaction until they committed. Once they committed then only read that data otherwise don't read data into my transaction

➔**REPEATABLE READ**

When we query the data from the database whatever the records of data returning locked, when we re-query the same we will get same data.

➔**SERIALIZABLE**

Who

➔Few Data bases will not supports serializable or repeatable read

➔So depends on the database we are using the database is going to support Isolation levels.

➔HSQL it is a in memory data base it doesn't supports Isolation level. MYSQL will support Isolation Level. Oracle will support Isolation Level. DB2 will support few levels only.

➔When you are using local transaction the isolation levels may not work at all.

➔You can't control the isolation level without working with global transaction why because it may not support by many of the drivers.

➔The default Isolation level is READ COMMOTTED, but if you want to change Isolation level from java you need to tell the isolation level while beginning the transaction.

➔Which can't be done unless until you are using global transaction the two phase committed drivers are there then only you can specify the isolation levels.

When we use Spring Transaction we will get

Isolation Levels	Dirty Read	Non Repeatable Read	Phantom Read
READ_UNCOMMITTED	Yes	Yes	Yes
READ_COMMITTED (Default)	No	Yes (Because we are not locking the record)	Yes (Because we are not locking the table)
REPEATABLE READ	No	No	Yes (Because we are not locking the table)
SERIALIZABLE	No	No(Record Locked)	No(Table Locked)

➔The highest level of Isolation is Serializable

➔The Lowest level of Isolation is Read Uncommitted.

➔If I want high performance within my application while retrieving the data from the data base I want super performance when I want to retrieve the data use READ_UNCOMMITED.

➔Serializable will give poor performance because while one transaction using the database other transaction are not allowed because the table has locked.

➔It is not recommended to use REPEATABLE READ because roll able locking is very costly.

➔For Example in FlipKart or such kind of applications when you added the product to the kart, when you are checking out then the product has to be assign to you and no one should be permitted. It is only possible way is when you use REPEATABLE READ.

➔Most of the time people are using READ_COMMITED only.

Spring Transaction Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx">

            <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
                  <property name="driverClassName"  value="oracle.jdbc.driver.OracleDriver"/>
                  <property name="url" value="jdbc:oracle:thin:@localhost:1521:xe"/>
                  <property name="username" value="spr_usr"/>
                  <property name="password" value="welcome1"/>
            </bean>

            <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
                  <property name="dataSource" ref="dataSource"/>
            </bean>

            <tx:advice id="txAdvice" transaction-manager="transactionManager">
                  <tx:attributes>
                        <tx:method name="newAppointment" read-only="false" isolation="READ_COMMITTED"/>
                  </tx:attributes>
            </tx:advice>
            <aop:config>
                  <aop:pointcut expression="within(com.st.service.*)"  id="pc1"/>
                  <aop:advisor advice-ref="txAdvice" pointcut-ref="pc1"/>
            </aop:config>
      <tx:annotation-driven transaction-manager="transactionManager"/>
      <context:component-scan base-package="com.st.controller,com.st.service,com.st.dao,"/>
</beans>
```

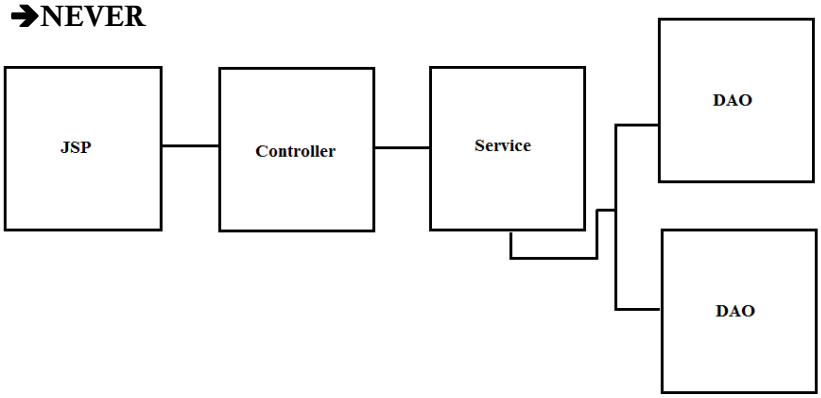
➔Here Transaction Advice is beginning the transaction using Transaction Manager.
➔Transactions Advice will tell to the Transaction manager begin the transaction in so and so isolation level.
➔When we enter into the method instead of entering into the service class method the control comes to advice method.
➔Advice will goes to the Transaction manager asking him to begin the transaction.
➔The transaction Advice has to tell in which method we want to begin the transaction by using read-only=”true/false”, and the programmer has to tell to transaction advice begin the transaction using so and so isolation level.
➔ By default isolation level is READ_COMMITED in most of the databases.

```
@Transactional(readOnly=false ,isolation=Isolation.READ_COMMITTED)
```

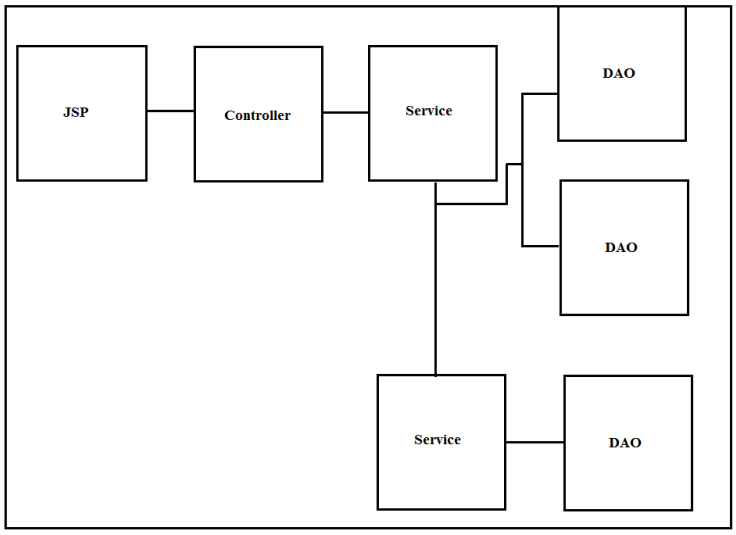
What are the Propagation levels supported by Databases?

There are 6 Propagation levels are there

➔ **REQUIRED**
- ➔**REQUIRED_NEW**
- ➔**SUPPORTED**
- ➔**NOT_SUPPORTED**
- ➔**MANDATORY**
- ➔**NEVER**



➔When it comes to spring architecture you can see service can call multiple DAO's also
➔When JSP send the request to controller, then controller delegate the request to the service is calling both the DAO.
➔So whenever Service calls both the DAO, The context of the transaction or the boundary of the transaction will representated by the business operation who is performing (Service).
➔That means Service is the responsible for committing the transaction or roll backing the transaction.
➔Some times for some specific business operation Service may wanted to call multiple services also.



➔Suppose I have an audit service (Cross functionality).
➔Audit is a common business logic which will not directly write in every method write it in one class, all the classes will call the same audit.
➔So when request comes to controller from JSP page controller will delegate the request to Service#1 and Service#1 has to call Service#2 never a controller will call two services, always the controller will going to call only one service of the business tier.
➔When the controller will call service#1 then service#1 will call service#2 , Service#2 also performing the operation by calling DAO.
➔ **But from DAO Exception will come then which one has to roll back? Service#2/Service#1/all has to be roll back, How does the control of the transaction should be there.**
➔**It depends on Propagation Level.**

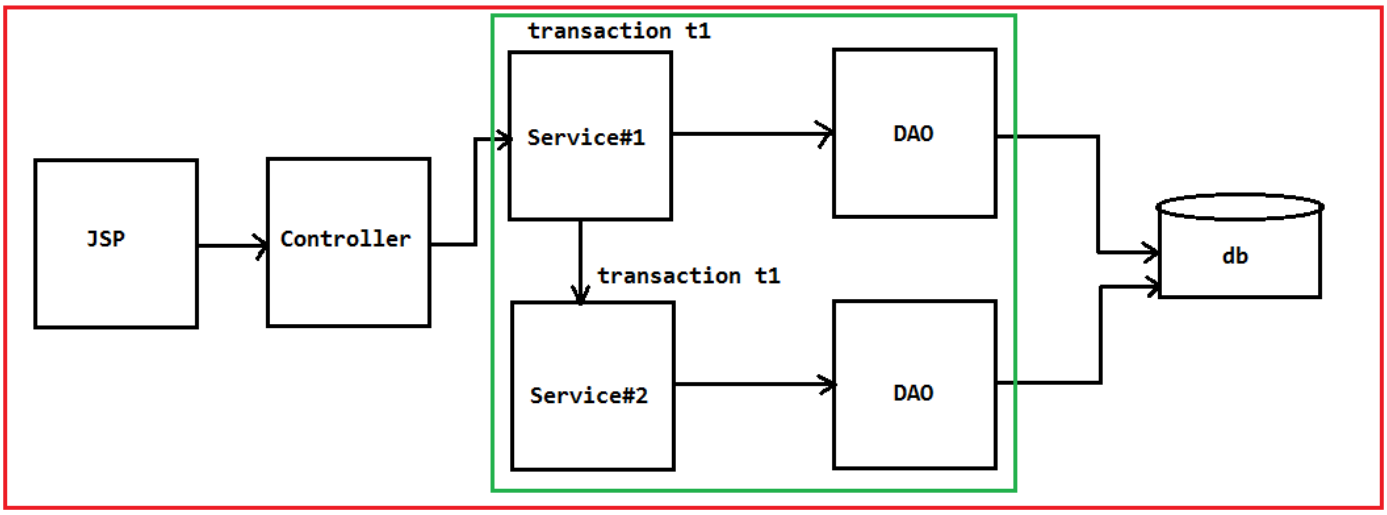
Why Controller will not call multiple DAO's?

➔Controller will never call two services.
➔Controller will call only one class of Service tier.
➔If controller will involving or interacting with more no of business tier component ,then the controller will tightly coupled with more no of business tier components.
➔To avoid it control will call one service only.

REQUIRED (Default)
➔**To manage the Transaction across multiple services we need REQUIRED propagation level.**

For Example:
➔Suppose we have 2 services.
➔When request comes to controller from JSP page controller will delegate the request to Service#1 and Service#1 called Service#2.
➔**If service#1 has a transaction t1 that is already there then service#2 will not start a new transaction.**
➔Service#2 will join in existing transaction t1.
➔That means service#2 required same active transaction t1 to perform the business operation.
➔If there is no active transaction then service#2 will begin a new transaction.

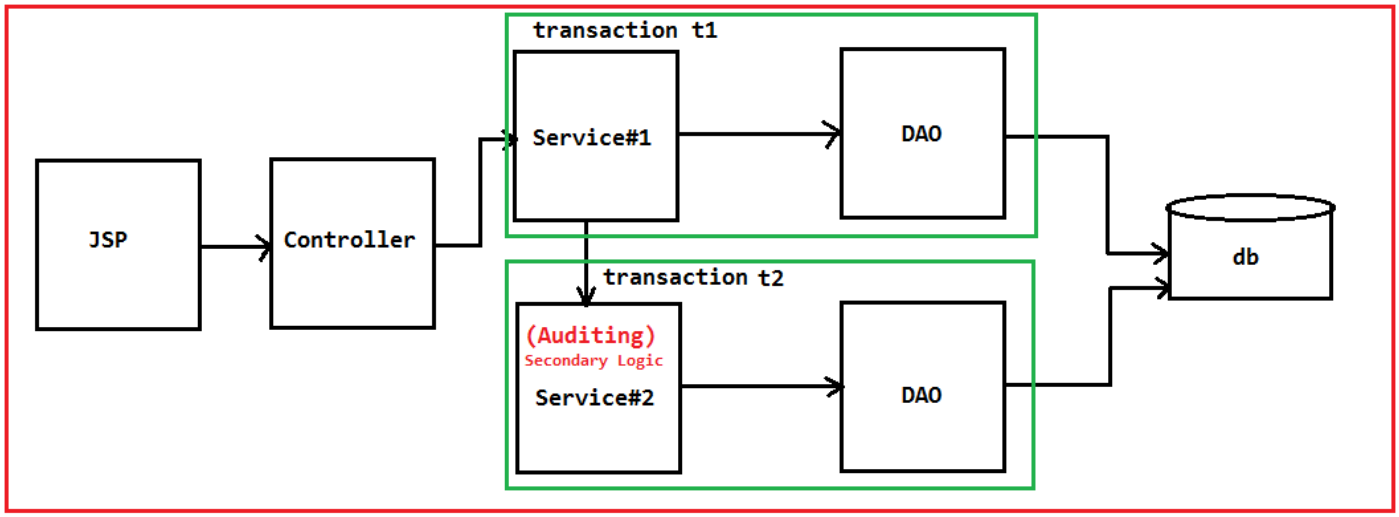
➔**When the Service#2 call the DAO to perform persistency operation at that time Exception will come then the DAO will throw the Exception to Service#2 then Service #2 will re-throw the Exception to Service#1 like this whole transaction will be roll backed.**
➔This is how we can manage the Transaction across all the services using REQUIRED Propagation level.



Multiple Services in Same
Transactional Boundary

REQUIRED_NEW
➔**Here If there is no transaction then begin a new transaction ,if transaction is there then also begin a new transaction that is REQUIRED_NEW.**

For Example:
➔Suppose we have 2 services.
➔When request comes to controller from JSP page controller will delegate the request to Service#1 and Service#1 called Service#2.
➔**If service#1 has a transaction t1 that is already there then service#2 will not join in the existing transaction t1 it will begin a new transaction t2.**
➔Service#2 will not join in existing transaction t1.
➔That means service#2 required a new transaction t2 to perform the persistency operation.
➔**When the Service#2 call the DAO to perform persistency operation (in Audit table) at that time Exception will come then the DAO will throw the Exception to Service#2 then Service #2 will not re-throw the Exception to Service#1 rather Service#2 has to handle it and maintain in log file and return the success value because if you don't write use audit also business logic will not get impacted, So it should not re-throw that Exception. Because the primary operation has completed in Service#1.**



Different transactional boundary within multiple Services

SUPPORTED

➔**When there is an active transaction I am going to join in same transaction, if there is no transaction then I will not execute without transaction that is called supported but I will not throw any Exception**.

For Example:

➔If there is a transaction is already being started by Service#1 then service#2 is going to join in the same transaction.

➔If there is no transaction then Service#2 will not begin a new Transaction if someone call and give the transaction then Service#2 will participate.

NOT_SUPPORTED

➔**I don't want transaction, if someone started I will suspend and I will work without transaction.**

For Example:

➔Suppose we have 2 services.

➔When request comes to controller from JSP page controller will delegate the request to Service#1 and Service#1 called Service#2.

➔**If service#1 has a transaction t1 that is already there then service#2 will suspend (Not rollback) the existing transaction t1 and it will work without transaction.**

MANDATORY

➔**You have to have a transaction to call me, I will not start the transaction**

For Example:

Suppose we have 2 services.

➔When request comes to controller from JSP page controller will delegate the request to Service#1 and Service#1 called Service#2.

➔**Service#2 will check whether the transaction is there or not there If not the throw an exception.**

Service#1	Service#2	Combination is Valid/Invalid
REQUIRED	MANDATORY	Valid
REQUIRED_NEW	MANDATORY	Valid
SUPPORT	MANDATORY	Invalid
NOT_SUPPORT	MANDATORY	Invalid

NEVER

➔I never support transaction If someone is calling having transaction I will throw Exception.

Note:

➔If any service will throw any checked Exception then the transaction will not get Rollbacked.

➔Because Checked Exception means Recoverable (alternate path of execution is there) that's why it will not roll back's the operations.

➔So in your application recommended to use Unchecked Exception.

➔ But sometimes requirement is it throws checked exception but I want to roll back then you have to do some extra configuration.

```
<tx:advice id="txAdvice" transaction-manager="transactionManager">
    <tx:attributes>
        <tx:method name="newAppointment" read-only="false"
            isolation="READ_COMMITTED" propagation="REQUIRED" rollback-for="<Checked Exception Name>" />
    </tx:attributes>
</tx:advice>
```