**Interview Questions about the project architectures:**

🡪Tell me about yourself?

Wish or greet both we each other then we have to tell or summarize the experience and tech of which type application we developed and roles and responsivities

🡪What is the current company u r working?

Then must we need to iam working write now in so and so company.

🡪What is the project u r working now and what is ur role, what are the modules and which module?

🡪What is your client and its location and which tech have used in this project?

🡪What is your recent project?

🡪What is your last project?

The last project that I have worked is almost 1½ or 2-years back in the year and the project name and then client name and which type of application and business domain of the project. If u permit me I can explain the current project more detail as compared the last project bcz it has been completed 1½ or 2-years if you want I can explain but if current project I will be more comfortably and in detail I can explain of permit.

🡪What is the project details?

We need to tell

Project Title,

Project Duration,

When it has been started.

When it has been completed

Who is your client

Client and its location,

Description of project

Roles and responsibilities

No.of modules

Which module r u working

What are tech and version numbers

🡪What is your project Architecture?

We need to tell the plz give me pen and paper I will be able tell clearly. If he asked you r good then we need to say bcz even I involved in the design aspects and learn these thing with help of my project tech lead and manager help.

Note:

After some time we can tell i involved in even design aspects as well. Get good impression in 1st time bcz if we wrong or lagging in some time then he will help us if we tell just give some moment time I will be think. Don’t say I don’t know.

Note:

We should not say I didn’t involve in that like that.

Generally the role of the developer will not limited to the certain module rather we need to say even I contributed in other modules as well sometimes whenever the bugs or and I know the every module but I worked in specific module and I know the end-end of project bcz I involved in the end-end bugs fixes as well.

🡪Did u generate the PDF?

Sorry boss I didn’t worked on the generations rather it is taken some other developer.

🡪Did u written the Ant built scripts?

Sorry I didn’t worked on the ant build scripts bcz it is taken care by another member of our team. But I know how to build the application using ant.

**🡪Project Architectures:**

🡪Multiple Design patterns/components are combined together to build a best solution is called as Architecture and these architectures are standard.

Most of the times the Architectures are depends on the nature of the application and tech we using to develop the application and so the architectures are different from the nature of application/project and tech in which we are developing.

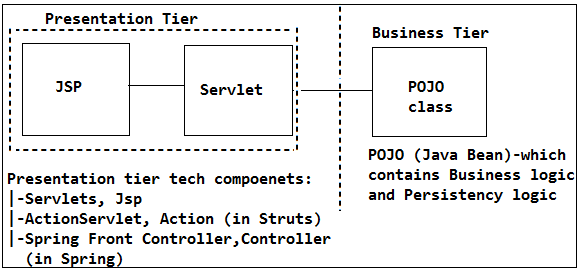
For example consider some of the combinations like

Servlet + JSP Based Project Architecture.

Struts + JSP Project Architecture

Spring + JSP Project Architecture etc

**🡪1. Servlet + JSP Based Project Architecture:**



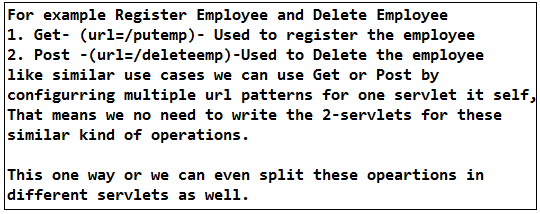
**🡪Servlet as Controller Component:**

(Or)

🡪Why servlet is called as Controller component?

The servlet is the one who will decides which operation has to be done based on the incoming request it is received.

That means the decision making of what kind of req/nature of req is coming get/post based on that it will performs the req processing.



**Separating the Persistency logic and Business Logic form Presentation tier:**

(Or)

🡪Why we need to separate the Business Logic and Persistency logic?

Servlet should not do persistency and Business Logic bcz

(i) If we switch from one persistency tier tech to another presentation tier tech or frame work like spring, struts then we need to re-develop the whole code in another presentation tire tech component. That means persistency tier logic and business logic will be tightly couple to presentation tier tech.

(ii) If we write the common business logic in the presentation tier tech components then this logic will not use by the non-web components. That is the reason we should not write Business logic and Persistency logic as part of the web tier components rather we need to write in an POJO class do that any class in our application can use this logic.

Note:

🡪The servlet behaves as singleton and POJO is non-single tone class bcz for every req servlet need to create the different obj for POJO otherwise race around condition will occurs so we need to declare a POJO class as local variable in a service() method so that for every req there will separate thread execution of POJO will be created hence POJO class obj is thread safe.

Initially there are no design patterns so POJO classes are used to write persistency and business logic but later Business delegate, service and Dao design patterns comes into picture where each and every logic independent on each other.

**Re-designing the Servlet + JSP Architecture using Design patterns:**

🡪Given a chance how can you re-develop/re-design the servlet + JSP Architecture?

We should not say directly with Dao, Business delegate rather we need to say if the business logic is more that is there in the POJO class that it self tells we should not use Servlet + JSP architecture rather we need to go for the Frame work like Struts and Spring.

That means if Business logic and Persistency logic is simple then we can use servlet + JSP but if the application logic more complicated then we need to re-design the our presentation tier tech to presentation tier frame works like Struts + JSP.

**2. Struts + JSP Based Project Architecture:**

We need to tell to the interviewer even though we don’t have Experience on the Struts I know the Struts Architecture as well bcz while learning spring my self-engaged to know the drawbacks of Struts and benefits of the spring, so given a chance myself I can explore amd can work on the Struts as well.

**🡪Purpose of Struts FW:**

Servlet is an API so that it cannot provide the boiler plate logic so most of the times developer has to write the boiler plate logic to handle the req, so that’s where to avoid this problems frame works comes in to picture.

🡪Struts provides Boiler plate logic like

1. Request-Wrapping Logic

2. Form Handling or Form management (Validating and accumulating the same field’s data back)

**1. Request-Wrapping Logic:**

class EmpInfo {

private String firstName;

private String lastName;

private String gender;

private String dob;

private String age;

//setters & getters

}

class EmpDao {

public void saveEmp(EmpInfo empInfo) {

}

public int insertTechDetails(techDetails) {

}

}

class EmpRegServlet extends HttpServlet {

public void service(req, resp) {

String firstName=null;

String lastName=null;

String age=null;

String dob=null;

String gender=null;

EmpInfo empInfo=null;

firstName=req.getParam("firstName");

lastName=req.getParam("lastName");

gender=req.getParam("gender");

age=req.getParam("age");

dob=req.getParam("dob");

//req wrapping

empInfo=new EmpInfo();

empInfo.setFirstName(firstName);

empInfo.setLastName(lastName);

empInfo.setGender(gender)

empInfo.setAge(age);

empInfo.setDob(dob);

EmpDao empDao=new EmpDao();

empDao.saveEmp(empInfo);

}

}

🡪Reading the data from the jsp page and populating the data in to an obj is called as "req-wrapping".

🡪Similarly if have multiple Jsp pages then we need to write the same type of req-wrapping logic in all the servlets of our application which is called as boiler plate logic which is shown below.

Boiler plate logic:

Duplication of the coded is not a boiler plate logic rather it an similar kind functionality or common functionality that we need to write in all the classes or in all the presentation tier components of our application or across the application/projects is called as Boiler plate code.

class GetEmpServlet extends HttpServlet {

public void service(req, resp) {

String empId=0;

String dept=null;

String domain=null:

String salary=0.0;

TechDetails techDetails=null;

empId=req.getParam("empId");

dpet=req.getParam("dept");

domain=req.getParam("domain");

salary=req.getParam("salary")

//req wrapping

techDetails=new TechDetails();

techDetails.setEmpId(empId);

techDetails.setDept(dept);

techDetails.setDomain(domain);

techDetails.setSalary(salary);

EmpDao empDao=new EmpDao();

empDao.insertTechDetails(techDetails);

}

}

🡪For example if we have 3-jsp's and each and every Jsp will have 30-fields then we need to write the req-wrapping logic for each and every Jsp in each and every servlet which is common type logic in all the servlets which is called as boiler plate logic.

🡪If we observe/if we maintain the req attribute and its corresponding attributes of class are similar that means the attributes of the POJO class and the req attributes name (input control names of JSP) are same so that we can automate this logic as follows.

class EmpRegServlet extends HttpServlet {

public void service(req, resp) {

EmpInfo empInfo=null;

Map<String, String[]> paramMap=null;

//we are takes [] bcz 2- input controls may contain same name

paraMap=req.getParamMap();

//used for to get all req params in one single map

empInfo=RequestWrapper.wrap(paramMap, EmployeeInfo.class);

empDao.saveEmp(empInfo);

//save then bind the resp to req scope

//forward the req to jsp

}

}

class RequestWrapper {

public Object wrap(Map<String, String> paramMap, Class classType) {

Object object=null;

object=classType.newInstance();

for(String attributeName : paramMap.ketSet()) {

Method method=classType.getDeclaredMethod("set" + attributeName);

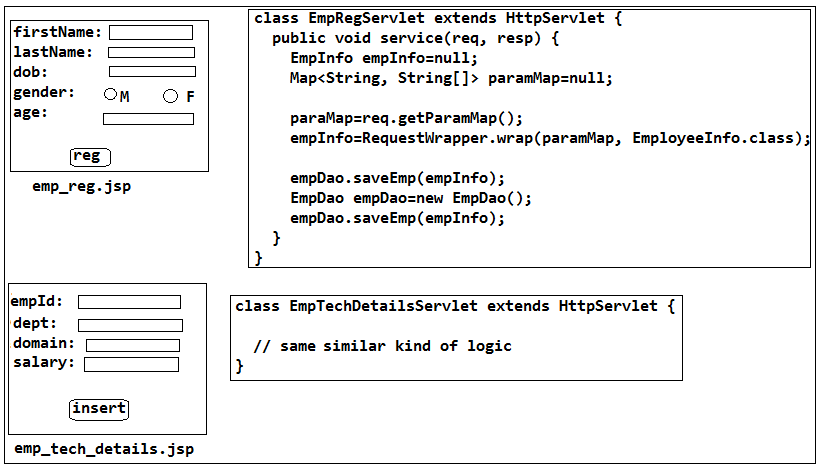
method.invoke(object, new Object[]{paramMap.get(attributeName)});

}

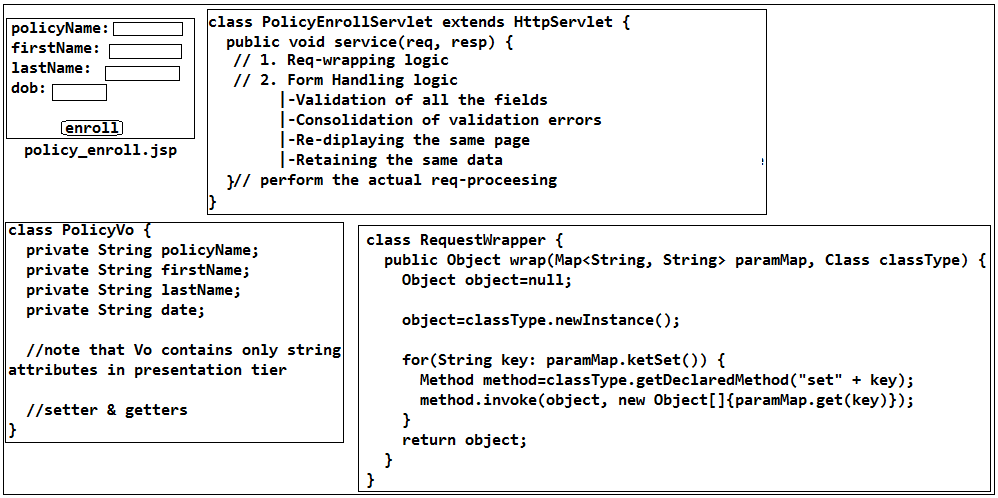
return object;

}

}



One more example:



🡪That means we are automating the reading and wrapping the req data in to an generic using reflection that means we are automating this process.

🡪So we are avoiding the boiler plate code in all the servlet by calling just wrap() that means we are automating by writing this logic. This type req-wrapping logic need to written not only in one project rather it need to written in each and every project that's where Struts Frame work comes into picture. So instead of writing this logic by the developers in each and every project, Struts has provided this logic so that we no need to write the req-wrapping logic.

🡪So just we need to pass which type of obj we want and req attributes to the Struts then it will gives obj of data directly.

🡪That means in order to automate this process of Req-wrapping logic in all the classes we need to use either of them given below Frame works, which are called as Presentation tier tech component.

JSF

Struts

Spring MVC

**2. Form Handling or Form management (Validating and accumulating the same field’s data back):**

class EmpRegServlet extends HttpServlet {

public void service(req, resp) {

EmpInfo empInfo=null;

Map<String, String[]> paramMap=null;

paraMap=req.getParamMap();

empInfo=RequestWrapper.wrap(paramMap, EmployeeInfo.class);

empDao.saveEmp(empInfo);

EmpDao empDao=new EmpDao();

empDao.saveEmp(empInfo);

}

}

🡪For example if we specified in DB for first name as unique constraint, so if the user has submitted the data and which is already there in the DB so it will through an SQLException stating the constraint violated. So now the Servlet should not display the error message stating that name already existed. bcz if first name is already existed in DB then there is no need to passing the req data to the Business logic to perform the logic to processing the data directly that's where we need to perform the validation before going to process the req by the Business logic. That if the req data is invalid then there is no point in processing the req and then displaying the error messages that is the reason servlet need to do the validation on data before the req is going to passed to the business to process so that we can avoid the performance issues and processing of wrong data and we can avoid Un-necessary Exceptions like ConstraintViolationExceptions.

🡪To validate and populate the errors at one single shot we need on List

List<String> errors;

empInfo=RequestWrapper.wrap(paramMap, EmployeeInfo.class);

if(empInfo.getForstName().length>10) {

errors.add("firstname canot be more thenn 10 chars")

}

if(empInfo.getLastname().length>10) {

errors.add("lastname canot be more thenn 10 chars")

}

if(errors.size()<0) {

//that means no erros

//so f/w the req to business req processing

//display the success page

}

🡪For example we are writing logic to reg an email Id and there is an existing email Id and it is throwing an Unique ConstraintvoilationException so how do u handle this case can u plz explain me?

We should not say we will display the error messages like that rather we need to say actually this type of Exception messages should not come rather we need to prevent this type messages for this we need to perform the validation once we wrap the req data in presentation tier logic before going to process the req so that we can avoid this type of issues and performance issues and we need display the error messages name already existing.

🡪Once errors raised then we should not display with empty fields with errors messages in the user interface rather we need to display with entered data and along with error fields, bcz the user interface may contain 50-fields so the user may feel in-comfortable again to fill all the fields that is the reason we should not display another jsp page, so that is the reason we need to accumulate the data entered data as back to the same jsp page along with error messages so that he correct error fields and he can quickly submit the req again so that he will feel comfortable.

🡪In order to do this the servlet need to know form which jsp page the req is coming so jsp page can sent an hidden form field in jsp page so that servlet can get the hidden form field and we need to write the logic to populate req field data in the list using <useBean> tag the and bind to req scope so that servlet can re-display the same data back to the same jsp page.

🡪That means the validation logic and accumulating the same data back to the user along with error messages if there exist any validation errors need to written by the developer in all the servlets of application or across the projects which is called as boiler plate logic.

🡪Here the target is not only validating the logic even get back to the response or errors if any to the client so that is the reason we need to write validation logic in the presentation tier component (ex: servlet) bcz it can interact with the client and it gives the errors back to client if any errors are there.

🡪So instead we writing this boiler plate logic and to avoid this logic by the developers Struts is provided this type of validation logic.

🡪So we need to write one class for for which we wanted to apply the validation and if we give this class validate to the Struts it will takes care of everything.

🡪That means Validating all the fields of req data logic and accumulating the same field’s data back to same jsp of the user page which is called as Form Handling or Form management.

For example if we have

1. Authentication validation logic,

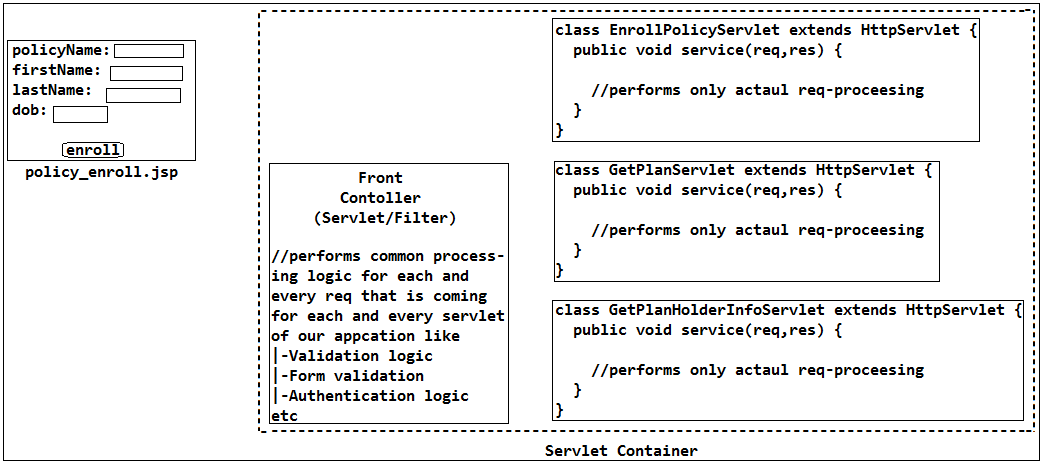
2. Form validation logic

3. Req-wrapping logic need to be write in all the presentation tier components

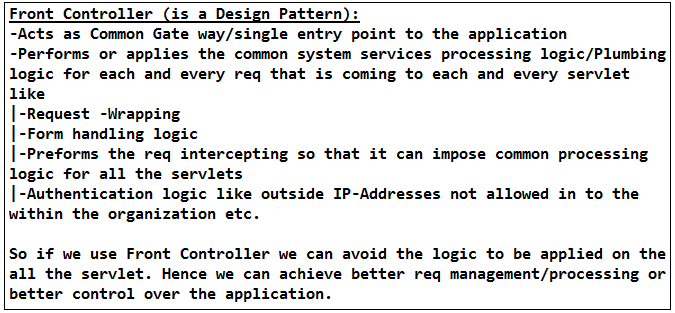
Instead of we writing or we calling this type logics in each and every servlet we can write in one web-component (either in the servlet or Filter) which can apply for all the servlets of our application or for other aplications so that actual business servlet free from the this type common processing logic so that actual req processing will taken by actual req processing servlet and common processing logic will takes care by another web-component so that actual req processing servlet perform better that’s where Front-Controller comes into picture which is component in the Struts Frame work.

The Front controller may be a servlet or may be a Filter bcz these can process the req but should not a Listener bcz Servlet or filter will be called for each and every req so that the common logic will be applied or automated for all the req-servlets, but should not be an Listener bcz listener is not meant for perform the req processing (that means Listener will not performs the req process) be called once in While of the application. That’s where Structs Frame work come with an Front Controller design architecture.

**Purpose of Front Controller:**



🡪 If we observe the architecture the req should not come directly to the servlet bcz if req directly comes to the servlet the each and every servlet need call the req common processing logic that’s why the req should not come directly to the served rather we pass to the Front Controller before going to the forward the req to actual servlet that is the reason the servlet container will takes the responsibility of forwarding the req to the Front controller before going to forward the req to the actual servlet to process the req.



🡪Front Controller (is a Design Pattern and it may be either a Servlet or Filter):

-Most of the times Front controller will be taken as Servlet in the industry.

-Acts as Common Gate way/single entry point to the application

-Performs or applies the common system services processing logic/Plumbing logic for each and every req that is coming to each and every servlet like

|-Request -Wrapping

|-Form handling logic

|-Preforms the req intercepting so that it can impose common processing logic for all the servlets

|-Authentication logic like outside IP-Addresses not allowed in to the within the organization etc.

|-Auditing

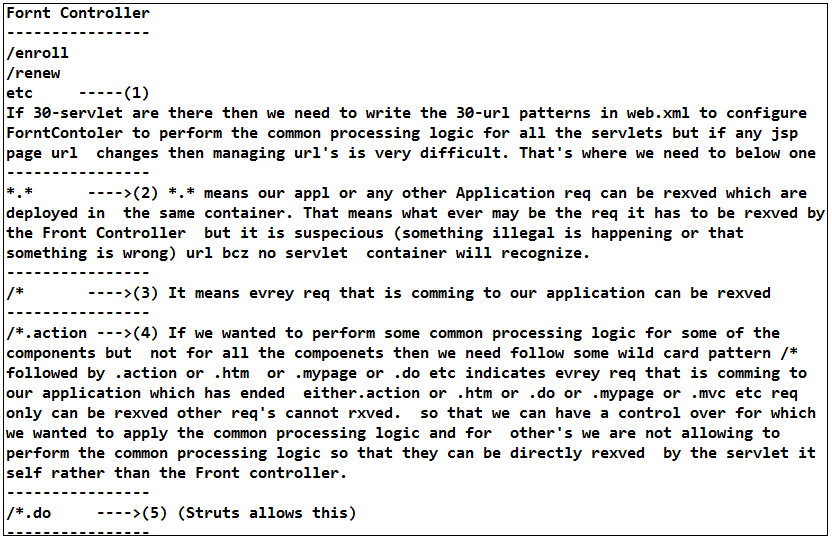
|-Logging etc

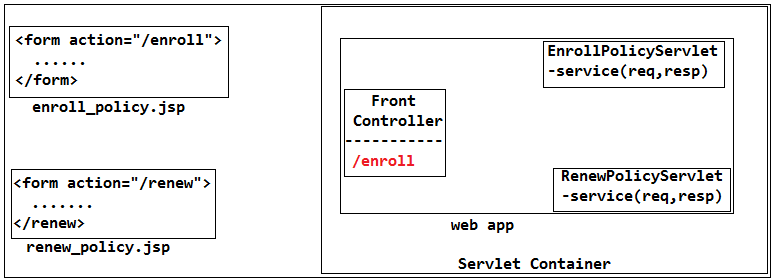
So if we use Front Controller we can avoid the logic to be applied on the all the servlet. Hence we can achieve better req management/processing or better control over the application.

Flow of execution:

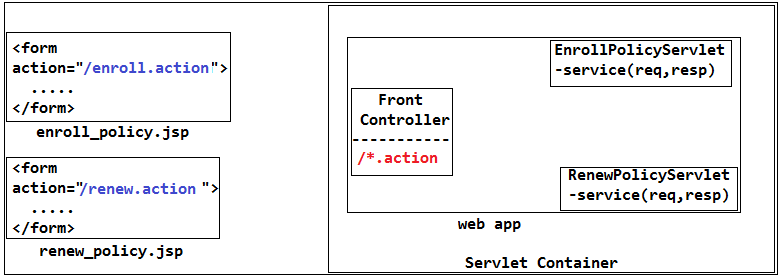
Whenever the req is coming the Servlet container it takes the incoming req URL and pics-up the context root and url pattern then goes to the web application deployment directory of the application and searches for whose context is path is matched with incoming root context that application then goes to the web.xml and searches for an servlet whose url pattern is matched with the incoming req url then creates the obj if obj is not there then forwards the req to that servlet but write now the req should not goes to the servlet directly so the servlet-container will forwards the req to Front Controller so front controller must be an web component then forwards to the servlet if common processing logic is success otherwise servlet container will not forwards the req to the servlet bcz if any Exception or error in common processing logic it sends error message to the user interface so that un-necessary req will not goes to the actual servlet to process the req hence better req processing can be achieved with help of Front Controller.

In order to achieve the common process logic to be applied on all the servlets we need to configure the Front controller with an url pattern that is coming as same as incoming url as follows.

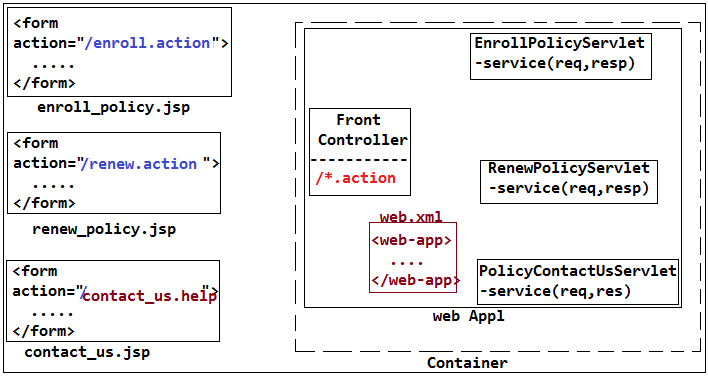




In the above diagram Front controller will rexves only enroll req and nenew req will be directly f/w to the Renew servlet.



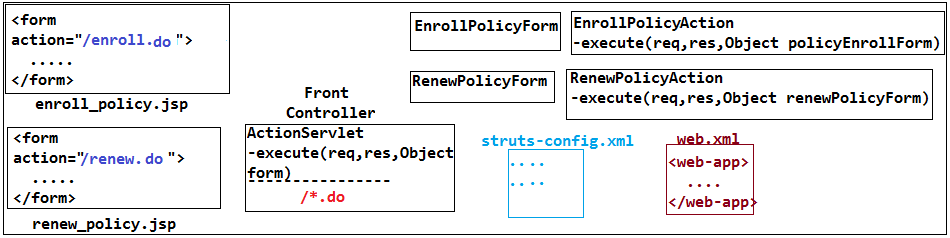
In the above diagram the req will be rexved by the Front Controller for both the req’s.

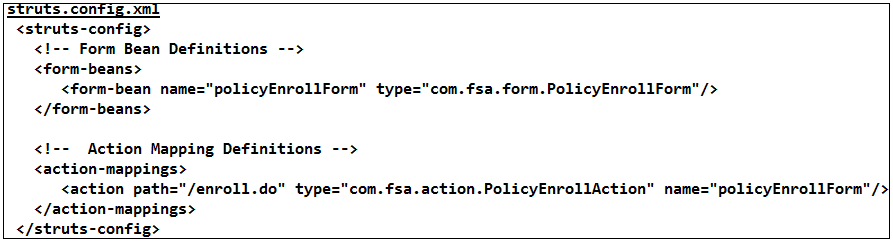


So now the once the req (enrol, renew) is coming to the container then req will forwarded to the Front Controller then but it will not forwards the req that is coming from contact\_us.jsp (/contact\_us.help) rather it will rderectly sends the req to the PolicyContactUsServlet bcz it does not matched with the /\*.action hence the req will be performed for which ever we wanted to apply.

🡪That means to perform the common processing logic for each and req we need one front controller so each and every frame need to provide this type front controller so struts has provided one class called as ActionServlet which will be acts as Front Controller.

**Internal Req Flow of Struts Frame Work:**



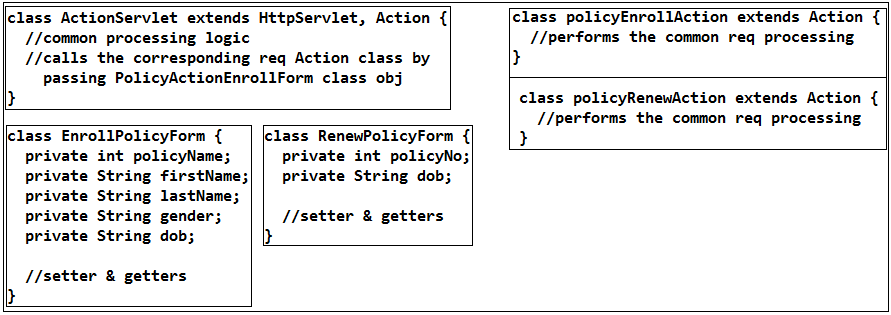


🡪The ActionController will performs the common processing logic that means the req has been already rxved by the Front Controller so it can call any Action class which perform the actual req and here we don’t need any servlet to rxve the req bcz ActionServlet already rexves the req hence there is no need of taking one more servlet to process the req bcz the front controller can call any class to perform the req processing. But Front controller cannot call any class to render data to process the req. That’s where our actual Action class need to extends form the Action class so that ActionServlet knows which methods has to be called bcz alredy FrontController also extends form Action so that it can call the method on the Action class (Note that Action is class in Struts)

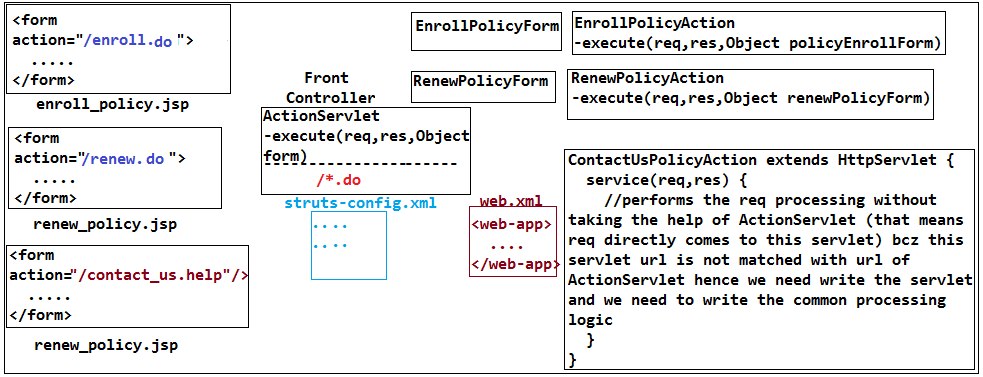
Once the req has been rxved by the Action Servlet it need to forward the req to the corresponding Actin class but it don’t know for which Action class it needs to forward the req, that’s where we need to write the struts-config.xml stating that for which Action class the ActionServlet need to send based on the incoming req.

Now once the enrol\_policy.jsp will sends the req then the req will be rxved by the ActionServlet and it will forwards the req based path we specified in the config.xml so that it takes type of the class creates the obj and calls the execute() but the Action will don’t know in what class the req form data has to wrapped bcz differernt forms will have different fields so we need to have multiple ActionForms so that ActionServlet can wrap the data in to the corresponding Form class that is the reason we need to write ActionForm class stating that for which form and for to perform which action we need to write corresponding ActionForm and need to configure this as <form-bean> in config file.

So once the req has been coming from the enrol>policy.jsp then the control comes to the ActionServlet then it will goes to the structs-config.xml and looks for the url matching with the path name then pics up the corresponding Action class and identifies the corresponding form class then creates the form class obj wraps the data into the form class obj then creates the obj of Action class obj and forwards the req to perform the req processing by passing req,res,formObj as params so that wrapping the data will be taken care by the ActionServlet.



If one of the form url is not matched with the ActionServlet then we need to forward the req to the corresponding servlet directly that means we need to extend from the servlet then only the req will be rexved to process the req hence we need to perform the common processing logic in that servlet by the programmer as shown in below diagram.

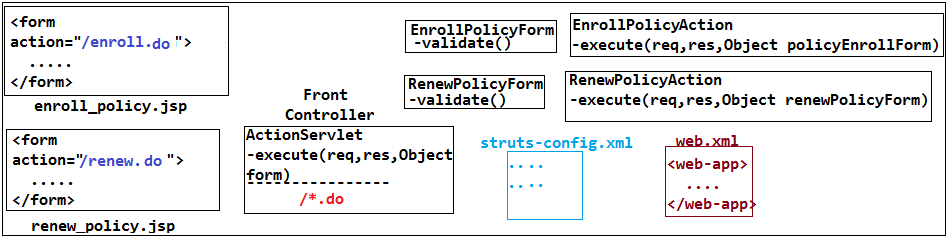


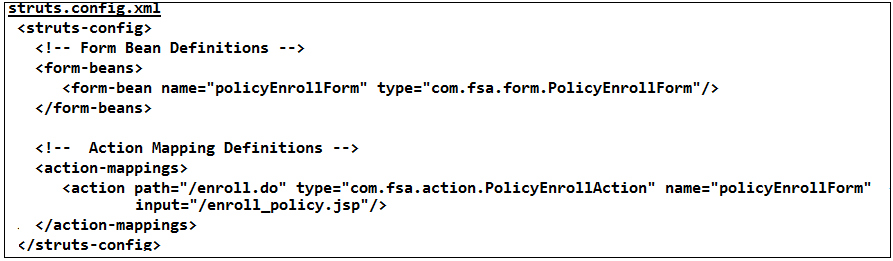
🡪The validation logic will be written in the of corresponding ActionForm by extending from the ActionForm class and override validate() so that ActionServlet will calls this method and we no need to write the exception logic and errors logic rather we need to tell the which type errors to the ActionServlet so that it will takes of everything and we need to write just validation logic in the ActionFrom class so that validation will done automatically by the ActionServlet for each and every req.

Note:

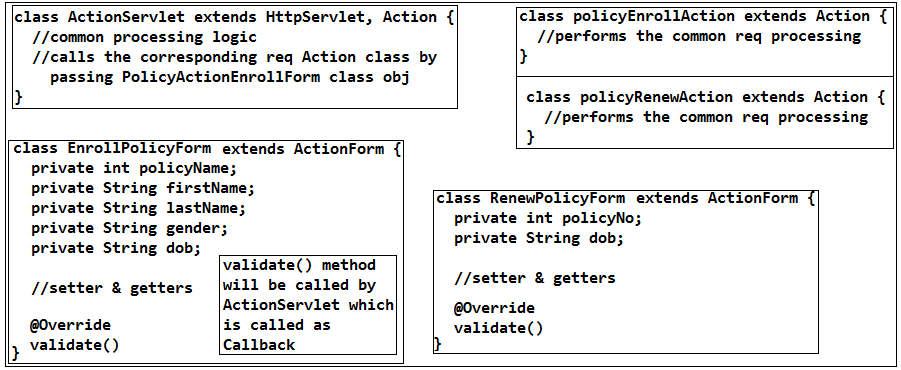
The validation logic is different for each and every form of different jsp pages so we need to write the corresponding validation logic in the validate() but we no need to write the exception logic to render and displaying same data to the user all thses will takes care by the ActionServlet for each and every req that is coming to the Action class.

**Final Flow:**





🡪In the above config file we specified <input=”enrol\_policy.jsp”> will used by the ActionServlet to give the error message, <forward=””> will be used to forward the req to corresponding jsp page.



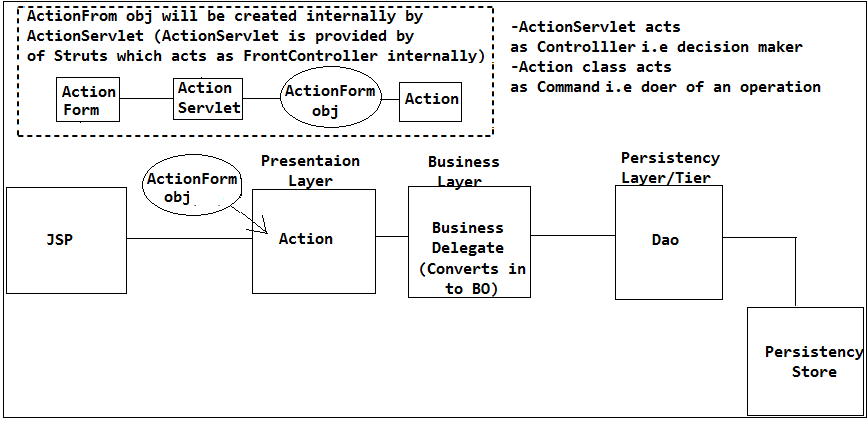
**Struts + JSP Based Architecture:**

🡪Can u plz Struts Frame work flow?

We need to tell the answer for this as FrontConroller (ActionServlet) explain total above.

🡪Can u plz Struts Project based project flow or what are components are there as part of ur application in Struts based Arch?

Now explain which is discussed below.



We should not write the Business logic in Action class bcz it is an presentation tier tech specific component if we switch from presentation tier tech (Servlet) to another tech or FW (Struts) then Servlet will be replaced with Action class or if we moved to spring Controller will be there that means again we need to write the code that is the reason we should not write Persistency and Business logic as part of the presentation tier components that's where Business Delegate comes in to picture.

**🡪Purpose Dao (Data Access Object):**

🡪Dao layer is a class which can perform the persistency logic operation and we can write any Dao's as part of our project.

🡪If we write the Business logic and Persistency logic as part of Business Layer then our Business logic will exposed to specific persistency store from which data is coming from.

🡪So if we wanted to switch from one persistency store to another persistency then Business logic and Persistency has been tighly coupled together so we cannot easily differentiate business and persistency logic to switch from one persistency store to another.

🡪That means always the BL should be independent of Persistency store so that if any changes are happen in the persistency then our BL will not gets impacted bcz BL always sends obj as a input Dao and takes object of data from the Dao that means the BL will not know from which persistency store the data is coming from hence Dao is mandatory otherwise our BL will be tightly coupled with persistency store.

🡪Advantages of Dao:

1. Abstracts the persistency Storage:

It ensures that none of the classes of our application has been exposed to the finite details from which persistency store the data is coming from. That means it acts as an layer in abstracting the rest of the classes in accessing the data from underlying persistency store (which may be DB, file or xml anything). So if any changes are done in persistency store then only Dao has to be modified but rest of the components are independent.

2. Abstract form persistency store vendors:

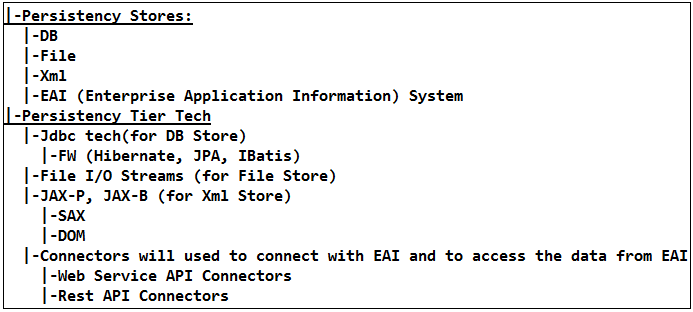
If we switch from one persistency storage vendor (Oracle DB) to another persistency store vendor (MySql DB) then only Dao gets modified but other classes of our application will not gets impacted.

3. Abstracts form the accessing tech:

If we wanted to switch from one persistency access tech to another persistency tech (like Jdbc, Hibernate, Jpa to access form DB and JAX-P (SAX or DOM), JAX-B for Xml) then we no need to do any changes in the business layer rather we need to change only in the Dao.

Note:

Current Version jdbc-4 which supports for the xml.



**Purpose of Business Delegate:**

**🡪Reasons for need going for Business Delegate:**

1. Managing and handling the Exceptions (converting tech Exception to Application Specific Exceptions)

2. Conversion of Web Application representation tier format values in to Business tier representation format values. (VO to BO to persist and BO to VO to render the response to Action class to JSP)

3. Managing and ensuring the transactions will be imposing across all the Dao’s.

**🡪1. Managing and handling the Exceptions:**

Assume that the Action class directly talk to the Dao without the help of Business Delegate then

a) For example the end user submitted the req to buy the products and payment has been happen and Dao will persist the data in the Orcle DB using Jdbc and if any Exception (Generally it is SQLException) has been raised while persisting the data in the DB then the Dao should not handle the Exception rather it need to throw the Exception bcz to display the friendly error message to the end user so that they can know the problem for which it is happing. So if the Dao report the SQLException to the Action class now the Action class should not throw the Exception to the end-user rather it needs to the display the proper error message to the end user so that they will feel comfortable.

class OrderProcessDao {

try {

// perform the persistency

} catch (SQLException se) {

throw se;

}

}

class BuyOnlineProductsAction extends Action {

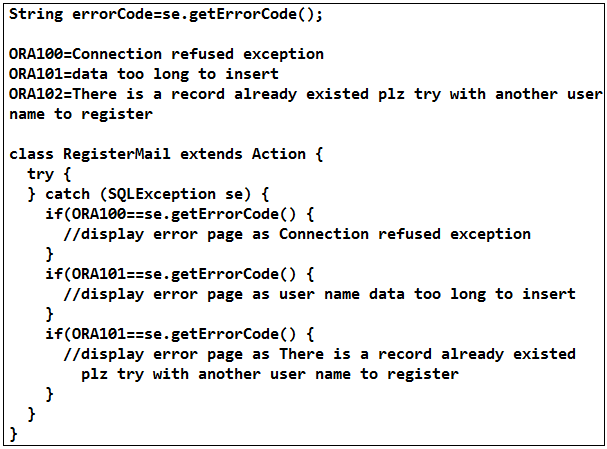
try {

} catch (SQLException se){

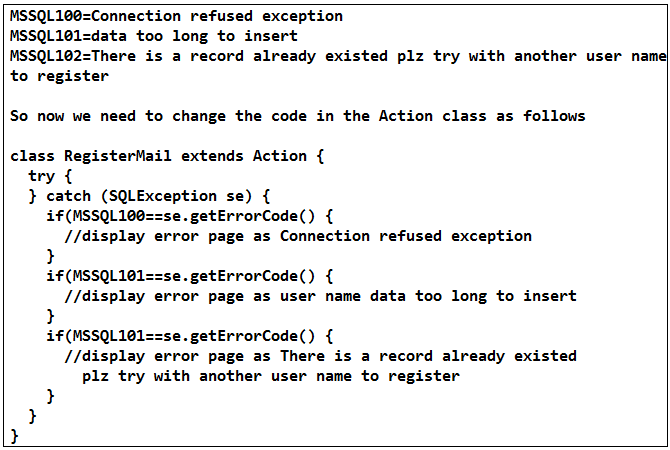
//display the proper error message using error page OOPs! Unable to process the order And your money will credited back within 24-hours.

}

}



If we switch from one persistency store vendor Oracle to another vendor MSSQL then error codes will be changed.

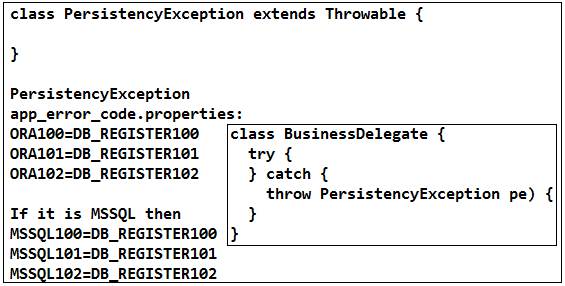


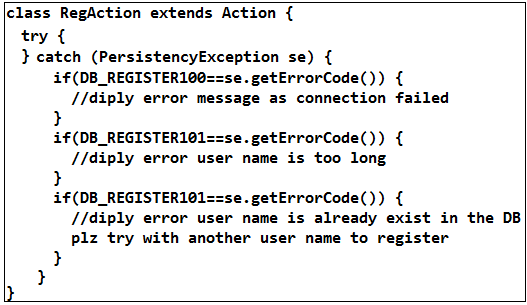
That means the presentaion tier components are exposed to the persistecny tier details so if any persistence tech are modified then we need to modify the code to display the friendly error messages so our Action class has been tighlty coupled to the persistency tier details.

So the Action class should not call directly the Dao that's where we need to use Business Delegate.

🡪If Business Delegate reads the SQLException then Business Delegate will tightly couple to the Persistency tier Exceptions and if any changes are happens in the Dao from one DB to another DB or one persistency store to another persistency store then also it will tightly couple to that specific persistency store Exceptions.

So in order to avoid this problem we will create a Generic Exception which will takes care makes our Business Delegate free from persistency tier. So now Business Delegate will takes care of wrapping/translating the SQLException into an Generic Exception called as PersistencyException.





If any changes are happen in Dao form one DB to another DB also there is no problem in Action class bcz it is not directly exposed to the persistency Exception rather it is referring the Application specific user defined Exception hence bcz of Business Delegate takes care of translation of SQLException into application Exceptions Action class if free from the persistency specific details.

That means if there is a change in the Persistency tier then Business Delegate will takes care of translating/wrapping the SQLException in to an Application/User-defined specific Exception so that our presentation tier component will not gets impacted.

Note:

Unless until the Exception thrown by the Dao and Business Delegate the Action class cannot handle it. If Either Dao or Business Delegate handles without throwing Exception then Action class will not the Exception to display the error message.

🡪We can write the Exception translation logic which is an Application specific Exception logic in the Dao as well but if we should not write it Application specific Exception is an Business Exception but not the persistency tier specific Exception bcz if business/application specific Exception is modified then persistency tier need to modify.

🡪Can we write if condition in a Dao to persist the data in a table?

We should not write any if conditions to which treble the data has to be store, always in which table the data will be stored is decided by the Business delegate.

🡪Where and how r u type casting the java.util.Date in to java.sql.Date such type of conversion logic?

We should always write the conversion logic as part of the Delegate but not in the Dao and Dao will always takes entity/business obj to persist the object of data in the DB.

**Note:**

We not only throw the Exceptions rather we need to trace Exceptions to identify what type of problems are going to happen as part of application. That means we need to log the Exceptions and well as we need to render the friendly error messages to the end user that’s where we need to wrap the same Exception trace in the another Exception which is called as Exception wrapping or translating. This is one of the great feature in java where we can trace/preserve the context of the Exception from beginning to the end. That means one Exception can be wrapped in to another Exception to preserve the context of the Exception from the beginning to end. That is the reason while writing the User defined Exception we are extending from the Throwable and we are passing our Exception as input the super class by writing the Excretion in the constructor so that our Exception will wrapped by the super class Exception.

class MyException extends Throwable {

public Except() {

super();

}

public Except(String message, Throwable cause) {

super(message, cause);

}

}

🡪How do u propagate/wrap the Exception that you have encounter with in the Delegate?

We should not say directly create new Exception class and we will through it rather we need to say We wrap the Existing Exception into Application specific Exception to preserve the context of who is actually causing the Exception then it will throw to the Action class.

Wrapping one Exception into another to know the technical details of the Exception we need Exception warp/translation so that we can log for tech developers, but we will not be working on the product rather are working in an project so we need to display unable to process the products list plz try again later. That means mapping technical errors to business errors which is called as Exception mapping has to be done for end user to display friendly error messages.

🡪How do handle Exceptions and what are the application specific Exceptions that u r using as part of your application?

(OR)

What are the application specific Exceptions that you are using as part of your application and where are trying to takes care translating/wrapping in to the application Exception tier and within which layer of application layer r u working on?

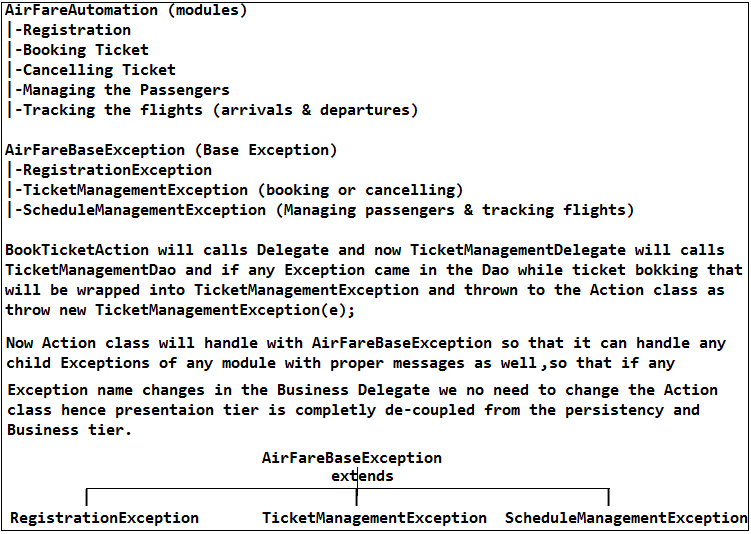
(OR)

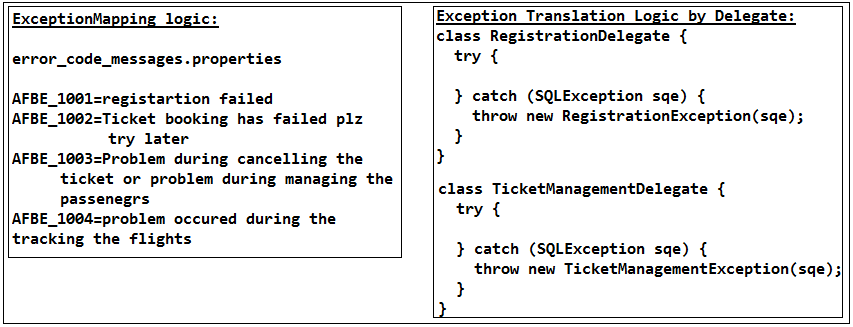
Wrapping one specific Exceptions in to Application/User-defined specific Exceptions:

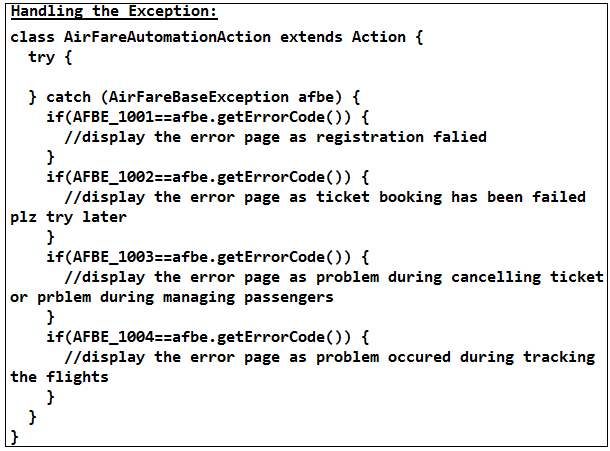
Application Exception is the one which will represents Exception information in an Application format rather than technological format so that Oracle or MySql or MSSQL-Server ultimately the information will be carried in an Application format, hence it is called as Application specific Exception.

Based on the technical errors that we wanted to handle we may can create multiple Application Exception classes which will gets translated as art of the Business Delegate into an tech neutral Exceptions.

Example with project modules:





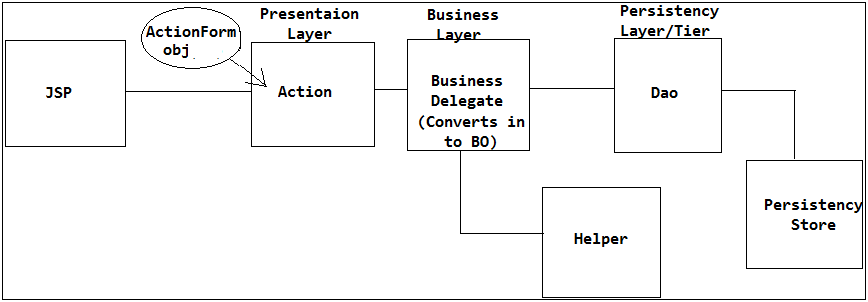


🡪If we switch from the Jdbc to Hibernate/Jpa we will get HibernateException /JpaExsception so that we no to change the Presentaion tier class Exceptions bcz it is referring to the Business/Application specific Exception so we need to change only in the Business for translation of Exception form the SQLException to Hibernate so that our presentation tier will becomes un-effected.

🡪We may think the Business layer will gets impacted bcz of the Persistency tier tech changed bcz it is translating the Persistency tier Exceptions to Business Exceptions, so we may think we can write this Exception translation logic in the Dao but we should not bcz if we write the Exception translation logic in the Dao then if persistency tech has been changed then we need to change the Exceptions in all the Dao’s which is an overhead solution that is the reason we will write business Exception translation in the Business Delegate so that we can easily change at one single place.

🡪What is application Exception and how to do u create your own application Exception as part of your application?

For all the Exceptions that are there in our project there will be an one base Exception class form which all the Exception classes are going to extends so that our Action can receive any Exceptions so that if any modifications done in the Exception classes also there will not be any impact on the Action class bcz it is referring Base Exception.



For example we have requirement to read the data from the xml then we need to use the JAX-P or JAX-B API then it will write one helper class to read the data from the xml and it will throws SAXParserException so that if we write the Exception translation logic in the Dao there if we throw to this business xml Exception also in the Dao then our business and persistency will becomes tightly coupled that is the reason we shouldn’t write the Exception tarnation logic n the Dao rather we need to write this in the Business Delegate only.

🡪Why action class itself should not perform the persistency and business operation?

Action class is an presentation tier Struts specific class so we should not write the BL and persistency logic as part of the Action class bcz if we switch from one presentation tier frame work to another then again we need to re-write the BL in the another frame work presentation tier class that is the reason we should not write the BL and persistency logic as part of the presentation tier rather we need to write in the Business Delegate.

**🡪Handling the Exceptions as part of our application:**

(Or)

How do you handle the Exceptions as part of your application?

We should not tell we will write one user defined Base class Exception from which rest of the user defined Exceptions will extends so that we can display the user friendly errors messages to the end-user.

He is asking how do you handle the Exceptions as part of your application that means he expecting how do you ensure there will not be any Exceptions or how do mitigate (avoid) the exception as part of your application. Rather we need say like this we ensure and we designed the class in such a way that where we wanted to minimize or eliminate the possibility of the Exceptions as part of our application by

(i) Validating the each and every filed of the of jsp page when the end user has submitted the req so that our business application we not get any invalid data so there will be less chances getting the Exceptions, so that we can eliminate like NuberFormatExceptions.

a) Username is have stuffiest characters or not

b) Password will contain the numbers and special symbols or not

c) If he is already existed person validate the password.

d) In numbers fields he entered number or string this type validation we had done.

Bcz of validations we will eliminate the invalid data flowing in to the Business tier class of our application so that we can avoid some sort of Exceptions raised by the business tier.

(ii) We should not we will never get the PrimaryConstaintVoilationException as part of our application so in order to eliminate this type of Exceptions we need to do stringent validation that’s where we need to check the user is already exist in the user so that we can display an error message to the end-user so that we can completely avoid the PrimaryKeyConstaintVoilationException as part of our application.

**Handling the runtime Exceptions:**

We can handle checked Exceptions by try and catch but there could be some more Exceptions which are unhandled Exceptions (unchecked Exceptions or Run time Exceptions) then such type exceptions will not handle by with the try and catch so this type exceptions will not be reported to the Delegate or to the Presentation tier rather these Exceptions will directly get by the application default Handler so it will displays will be displayed at the end-user as ugly error messages, so avoid this type of Exception will displayed at the end-user side we need configure **error page configuration in the web.xml** so that we can display an Generic error messages (like internal problem plz visit later) for retime errors so that end-user will not gets any impact.

That means if the exceptions are in our control like checked exceptions we will display the corresponding errors messages by converting the checked exceptions in to applications specific exceptions and if the things are not in our control like unchecked exceptions like runtime exceptions then will display the generic errors messages which is configured as part of the web.xml

🡪How can u handle the run time exceptions?

(Or)

🡪What is error page when to use error page configurations?

When there are the Exceptions that are going to occur at the run time then those will not be managed by the application components, that if any exception that are not managed by the application components then those are called as run time exceptions then only we need to configure the generic error page in the web.xml which will gets display the generic error message for all the run time exceptions rather than the ugly stack trace. The error is not meant for the checked exceptions it is meant for the unhandled/unchecked exceptions like run time exceptions.

🡪Did you got any primaryKeyVoilationExceptions As part of your application if occur how did u handle this?

Actually we should not get such type of Excerption rather we need to avoid/eliminate this type Exceptions by checking the DB whether the data is already exist or not so that we can eliminate this type Exception permanently as part of our application. So we need to write this checking and validation logic as part of the ActionForm class validate() method so that our application can free from such type of Excisions so that strength of the application becomes more.

**Validation are 2-types:**

1. Client side validations (Java Script and JQuery)

2. Server side validations (we need to write the validation logic as part of classes and these will triggered by the ActionServlet in the validate() method)

🡪We should not use both the validations it always depends on the nature of the application we are developing.

1. If we are developed the Intranet applications then there is no need of validation logic will be written server side apart from validating the credentials bcz the if it is an internet then there will not be any chance to used by the other public so that we can impellent Client validation easily so that we can avoid the network traffic as part of our application.

2. If it is an public network application then we need to do every validation at the server side as well bcz the hackers may change the req bypassing directly to the application or by so then our application will becomes raises the exception and if like this req’s are coming more then our application will becomes crash due to the unnecessary stack traces as part of our application which leads to poor performance of the application. That is the reason we need to do server side validation if it is an public network application so that we can avoid un-authorized req’s coming to the application and if any invalid or un-authorized req’s comes we can reject by validating them our application will becomes safe.

That is the reason always server side validation are more secure than client. It is even recommended to implement nominal validations at client side so that we can avoid the un-necessary round trips between the server and client. If client validations are successfully done then we need to send an acknowledgement to the server stating that client side validations are done so that again nominal validation at server side will be eliminated with conditional exception so for this we need to set one flag at the client side and if this true then we should not do the sever side if it false then only we need to do the server side validation for nominal cases so that we can reduce the server round trips.

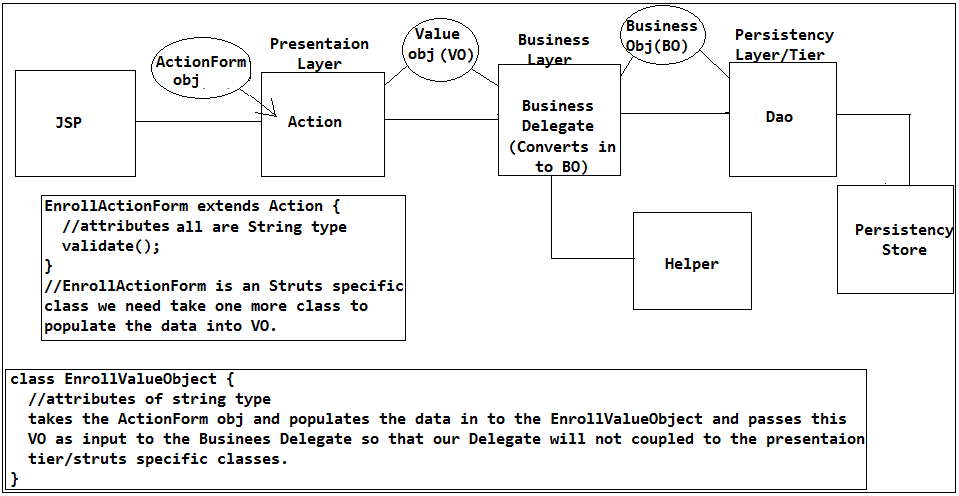
But most of the times we will prefer to validate server side again even though the client side validation are happen.

**🡪2. Conversion of Web Application representation tier format values in to Business tier representation format values (VO to BO to persist and BO to VO to render the response to Action class to JSP):**

ActionForm always carries the data that user has submitted from the jsp page and ActionForm will contains same no.of fields and same names of the fields as attributes that are coming from the jsp. ActionForm will carries the data in the form of String or may be in the form any primitives as well like int bcz ActionServlet will takes jsp field name and introspects to the ActionForm type attribute and it will converts in to the corresponding type as well.

But there is problem with this bcz if the user has submitted the age as string like “xyz” then ActionServlet will takes that age “xyz” and introspects the ActionForm age type and it will tries to converts the “xyz” in to an int but ir is not possible hence it will throws an NuberFormatException if client side validations are not done. Hence even though the ActionServlet has capable of converting the string to corresponding type attribute value we should not use this bcz it will converts correctly if the data is valid otherwise it will throws an Exception hence we should not take the BO from the ActionForm rather we need to take only as ActionForm obj as like string then we need to type cast to the corresponding type and we need validate the logic in the validate() and if any invalid data is found then we need to send back as a error filed so that ActionServelet will validates. Then we need to send the ActionForm obj to the Business delegate so that it will takes care of converting business data requirements so that it will pass the business data as to the Dao if Business Delegate is not there then there will not any conversion of ActionForm obj to BO hence conversion also need to takes by the Dao so that business data will be tightly coupled hence we need to use the Business Delegate to convert the ActionForm obj to BO.

But there is problem with the above scenario bcz the ActionForm obj is an Struts Specific class so if we give the ActionForm obj to the Business Delegate the our business Delegate will tightly couple to the presentation tier class. So in order to avoid this VO comes in to picture.



While converting the VO to BO we will not get any NumberFormatException bcz already validations has been completed in the ActionForm validate() method so we eliminated the chances of getting the Exceptions as part of our application. Hence in the code we will write the comment as NuberFormatException will never come in our application in the presentation tier itself. Hence there is no need of placing try and catch for the NumberFormatException.

🡪Once the data has been persisted in the Dao then Business Delegate will takes response BO obj from the Dao and it will converts that Business obj into VO and then passes to Action class now Action class will binds to req scope and passes to the JSP to render the response.

Ex:

Business debate will takes care of converting the Date to String to render the data as part of JSP that means BO to VO.

That means Business Delegate will takes care of converting VO to BO and BO to VO.

🡪Have hot any NumberformatException as part of your Business tier?

We should not get the NuberFormatException bcz we need to validate the data before the data is coming to the Business tier.

**3. Managing and ensuring the transactions will be imposing across all the Dao’s.:**

class StudentDao {

Connection con=null;

PreparedStatement pstmt=null;

boolean isSuccess=false;

public void svae(StudentBo bo) {

try {

con=DM.getConnection(driverClass);

con.setAutoComit(false);

//create thepstmt

//substitute the params and execute it

isSuccess=true;

} catch(SQLException sqe) {

throw new SQLException(sqe);

} finally() {

if(pstmt=!null) {

pstmt.close();

}

if(con=!null) {

if(isSuccess) {

con.commit();

} else {

con.rollback();

}

}

}

}

}

🡪How did u decide how many Dao's and how many Business Delagtes will be there as part of your application?

Within our there will be a Multiple Dao's and multiple Business Delegates. The no.of Dao's and will be decided based on the group of related functionalities or for every Use case that is there in the Project requirement document we will create the Dao's.

That means Dao's will be created based on the Use cases or related functionalities for example

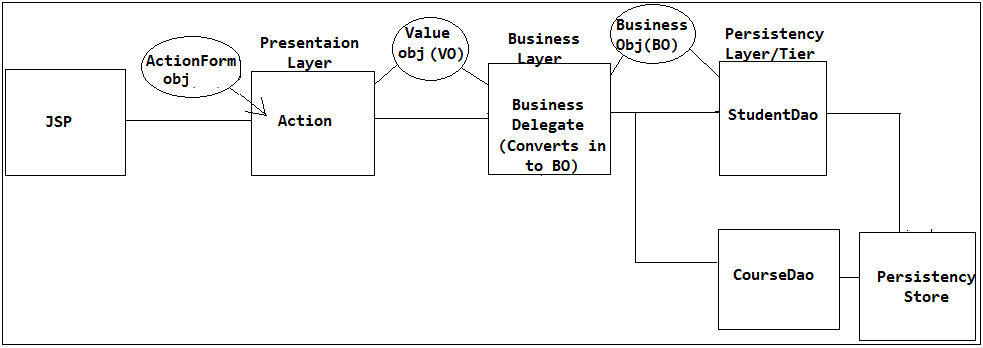
UserRegisterDao

Logging

PasswordAssistanceDao

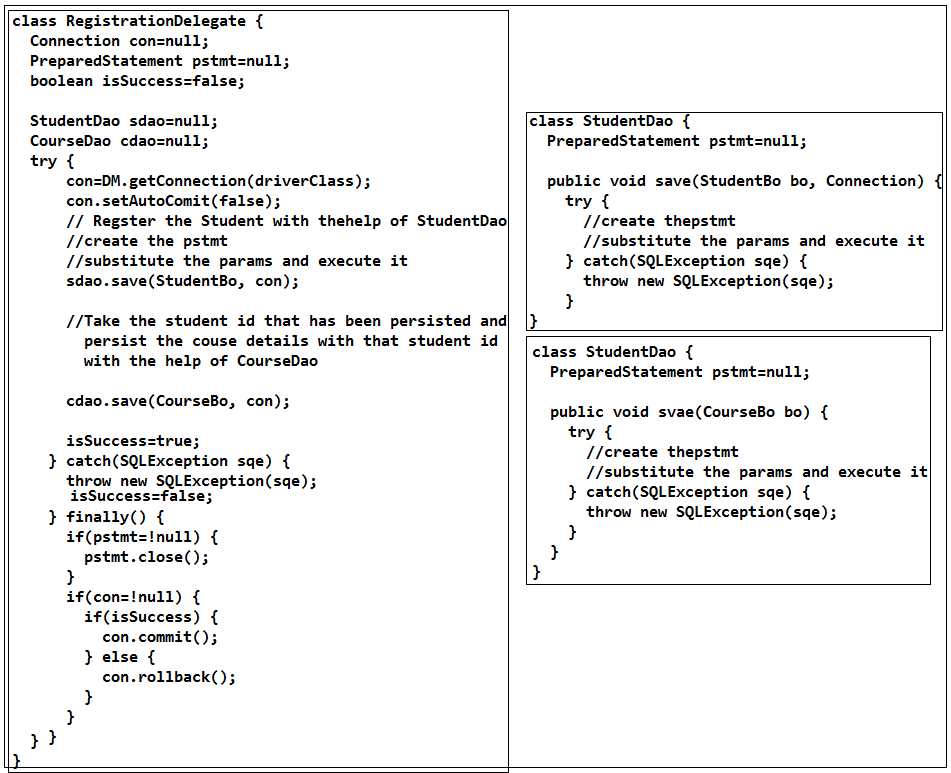
UserPreferencesDao etc.

🡪In an application Business Delegate not only talks to the Dao rather if needed it may talk to another Dao as well based on the use case that we have.



For example we have an use case where we have register the student with corresponding course, That means once Student Registered then with that id we need persist the to persist the course details.

Here we are writing the transactions (commit and rollback) in the individual Dao’s then if StudentDao is successful and committed in the StudentDao but while persisting the CourseDao exception has came then CourseDao will be failed so that Student got registered without any course bcz of individual transaction in individual Dao’s which is an invalid transaction. That’s where transactionality will imposed in the Business Delegate itself rather than writing the transactionality in the Dao’s to preserve the valid transactions. So that of Exception came in the CourseDao then both gets roll-backed bcz transactionality is in the control of the BusiessDelagte bcz Delegate itself passes the same connectio and maintains the transactions.



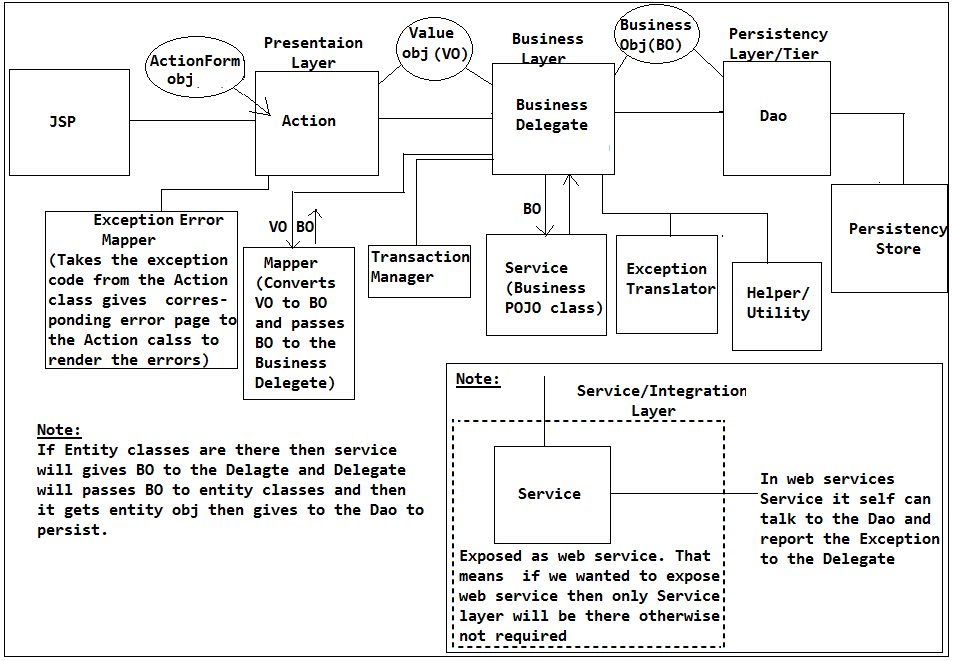
🡪Where do u write the business logic?

🡪In small scale applications only the Business delegate will be acts as Business tier and as well as delegation bcz to eliminate the more no.of classes and complexity.

🡪In general Business Delegate will not do any business operation if it is medium scale/enterprise application rather Business Delegate will acts as just delegator. That means we have one more POJO class called as Service layer which acts as Business tier and Business Delegate will acts as Delegator.

🡪Exception translator will be used to convert the tech specific Exception in to Application specific Exceptions so that Delegate will be called as Delegator and no more it is called as doer.

🡪Action class will gets the Exception and gives to the Exception Mapper this will gives the error page to the Action class so that Action class will displays the error page to the end-user.



That means the Delegate will gets all the Exceptions from all the classes and will gives to the Exception translator and then gets the translated Exception and gives to the Action class so now we can say the Delegate is will be worked as Delegator and it will not do anything.

🡪How are managing the transactions and where do u manage the transactions as part of your application?

In this way Transactions will be taken care by the Business Delegate in case of Struts. Explain the above Use case related our project.

Note:

In case of Struts + Hibernate explain as per our project.

🡪Can you plz explain the Architecture of the project?

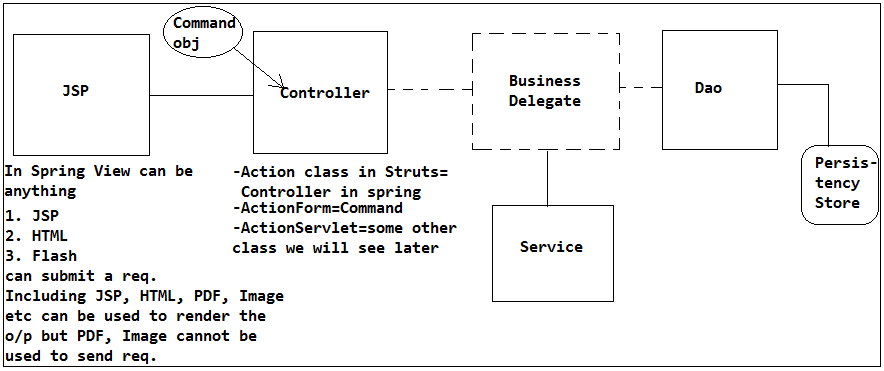
Iam the one whose working within the project from last 1.5 years and have been worked on the some module and the architecture of the application is almost similar so now I wanted explain the architecture of my project based my module, (so now we need to ask the pen and paper to draw the architecture) then explain the flow if it is Struts or f it is Spring and each and every component purpose.

Note that always talks about my project then only he will trust us we employee.

**🡪3. Spring + JSP based Project Architecture:**

🡪In case of Struts Frame work the view always be assumed as JSP only but Spring will not limits us to take as JSP only it even allows us to take HTML, Type space, Flash, PDF or it could be a image that means anything we can take as view.

🡪That spring is only the frame work who provided a flexibility of taking view as anything later Struts-2 will also adapted this concept form spring MVC.



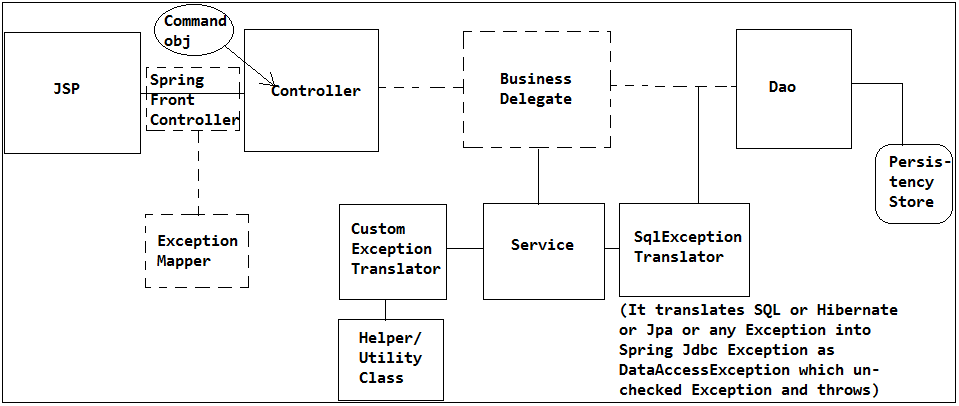
**🡪Reason’s to eliminate Business Delegate as part of Spring Architecture:**

🡪In case of Spring MVC once there has been received by the Controller the presence Business Delegate will not be there so the Controller can directly talk to the Service bcz

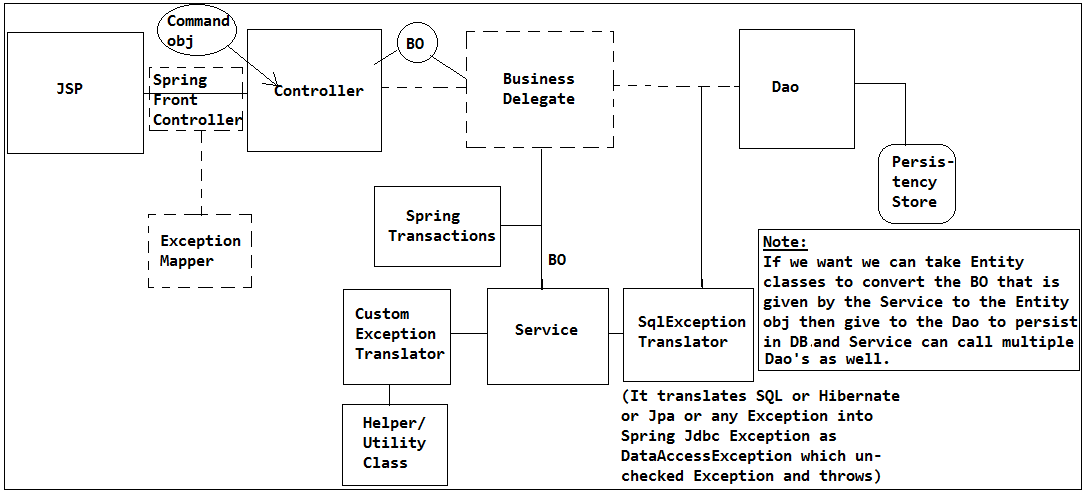
(i) Let us assume whenever the user submitted the req form the JSP the req will be received by the Controller now the controller will forwards the req to the Delegate and delegate will talks to the DB and if any Exception has been raised then Dao will throws an Exception to the Business Delegate to convert/translate the Exception in to the Application specific Exception then need to throw to the Controller so that Controller can display the error message to the end-user. But when we are working with spring MVC then there is no need of translation of Exception by the Delegate bcz whenever the Exception failed in the Dao the Spring-Jdbc will converts that Checked Exception into un-checked (Run time) Exception of spring specific DataAccessException. So spring Jdbc is an Un-checked Exception hence there is no need of writing try and catch to throws to the end-user. That means none of the classes need not to throw the Exception bcz it is a run-time Exception so it will be directly reported to the end-user. That means when Exception has thrown none of the classes will have the presence of the Exception that is the reason it is reporting directly to the end-user as ugly error message so in order to avoid this our controller need to write the try and catch block to display the appropriate error messages, but if controller is doing this job then controller will tightly couple to the persistency tier Exception of spring run time exception that where Spring run time (Spring front controller) has provided one component called as “**Exception Mapper**” so we no need to write the try cacth bcz it is an un-checked Exception so that Exception directly fed to the Spring Run time and it will takes care displaying the error messages but still we need to convert persistency tier Exception to the Application specific Exception and throws to the Spring Run time by the Delegate so in order to avoid this spring has provided one more component called as “**SqlException Translator**” which is which will translates the jdbc SqlExceptions or Jpa or Hibernate or IBaits or any Exceptions in to the spring specific Exception called as DataAccessException which is an un-checked (run-time) Exception so that none of the components will need not to throw by writing try and catch hence these will directly flows in to the Controller but in the controller as well didn’t written the try and catch that’s where Spring Run Time (Spring MVC component) will takes care of Mapping the corresponding Exception in to the error page and displays the error page to the end-user so we don’t need Business Delegate the is the reason Service can directly can talk to the Controller and Dao.

If we observe in the architecture the SqlException is sitting between the Service and Dao, If we have an helper class which is used by the Service to parse the XML then it will throws SAXParserException or PasrserConfigurationException to the Service the it will throws to the ForntController which is an tech specific Excerption bcz so now again we need to translate the this Exception into Application specific Exception so we need Delegate but in order to avoid this the spring has allowed us to write an “Custom Exception Translator” in between the helper and the Service class so that we don’t need Delegate and we don’t need to try and catch anywhere bcz all these are Un-checked Exceptions. So now if we give “**Custom Exception Translator**” to the spring run time it will take care to do this spring internally uses **AOP Throws Advice**. Hence we no need Delegate.

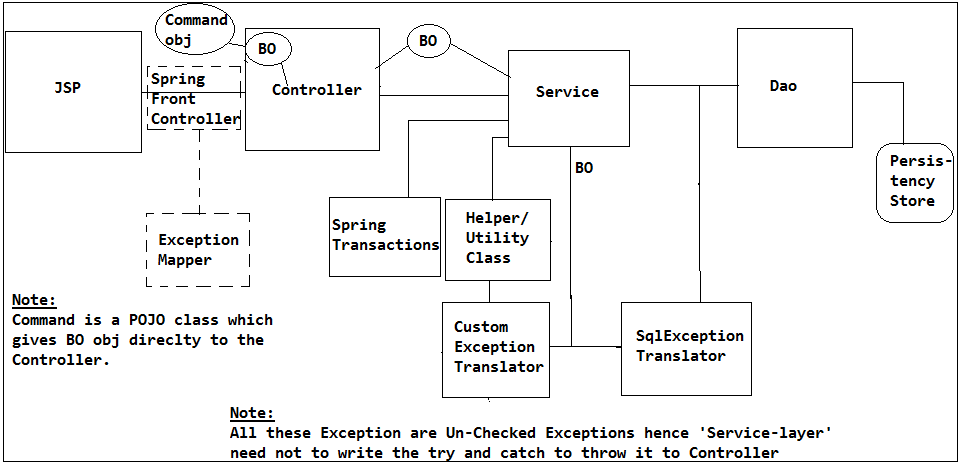
(ii) But still we need to use the Delegate bcz in struts we will get ActionForm obj which is an Struts specific class obj hence ActionForm will takes the POJO class and it will give the VO obj to the Delegate and our Delegate need to convert VO to BO and from if Dao given BO to VO to render the response to the JSP. But in case spring the **Command class is which is POJO class** hence it can be directly fed as input to the Delaget to convert VO to BO but the Spring itself takes care of providing VO to BO to the Service using “**Property Editors**” Property Editors are used to convert the JSP filed String data in to the corresponding primitive types and in cases spring we can use obj as well so that it takes care of converting JSP form field String data in to corresponding BO **without any need of VO** hence we no need any Delegate. But if any NumerFormatException is came while converting JSP fields to the BO the spring itself display the graceful (friendly) error messages to the end-user that means it performs validation error messages as well with same fields of data with proper valid error messages.



(iii) But still the transactionality is maintained by the Delegate so in order to avoid spring has provided Spring Transactions so that we completely avoid the Business Delegate. So now we need to impose the transactions as part of Service itself using Spring Transactions without writing transactionality logic by our own.



So by avoiding the Business Delegate we can remodel the Spring Based Project Architecture as follows.



That means our moto is now how to apply the transactions as part of the Service layer using the Spring Transactions.