**Spring Web MVC**

**Q) What is an enterprise application?**

* Enterprise means business organization
* Enterprises (business) provides business services to customer
* Any computer application that is used to computerize business Service provided by the enterprise is known as an enterprise application Or simply a business application

**Examples:** Banking application, Railway reservation system, Hospital management system, hotel management system … etc.

* Enterprise application can be developed using different languages like java, .net, PHP …etc.
* In java we can develop enterprise applications using AWT, swings, applets, servlet, JSP …etc.
* Enterprise applications are two types
  + Desktop based applications
  + Web based applications
* Desktop based enterprise applications can be developed using swings, applets …etc.
* Desktop based application services can’t be accessible over the web. So we can’t get business services 24X7. To get business services 24X7 we have to use web based applications.
* Web based Enterprise applications can be developed using **servlet, JSP**.
* Now we have to learn which is the best way to develop web based enterprise application.

Enterprise Application developing with only **Servlet** technology

Input.html => CalculateInterestServlet.java

**Input.jsp**

* Enter a/c no and submit the form.

**InterestCalculateServlet.java**

* Receives user input
* Validate user input (if any improper data, redisplay input.html with validation errors)
* Get connection
* Execute query
* Get balance
* Calculate interest
* Display interest

**Problems:**

* Servlet code is not reusable
* Web designing tools can’t be used
* Parallel development is not possible

Enterprise Application developed with only **JSP** technology

**input.jsp**

* Enter a/c no t send request to **calucateinterest.jsp**

**calucateinterest.jsp**

* Receives user input
* Validate user input (if any improper data redisplay input.jsp with validation errors)
* Get connection
* Execute query
* Get balance
* Calculate interest
* Display interest

**Problems:**

* JSP Code not reusable.
* Parallel development is not possible.

Enterprise Application developing with **JSP and Servlet** technologies

Input.jsp => InterestCalculateServlet.java => displayInterest.jsp

**Problems**

* Servlet code is not reusable

**To solve the above problems sun Microsystems introduced two design patterns for developing enterprise applications**

1. Model 1 (Page centric model)
2. Model 2 (MVC)

**Q.) What is design pattern?**

* Solution for the recurring problem.

**Examples:** MVC, DAO, singleton, factory, service locator, business delegate, session facade …etc.

* **Spring MVC, Struts, JSF, Wicket, ADF, Webwrok, Tapestry, Flex …etc** implements MVC design pattern.

**Q) What are the different kinds of Logics of web based java application has?**

* Any web based enterprise application has 4 logics.
* Presentation Logic
* Application Logic/Controlling Logic
* Business Logic
* Persistence logic/Data access Logic.

**1) Presentation Logic:** Code used to present the input and output is known as presentation logic.

**Note:** HTML, JSP, Velocity, Freemarker...etc technologies are used to implement presentation logic.

* A Real time web application JSP contains the following things.
  + HTML Tags.
  + JSP Tags
    - EL
    - Custom Tags
    - JSTL Tags
  + Web application Framework given Tags.

**2) Application Logic:** Code that performs the followings tasks is known as application logic or controlling logic.

1. Receiving the client Request
2. Capturing user input.
3. Performing server side validation on user input.
4. Controlling the flow of application.

**Note:** Servlet technology is best suited for application logic.

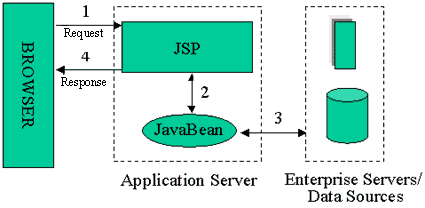
**3) Business Logic:** Programmatic implementation of business rules is nothing but Business Logic.

**Note:** Currently in the industry **java bean, Spring bean & Enterprise Java Bean(EJB), Web services**  used for implementing Business Logic.

**4.) Persistence Logic:** The code which is used to deal(read, write, update, delete …etc.) with database is known as Persistence logic or Data Access Logic.

**Note: JDBC, HIBERNATE, JPA, Toplink, JDO, Ibatis, Spring DAO, Spring ORM** …etc are used to implement persistence logic.

**Model 1 or page centric model**

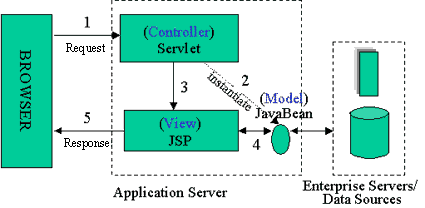


1. Request received by JSP sent by the browser. And capture the user input.
2. Create java bean object which contains business logic. And call the business method.
3. Java bean if required it will communicate with DB to get the required data. Hand over the processed data to the JSP back.
4. JSP will present the processed data to the end user.

**Analysis:**

* There is no clear separation of responsibilities. JSP acting as both controller and view. Only business logic is separated using java bean. But application logic and presentation logic is mixed in JSP.
* For small scale application development it is advisable to use. But now a day’s even small scale applications also developing with **MVC** only.

**Model 2 or MVC**



****

1. Controller (servlet) receiving the client request, capturing the user i/p and performing server side validations.
2. Controller invoking the business method of the model (java bean).
3. Model contacting the database, getting the business data from the database, processing the data by applying the business rules and holding that processed data.
4. Model giving back the response to Controller.
5. Keep the processed data in heap memory (request/session/servletContext).
6. Controller switching the control to appropriate view of the application.
7. View (JSP) contacting the heap memory (request/session/servletContext) and getting the processed data.
8. View producing the response (HTML) screen required for the end user and sending that web page via web server to the web client (Browser).

**Q.) What is the problem with Model?**

* Now model is having three responsibilities
  + DTO (Data Transfer Object)
  + DAO (Data Access Object)
  + Business Object
* Now a day’s model we will divided into three objects.

**Advantages of MVC**

1. Clear separation of responsibilities. That is Clear separation among presentation logic, application logic and (business logic + Data Access Logic)
2. Code reusability.
3. Single point entry for the application.
4. Easy to test.
5. easy for maintain and future enhancements
6. Parallel development.
7. Supports RAD.
8. Supports multiple view technologies.

**Draw backs of MVC**

1. Increased complexity to implement controlling logic.
2. Implementing single point of entry (controller) is difficult

* So using MVC implemented framework is advisable. The following are the popular MVC implemented web application fra+meworks.
  + Struts (Apache Software Foundation)
  + Spring MVC (spring source)
  + JSF(Java Server Faces) (Sun Microsystems)
  + ADF(Application Development Framework) (oracle)
  + Webwork
  + Tapestry
  + Wicket
  + Flex …etc.

**Q) What is the purpose of spring MVC?**

- Spring MVC used to develop the web applications that uses MVC design pattern.

- Spring MVC is meant for making web application development faster, cost-effective, and flexible.

**Q) What are the java technologies used in a spring application?**

* Servlet
* JSP
* Java Bean

**Advantages of Spring MVC Framework**

* Clear separation of responsibilities. Because it implements MVC design pattern.
* Loose coupling among Model, View and Controller.
* In-built front controller
* Validation implementations is simplified
* Exception logic is simplified
* User input is available in the object oriented format.
* Forward logic is simplified.
* Consistent view support with tiles concept.
* Internationalization (i18n) logic is simplified.
* Spring MVC provides a set of custom JSP tags, which are useful to implement presentation logic.
* Also spring can integrate effortlessly with other popular Web Frameworks like [Struts](http://www.javabeat.net/articles/struts/1/), Webwork, [JSF](http://www.javabeat.net/articles/jsf/1/), and Tapestry …etc.
* Integration with other View technologies like Velocity, Free marker, Excel and PDF …etc.

**Q) What is Framework?**

* A Framework is a reusable semi finished application that can be customized to develop a specific application.

**Types of frameworks**

1. Web frameworks
2. Application frameworks

* Web framework will provide an environment to design and execute only web applications.
  + Ex: Struts, JSF, Webwork, wicket …etc.
* Application framework will provide an environment to design and execute distributed applications.
  + Ex: Spring, JBoss seam

**Differences between struts and spring**

1. Struts is web frame work there by supports only web applications, where as spring is web application framework there by supports both web applications and distributed applications.
2. Struts follows only MVC design pattern. Whereas spring implements so many design patterns like singleton, MVC, factory, service locator …etc.
3. Struts will allow only JSP as view but spring MVC allow all types view technologies like JSP, velocity, free marker, PDF, excel …etc.
4. Struts only support preprocessing, but spring supports both preprocessing and post processing.
5. In struts we cannot customize common logic to particular controller. But in spring it supports to customize common logic to particular one controller.
6. Struts will not provide integration built-in support with other technologies, where as spring provides built-in supports integration support with all technologies.
7. Struts is heavy weight framework. Whereas spring is light weight framework.
8. Struts is having better tag library support than spring.
9. In struts user data is available in Object Oriented format which is frame work specific object. Where as in spring it provides user data in POJO.

**Note:**

1. Struts 2.x allow all view technologies.
2. Supports preprocessing and post processing.
3. Provides integration built-in support with other technologies.
4. It is a light weight framework.
5. Struts 1.x has pre-defined support for html, bean, logic, nested, tiles tags …etc. But in struts 2.x version all the tags are combined as single classification named as ‘struts-tags’.