

U-4 JSP & web services

JSP

- stands for Java server Pages.
- Used to build dynamic web pages.
- Built ~~on~~ on top of servlet.
- In JSP we can embed Java code within HTML and XML
- It is key component of Java Enterprise edition.

Adv / Features

- 1) Useful for server side programming
- 2) Can be used with servlets to build business logic for any application.
- 3) Dynamic contents can be handled.
- 4) we can use our own custom tag libraries.
- 5) JSP can be used to build simple as well as complex web applications.
- 6) It follows MVC architecture to develop an web application.



Study material provided by: Vishwajeet Londhe

Join Community by clicking below links



Telegram Channel



https://t.me/SPPU_TE_BE_COMP

(for all engineering Resources)

SPPU Engineering & Technical
UPDATES
WhatsApp channel



WhatsApp Channel

(for all Engg & tech updates)



[https://whatsapp.com/channel/
0029ValjFriICVfpcV9HFc3b](https://whatsapp.com/channel/0029ValjFriICVfpcV9HFc3b)



Insta Page

(for all Engg & tech updates)

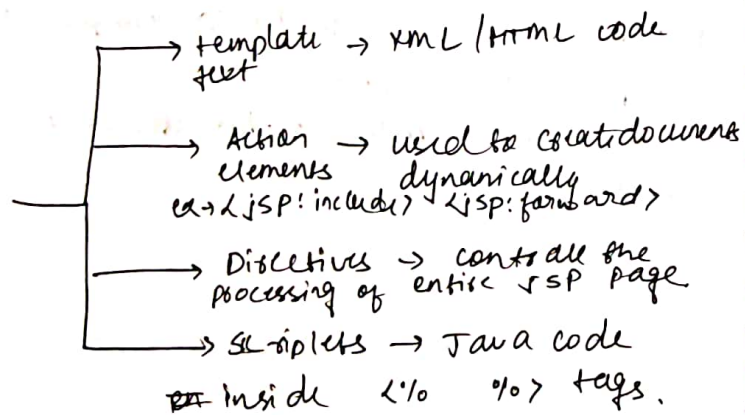


@SPPU_ENGINEERING_UPDATE

[https://www.instagram.com/
sppu_engineering_update](https://www.instagram.com/sppu_engineering_update)

JSP	SERVLET
1) .jsp extension	1) .java extension
2) View component of MVC	2) Controller component in MVC
3) To create dynamic web application	3) To handle request & process data
4) Generates dynamic content	4) Generates dynamic content on compilation
5) Slower than Servlet	5) Faster than JSP
6) Can define custom tag libraries	6) Cannot form custom tag lib.
7) Lower reusability	7) High reusability
8) Easier for Prog.	8) Too Difficult as it req. Java concepts

JSP
Structure



JSP Life cycle

Life cycle refers to series of steps that JSP goes through from its creation to destruction.

1) JSP compilation :-

→ when JSP is accessed for the first time, the web container compiles it into a servlet.
→ this step happens only once
JSP code → Servlets

2) JSP initialization :- & [init()]

→ In this stage 'init()' method is called which is used for initializing resources, such as database connections etc.
→ This is called only once ~~in~~ during life cycle.

3) JSP Execution / Service ⇒ [service()]

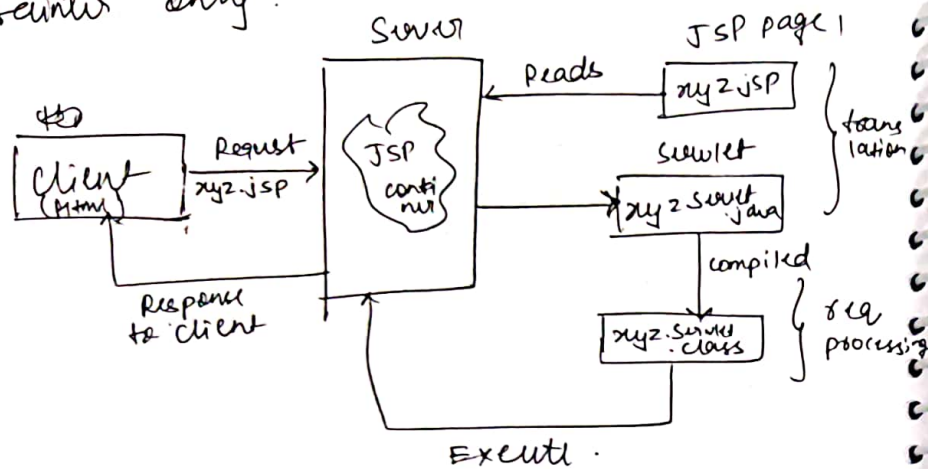
→ After initializing the new JSP page is ready to handle incoming requests.

Hence "service()" method is invoked which process the request, executes the JSP code, produces dynamic content and produces HTML response which is sent to the client.

4) JSP Destruction [destroy() :- [destroy()]]

~~the~~ destroy() method is called when there is no need of JSP page. This method is used for releasing the resources held by JSP page only called once.

JSP pages can be processed using JSP container only.



JSP application can be

run on \rightarrow JDK
 \rightarrow Tomcat web server

JSP DIRECTIVES

\Rightarrow They control the processing of entire JSP page
 $\langle \% @ \text{directive name} [\text{attribute name} = \text{"value"}] \% \rangle$

1) page directive

\Rightarrow This provides the information about the page. Its various attributes are
 i) import, extends, language, session.

$\hookrightarrow \langle \% @ \text{page import} = \text{"java.io.*"} \% \rangle$

2] Include directive \pm
 \rightarrow This is used to copy the contents of one JSP page to another
 $\langle \% @ \text{include file} = \text{"calculator.jsp"} \% \rangle$

3] Tag lib directive \pm
 \rightarrow This allows the user to use custom tags in JSP.
 $\langle \% @ \text{taglib uri} = \text{"taglib"} \text{ prefix} = \text{"sample tag"} \% \rangle$

We can define certain actions in JSP like include action.

$\langle \text{jsp:include} \rangle$

with this we can include different files in JSP page

include action

include directive

- 1) $\langle \text{jsp:include} \rangle$
- 2) includes diff files in JSP
- 3) Determined at runtime
- 4) Can include JSP & non JSP resources
- 5) Content is merged to output of both

- 1) $\langle \% @ \text{include} \% \rangle$
- 2) Copy content of another JSP page
- 3) Determined at translation
- 4) Can include only JSP resources.
- 5) Content is not merged to the output.

Implicit Objects

- These are predefined variables used to access request and application data
- used by scripting elements

Some of them are:

1) request :

- It represents the Client's HTTP request and provides access to req-specific info.
- retrieve user input
- ex → `get contentLength()`

2) response

- Represents the HTTP response that will be sent back to client
- Provides methods for manipulating the response
- ex → `addCookie()`
`addHeader()`

3) session

- Represents user session and store session info
- Can store / retrieve user session attributes
- `session.setAttribute("attr name", attr value);`

4) out

- provides method related to Input / Output
- To write HTML / text to response
- `out.println("Hello")`

5) application

- Represents server context
- allows access of application resources.
- `log()`
`getServerInfo()`

Java Beans Classes and JSP

- Java Beans are reusable components.
- Beans are used as instance of class in JSP Page
- we must specify scope of beans in JSP Page
- Bean is basically Java class with `getXX` and `setXX` methods.
- Java Bean class provides two methods for accessing the object
 - 1) `getProperty Name()` → called accessor, It returns the property value
 - 2) `setProperty Name()` → This is called mutator. Used to assign values to properties.

JSP action tags are special tags used in JSP to perform dynamic actions or execute Java code within the page

→ an example is '`<jsp:useBean>`' tag is used to access Java Objects, (JavaBeans) within JSP page. without writing explicit Java code

⇒ It eliminates the need for manually ~~inserting~~ representing Java objects in JSP Pages

MVC arch.

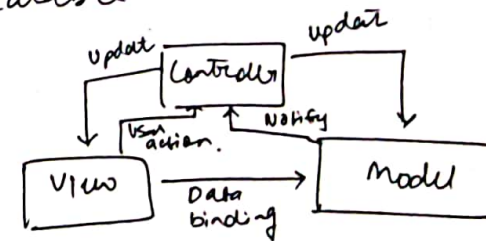
- ~~The~~ It is design model of JSP
 - MVC ⇒ Model View Controller
 - Basic idea in MVC design model is to separate out design logic into three parts → m, v, c
 - Any server app. can be divided into three categories i.e. model, View, Controller
- ↓
business
logic

↑
presen-
tation

↑
request
processing

- 1) ~~business~~ Model : It is concerned with the business logic applied on the application data i.e. code logic. It represents application data.
- 2) View : It deals with presentation, how the application will look. (It is responsible for displaying the data)
eg. background color, font.
- 3) Controller : It is also called request processing it just combines business logic and presentation i.e. (model & view) It is done in order to get response. It receives user input and translates it into actions to be performed on model or view.

- separate business logic (parsing & ~~req. processing~~)
- independent development
- Reusable model
- scalable



Web Services

- 1) They are software system that are deployed by web browser using web protocol.
- 2) They are software systems that allow diff appli... to com... and exchange data over internet using protocols.
- 3) They provide platform independent and language neutral approach.

ex → Currency converter
Credit card validation system.



Components & Architecture

- 2) SOAP → simple object access protocol (XML based)
 - ↳ allows exchange of messaging over HTTP.
 - ↳ used for exchange of information.
 - ↳ define rules and structures for

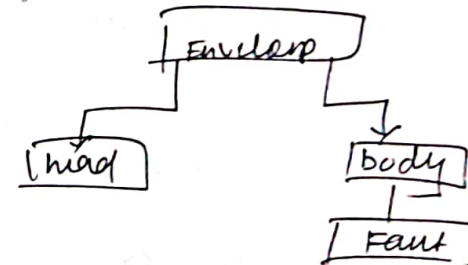
3) WSDL → web service description language. used for abstract description of web services. (describes the web service)
Client ~~must~~ reads this description and understand what input data is needed to perform web service.

4) UDDI is Universal ~~discovery~~ Description and Discovery Integration
↳ It contains directory of web services which are registered and from there service client can pick any ~~web~~ services.

4) XML → for data description or simple XML file.

WSDL [UDDI
WSDL
SOAP
XML]

Structure of SOAP

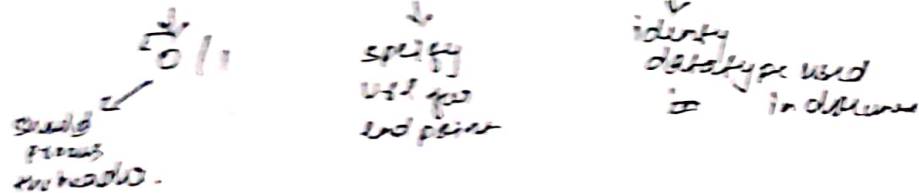


SOAP

- TEST of SOAP API
- identify XML as a SOAP msg

header

- It contains header information
- It contains other attributes must understand, action & encoding style



SOAP	REST
1) Complex XML messages and protocol	1) Uses lightweight simple protocols.
2) encoded XML so hard to read	2) Can be encoded in JSON, XML or even plain text
3) Less flexible rigid & highly structured	3) More flexible simple structure
4) Stateful communication	4) Stateless communication
5) Used for complex and strict controls	5) Used for simpler application like mobile devices,

STRUTS

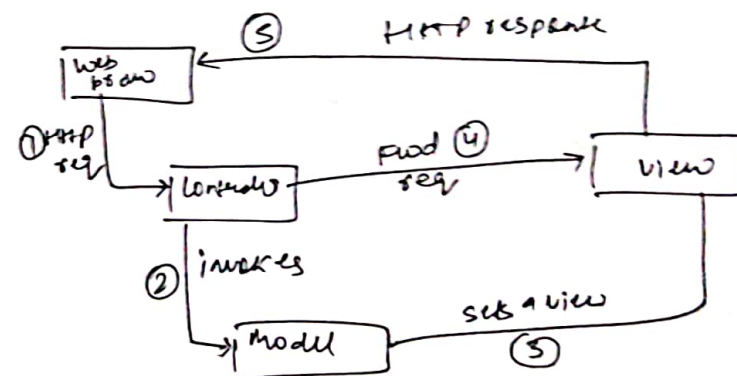
- framework used for developing Java web apps.
- latest ver 2.5
- ~~is~~ MVC arch. based on it
- Struts 2 provides support of MVC

features

- complex app - can be built easily using MVC
- The designs made are simple

Architecture

Based on MVC.



CONFIGURATION

→ two main files are used that are struts.xml and web.xml.

struts.xml

- primary configuration file in Struts 2
- responsible for defining overall structure of application & configuring components.
- specifies mapping between URL and ~~actions~~ ^{controllers} actions, interceptors etc.

web.xml

→ std config. file used in ^{Java} EE applications

In Struts 2 data tags are used to manipulate and display data on web pages.

- 1) <s:property>
- 2) <s:set>
- 3) <s:iterator>
- 4) <s:if> & <s:else>
- 5) <s:url> → creates url for action.

INTERCEPTOR

→ It is a feature that allows you to intercept and modify the behavior of request processing flow.

→ They are used to perform operations like authentication, validation & exception handling

→ ~~It gets called~~
It is invoked before & after action execution.

Some Interceptors (predefined) in Struts 2

- 1) Check box → handles checkbox input
- 2) Cookie → adds cookie to current action
- 3) Create session → creates Http session object
- 4) Clear session → unbinds Http session obj.
- 5) Validation → provides validation support.

Create custom Interceptors

- we need to implement interceptor
- which contains a method intercept()
- which calls invoke() method

