

(473) Pg

U-4 (JSP & Web Services)

Page No.

Date

→ JSP

- Java Server Pages
- alternative way than servlet
 - build web pages.
 - built on top of servlet.

Advantages -

- server side programming
- along with servlet → business logic for any application-
- handle dynamic contents.
- custom tag libraries
- specification & not a product
- variety of applications.
- component of J2EE.

Document Structure -

- 1) Template Text → XML of HTML code → used in JSP docs
markup → template text
- 2) Action elements - create documents dynamically.
- 3) Directives - controlling processing of entire JSP.
- 4) Scriptlets → Java code enclosed with `<%>` and `<%>` tags

Running JSP app

- Java Development Kit.
- Tomcat Web Server.

4K - 1-100.
5M - 1-138
6K - 1-72.

Page No. _____
Date _____

→ **Directives** → control processing entire JSP pages -
gives direction to server regarding processing of a page.

Page Directive
↓
info about page

- i) import
- ii) language
- iii) Content Type
- iv) extends
- v) session
- vi) Buffer
- vii) autoFlush
- viii) Is thread safe
- ix) Info
- x) errorPage
- xi) IsErrorPage

Include directive

→ copy content of one JSP page.
→ include code of
one file in another.

Jaglib directive

↓
user to use
custom tags in
JSP.

↓
custom tags by
user.

• Scripting Elements

- 1) Expression
↓
expression in JSP
- 2) Scriptlet
↓
<% and %>
- 3) Declaration
↓
declare variable or method.

→ Lifecycle of JSP

1) Translation -

- determined by semantics of JSP.
- semantics → directives, action & custom action.
- JSP page → read, parsed & validated.
- No errors → servlet class created

2) Compilation -

- file created → translation phase → class file.
- java code → validating & syntax errors.

3) Loading & Instantiating.

- servlet classed → loaded into memory & after loading JSP container.
- create instance of servlet class.

4) Jspinit() → same in servlet.

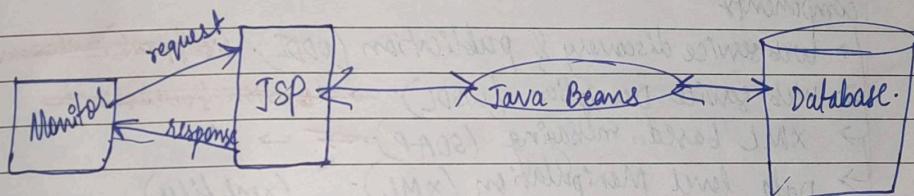
Jpservice

Jspdestroy()

JavaBeans

- reusable components
- declare properties, setter & server-side-
- set of conventions.

- characteristics
- no argument constructor
 - constructor public
 - implement serializable Interface
 - getter & setter methods
 - reuse software components
 - properties & methods → exposed to another application.



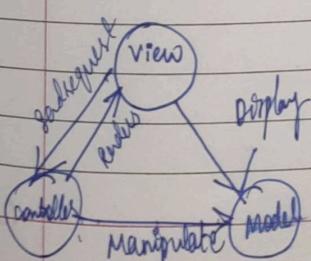
- properties →
 - `getProperty Name()` → read to property → accessor
 - `setProperty Name()` → write property → mutator
 - named attribute → accessed using object

→ MVC architecture (Model - View - Controller)

* Model → • shape of data • maintains data • receives & stores

* View → • User Interface • View display data • Modify Data

* Controller → • handle user request • VRML request • response → model data



- Adv →
 - easy maintain
 - easy to extend
 - easy to test
 - Navigation control is centralized

→ Web Services

- ↳ software systems → that displayed → web browser → ~~web page~~
- ↳ some software appln → end-users directly.
- ↳ independent → specific hardware or software.

examples

- ↳ credit card validation system → id pass & web service.
- ↳ weather forecast system → location & web service.
- ↳ currency converter → destination currency.

components

- ↳ Web service discovery & publication (UDDI). (~~refers to~~) ~~web services~~
- ↳ Web service Description (WSDL) → ~~location & web services~~
- ↳ XML based messaging (SOAP) →
- ↳ Data level Manipulation (XML) → (xml files)

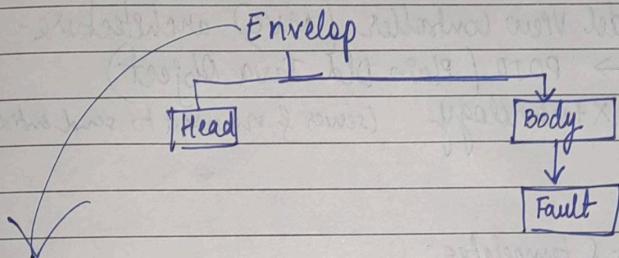
Features

- ↳ XML Based
- ↳ loosely coupled
- ↳ coarse grained
- ↳ ability to be synchronous or Asynchronous
- ↳ support remote procedure calls (RPC)
- ↳ supports document exchange.

→ SOAP

(Simple Object Access Protocol).

- based on XML
- web services → exchange info
- W3C recommendation
- exchange info → HTTP
- client-server communication → RPC.
-



1) Envelope → root of SOAP docs. → Namespace & encodingStyle.

2) Header → Header Info → mustUnderstand, actor & encodingStyle.
specifies Dest. specifies URL datatype

3) Body → SOAP message → ultimate endpoint of message.

4) Fault → optional element of SOAP message.

- represent error code
- faultcode → code for fault
- faultstring → gives details about fault.

→ Struts -

- framework software used → Java Web Applications
- struts framework → Craig McClanahan & Apache Software Foundation
- latest → struts 2
- open source product

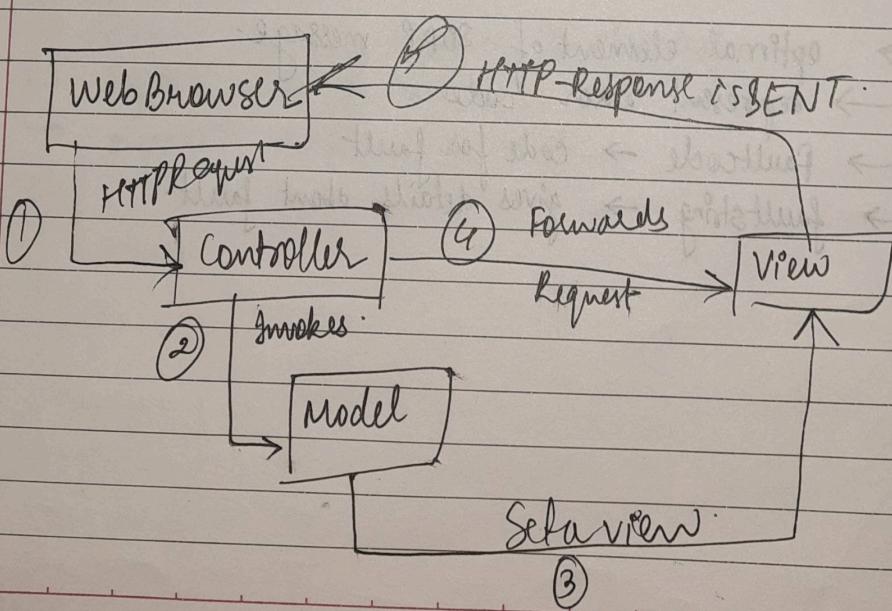
Features -

- based on Model View Controller (MVC) architecture
- action class → POJO (Plain Old Java Object)
- Struts 2 → AJAX technology (server & no need to send entire page)
- tag support
- integration
- support for views & templates
- multiple views

Adv - complex web appln → Model, View & Controller

- design using struts are simpler, consistent & easy to maintain

Architecture → MVC A



→ Interceptors -

- action dynamically intercepts.
- executed before & after action execution.

Advantages

- services of preprocessing & postprocessing
- validation, exceptional handling & logging
- coupled with request & response cycle.

