Unit 4 - Topics

- Introduction to JSP
- Client side scripting versus Server Side Scripting
- JSP Vs JavaScript
- Advantages and disadvantages of JSP
- Client and Server Responsibilities

Introduction to JSP

- JSP – Java Server Pages
- It is a server side programming
- It helps in creating dynamic web application that are platform independent

Client Side Scripting

- Executed by web browsers
- Used for validation and interactivity
- Used for dynamic web page alteration

Server Side Scripting

- Executed in web server
- Used to create dynamic pages
- Used to access database in server

Client Side Scripting	Server Side Scripting
Source code visible	Source code not visible
Execution based on browser	Not dependent on browser
Runs on users computer	Runs on web server
No Data security	Data security
Javascript	JSP, PHP

JSP	JavaScript
Base on Servlet	Lightweight scripting language
Dynamic features on server side	Dynamic features at client side
Web server is needed	Browser with JavaScript Engine is needed
Maintained by Java Specification Group	Maintained by ECMA TC-39
HTML can be embedded	HTML cannot be embedded

Advantages of JSP

- Easy to use and learn
- Support JAVA API's which can be easily integrated with HTML code
- Can be opened on any browser
- Changes can be made at logic page rather than individual page

Disadvantage of JSP

- Database access in not easy
- Debugging JSP is hard
- Compilation time is high for the first time

Client Responsibilities

- Handle user interaction
- Initiates a request to the server and waits for the reply
- When the server sends the reply it process it and display it to the user

Server Responsibilities

- Listen for client request
- Process request from client by sending it to web application
- Execute the web application
- Send the result to the client

Unit 3 & Unit 4

Section 3.4

Topics

- Live Connect

Section 4.1

Topics

- Introduction to JSP
- Client side scripting versus Server Side Scripting
- JSP Vs JavaScript
- Advantages and disadvantages of JSP
- Client and Server Responsibilities

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Unit 4 - Topics

- Introduction to JSP
- Client side scripting versus Server Side Scripting
- JSP Vs JavaScript
- Advantages and disadvantages of JSP
- Client and Server Responsibilities

Introduction to JSP

- JSP – Java Server Pages
- It is a server side programming
- It helps in creating dynamic web application that are platform independent

Client Side Scripting

- Executed by web browsers
- Used for validation and interactivity
- Used for dynamic web page alteration

Server Side Scripting

- Executed in web server
- Used to create dynamic pages
- Used to access database in server

Client Side Scripting	Server Side Scripting
Source code visible	Source code not visible
Execution based on browser	Not dependent on browser
Runs on users computer	Runs on web server
No Data security	Data security
Javascript	JSP, PHP

JSP	JavaScript
Base on Servlet	Lightweight scripting language
Dynamic features on server side	Dynamic features at client side
Web server is needed	Browser with JavaScript Engine is needed
Maintained by Java Specification Group	Maintained by ECMA TC-39
HTML can be embedded	HTML cannot be embedded

Advantages of JSP

- Easy to use and learn
- Support JAVA API's which can be easily integrated with HTML code
- Can be opened on any browser
- Changes can be made at logic page rather than individual page

Disadvantage of JSP

- Database access in not easy
- Debugging JSP is hard
- Compilation time is high for the first time

Client Responsibilities

- Handle user interaction
- Initiates a request to the server and waits for the reply
- When the server sends the reply it process it and display it to the user

Server Responsibilities

- Listen for client request
- Process request from client by sending it to web application
- Execute the web application
- Send the result to the client

Unit 3 & Unit 4

Section 3.4

Topics

- Live Connect

Section 4.1

Topics

- Introduction to JSP
- Client side scripting versus Server Side Scripting
- JSP Vs JavaScript
- Advantages and disadvantages of JSP
- Client and Server Responsibilities

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

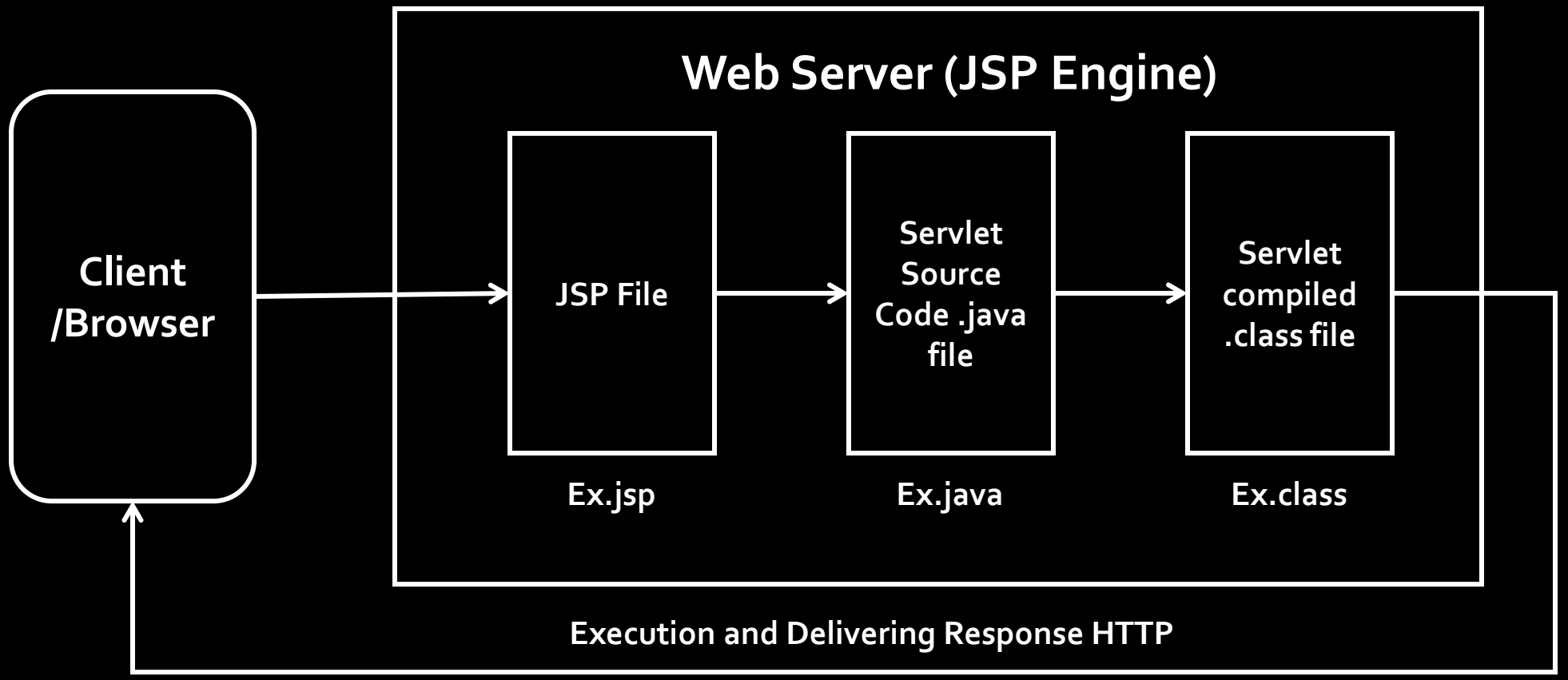
Topics

- JSP Architecture
- Life Cycle of JSP page
- JSP vs Servlets
- JSP vs ASP.NET
- List of JSP Servers

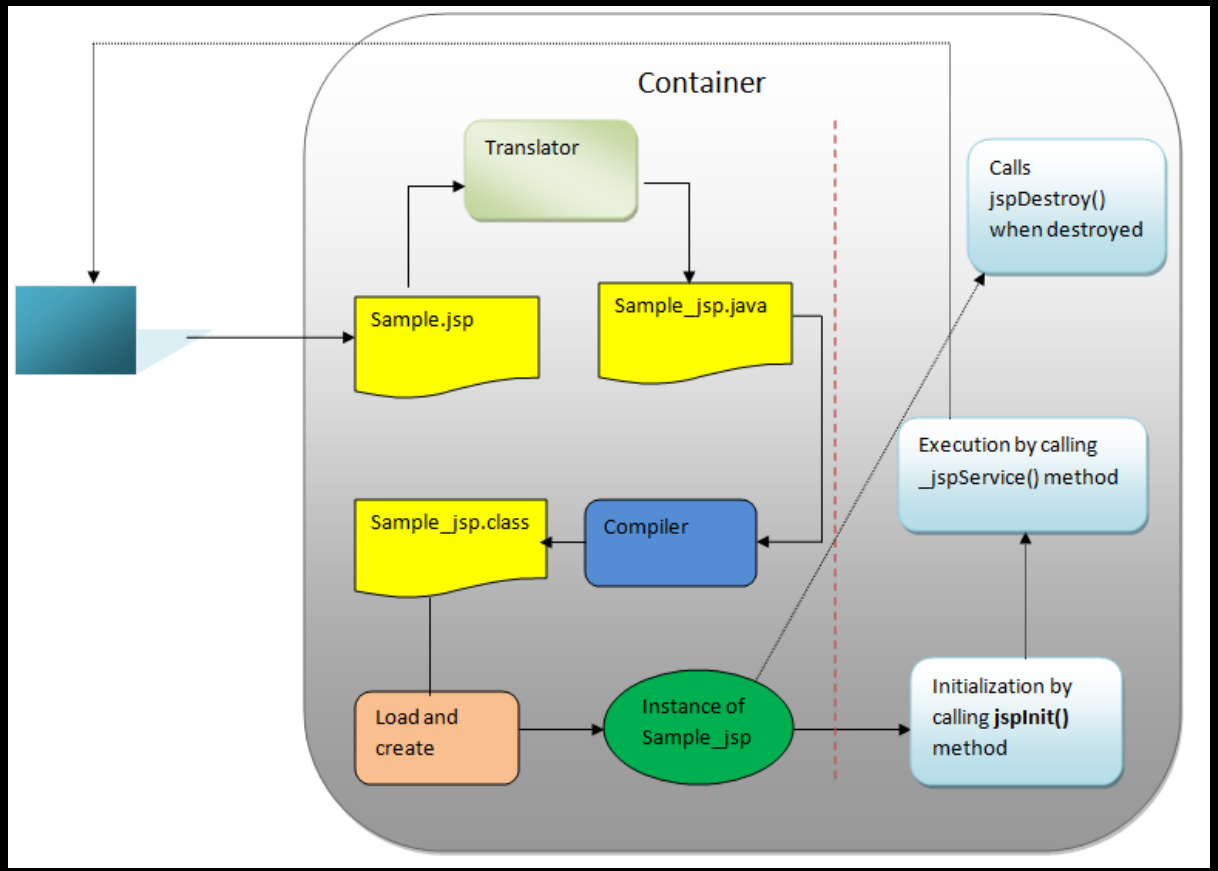
JSP Architecture

- Important component of JSP architecture is the JSP engine.
- JSP engine is needed to process the JSP page

Architecture Diagram



Life Cycle of JSP page



Phases in JSP Page lifecycle

1. Page Translation
2. Page Compilation
3. Load Class and Create Instance
4. JSP initialization (Call jspinit())
5. JSP execution (Call _jspServcie())
6. JSP clean up (Call jspDestroy())

SERVLET	JSP
Servlet is a java code	JSP is a html based code
It is HTML in Java	It is Java in HTML
It is faster than JSP	JSP is slower than servlet
It accept all protocol request	Accept only http requests
Methods can be overridden	Methods cannot be overridden

JSP	ASP
Java Server Page	Active Server Page
JSP is free	ASP is paid
Code is compiled at runtime	Code is interpreted
Executes faster	Execution time is less
Provide better security	Provides poor security

List of JSP Servers

- **Apache Tomcat**
  - Open Source Software
- **Glassfish**
  - Open Source software by Sun Microsystems
- **Jboss**
  - Open Source Software

Unit 4

Section 4.1

Topics

- JSP Architecture
- Life Cycle of JSP page
- JSP vs Servlets
- JSP vs ASP.NET
- List of JSP Servers

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- JSP Elements
  - Comments
  - Directives
  - Scripting Elements

JSP Elements

- JSP elements are written inside <% %> tags
- Code inside the tag are processed by JSP engine
- Other text in the JSP page is considered as HTML code

Example 1

Comments

- A JSP page can have two types of comments
  - HTML Comments

    <!-- Comment Text --!>

  - JSP Comments

    <%-- JSP Comments --%>

Example 2

Directives Elements

- Used to provide special information to the JSP engine
- Directive elements are handled at the translation phase

  Syntax

  <%@ *directive attribute* %>

Types of Directives

- Three Types of Directive
  - Page
  - Include
  - Taglib

Page Directive

- Used to define page dependent attributes
- Eg, session tracking, error page, import, etc

<%@ page ..... %>

Example 3

Include Directive

- Used to include one file to another file
- Included file can be HTML, JSP, text files, etc.

<%@ include file... %>

Example 4

Taglib Directive

- Used to declare tag library, which can contain custom tags.
- Tag library is a set of user-defined tags

```
<%@ taglib uri="uri" prefix="tagprefix" %>
```

Example 5

Scripting Elements

- JSP scripting elements are written inside <% %> tags
- Three types of scripting elements
  - Declarations
  - Scriplets
  - Expressions

Declaration tag

- Used to declare variables and methods
- Variable declared here can be accessed by all the methods

<%! *declaration* %>

Scriplets Tag

- Used to write Java Code inside JSP Page.

<% java code %>

Example 6

Expressions Tag

- Used to compute small expressions
- It is always single line
- The result is included in the HTML page

<%= *expression* %>

Example 7

Sample page

- Create a JSP page to change color of the page based on time

Example 8

Unit 4

Section 4.2

Topics

- JSP Elements
  - Comments
  - Directives
  - Scripting Elements

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Implicit Objects

Implicit Objects

- Created by JSP Engine during translation phase
- These objects are java objects
- They can be directly used in scriplets

List of Implicit Objects

- Request
- Response
- Pagecontext
- Application
- Out
- Config
- Page
- Session
- Exception

Request Object

- It is used to get the data on a JSP page which has been entered by the user
- It is an instance of HttpServletRequest Class
- It is used to request the information like parameter, header information, server name, etc..

Important Method

- request.getParameter() – used to get the value of the request parameter

Syntax

request.getParameter(“Name”)

Example 9

Additional Methods

- getRequestURL() – return full URL of the client
- getCharacterEncoding() – return the character set in which the page is encoded
- getServerName() – return the server name
- getServerPort() – return the port number

- getRemoteHost() – return the name of the computer which request the server
- getQueryString() – return the query string
- getMethod() – return method used for request
- getRemoteUser() – return the name of the user

Response Object

- It is used to modify the response sent to the browser
- It is an instance of HttpServletResponse Class
  - void setContentType(String type)
  - void sendRedirect(String Address)
  - void addCookie(Cookie Value)

Pagecontext Object

- It is an implicit object of type PageContext class.
- It is used to set, get or remove attributes from
  - Page
  - Request
  - Session
  - Application

Application Object

- It is an instance of ServletContext class.
- It is used to get the context information and attributes in JSP.
  - Object getAttribute(String attName)
  - void setAttribute(String attName, Object obj)

Out Object

- It is an instance of JspWriter class
- Used to write content to the client.
  - void print()
  - void println()
  - void newLine()
  - void clear()
  - void flush()

Config Object

- It is an instance of ServletConfig Class.
- It is used for getting configuration information for a particular JSP page.
  - String getInitParameter(String pram)
  - Enum getInitParameterNames()
  - String getServletName()

Page Object

- It refers to the current servlet instance
- this keyword is used to refer the current JSP page.
- It is rarely used

Session Object

- It is an instance of HttpSession class
- It is most frequently used implicit object.
- It is used for storing user's data to make it available on other JSP pages
  - void setAttribute(String attName, Object Obj)
  - Object getAttribute(String attName)

Exception Object

- It is an instance of java.lang.Throwable class
- It is used for exception handling in JSP.
- The exception object can be only used in error pages.

Object Scope

- JSP provides 4 scopes
  - Page Scope
  - Request Scope
  - Session Scope
  - Application Scope

- **Page Scope**
  - Objects can be accessed only from within the same page
  - Out, exception, response, config, page and pageContext
- **Request Scope:**
  - Objects can be accessed only for the current request. Single request may include multiple page
  - request

- **Session Scope**
  - Objects can be accessed from pages that belong to the same session
  - session
- **Application Scope:**
  - Objects can be accessed from any pages across the application
  - Application

Unit 4

Section 4.3

Topics

Implicit Object