Servlet

NILESH BHANDARE

Agenda

2

- Basic Introduction
- Tomcat Server Installation & Configuration
- Servlet Creation by Implementing Servlet interface,
- Servlet Creation by extending GenericServlet class,
- Servlet Creation by extending HttpServlet class
- Many More ….

Prerequisites

3

- Static Website & Dynamic Website
- Client Server Architecture
- Web Server
  - supports http protocol only.
  - contain enough security to prevent unauthorized users.
  - enough services to develop effective server side program.
  - Tomcat server, web logic server, etc.
- Application Server
  - Any protocol can be supported.
  - Provides 100% security to the server side program.
  - provides effective services to develop server side program.
  - web logic server, web sphere server, pramathi server, etc.

Background

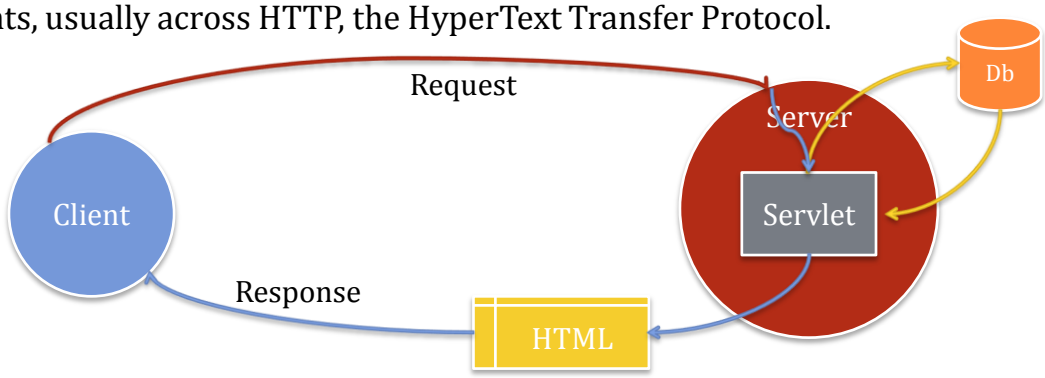
4

- In the initial days of server side programming there is a concept called CGI and this was implemented in the languages called C and PERL. Because of this approach CGI has the following disadvantages.
  - Platform dependency.
  - Not enough security is provided.
  - Having lack of performance. Since, for each and every request a new and separate process is creating (for example, if we make hundreds of requests, in the server side hundreds of new and separate processes will be created)
  - To avoid the above problems SUN micro system has released a technology called **Servlets**

Servlet

5

- A **servlet** is **a small Java program** that runs within **a web server or application server.** Servlets **receive and respond to requests** from Web clients, usually across HTTP, the HyperText Transfer Protocol.



Installation & Configuration

6

- Tomcat Server
  - https://youtu.be/5TEmvi5kTM4
- Configuration with IDE
  - https://youtu.be/aNFpjKGKb64
- If you are not able to get Server option in Preference
  - https://www.youtube.com/watch?v=PYdvP5-Tcs4

Ways to create servlet

7

- The servlet example can be created by three ways:
  - By implementing Servlet interface,
  - By inheriting GenericServlet class, (or)
  - By inheriting HttpServlet class

1. Create Servlet using Javax.servlet Servlet Interface

8

```
Class myservlet implements
Servlet
{

Override all methods;

}
```

User Defined Class

Inherited all properties **Implements Interface**

Servlet interface

1. **public void init (ServletConfig config)-** initializes the servlet.
2. **void service (ServletRequest request,ServletResponse response)-**provides response for the incoming request.
3. **public void destroy()**is invoked only once and indicates that servlet is being destroyed.
4. **public ServletConfig getServletConfig()** returns the object of ServletConfig.
5. **public String getServletInfo()**returns information about servlet such as writer, copyright, version etc.

1-3 Life Cycle Methods & 4-5 Non Life Cycle

Mapping
Need to tell server there is a servlet and how to execute.
Web.xml
Deployment Descriptor

Web.xml

9

```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
           http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
           version="3.1">

      <servlet>
          <servlet-name>comingsoon</servlet-name>
          <servlet-class>mysite.server.ComingSoonServlet</servlet-class>
      </servlet>

      <servlet-mapping>
          <servlet-name>comingsoon</servlet-name>
          <url-pattern>/*</url-pattern>
      </servlet-mapping>

</web-app>
```


Demonstration – Simple Servlet

10

File→New→Dynamic Web Project

Create Package under java /main→Class File→implement Servlet Interface→Override All 5 Methods→Save.

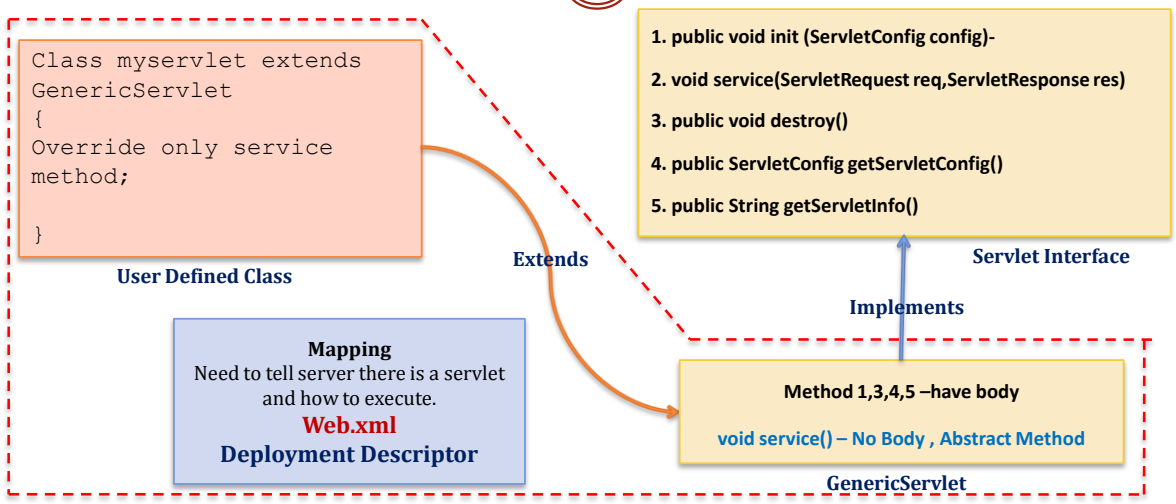
Create Web.xml under WEBINF folder->Add two Main Tags and Two subtag in each main tag

Write index.html ->under the folder webapp→ run project

Add html output in class file.

2. Create Servlet using GenericServlet

11



Demonstration – GenericServlet

12

File→New→Dynamic Web Project

Create Package under java /main→Class File→extends GenericServlet Class→ Override only service method → Save.

Create Web.xml under WEBINF folder->Add two Main Tags(servlet & Servlet Mapping) and Two subtag in each main tag

Write index.html ->under the folder webapp→ run project

Add html output in class file.

3 Create Servlet extends HttpServlet class

13

```
Class myservlet extends
HttpServlet
{
override at least one
method of http;
}
```

User Defined Class

extends

Mapping
Need to tell server there is a servlet and how to execute.
Web.xml
Deployment Descriptor

Servlet Interface

Implements

GenericServlet

Method 1,3,4,5 –have body, void service() – No Body , Abstract Method

inherited

A subclass of HttpServlet must **override at least one method** usually one of these

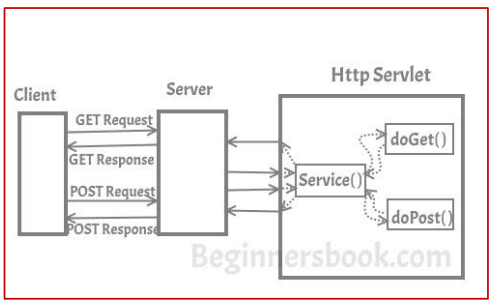
`doGet, doPost, doPut, doDelete, etc`

Http Servlet

HTTP servlet

14

- The HttpServlet class extends the GenericServlet class and implements Serializable interface
- HTTP Servlet doesn't override the service() method. Instead it overrides the doGet() method or doPost() method or both

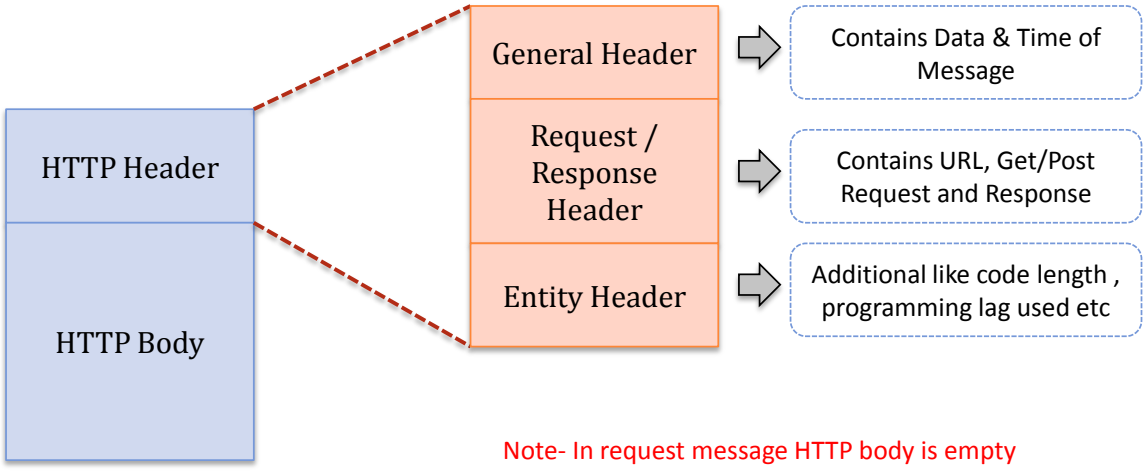


Recall – Client Server and messages

15

Request and Response Message Format

16



Note- In request message HTTP body is empty

Get Request Method ⟷ Post Request Method

17

Get	Post
URL	URL + Hidden Body (Data + Message)
Unsafe	Safe
MaxLength-2048	Unlimited

Methods of Http Servlet class

18

- **doGet**, if the servlet supports HTTP GET requests
- **doPost**, for HTTP POST requests
- **doPut**, for HTTP PUT requests
- **doDelete**, for HTTP DELETE requests
- **init** and **destroy**, to manage resources that are held for the life of the servlet
- **getServletInfo**, which the servlet uses to provide information about itself
- **doHead**, It receives an HTTP HEAD request from the protected service method and handles the request
- **doOptions**, called by the server to handle a OPTIONS request.
- **doTrace**, called by the server to handle a TRACE request.
- **getLastModified**, returns the time the HttpServletRequest object was last modified, in milliseconds.

JSP

19

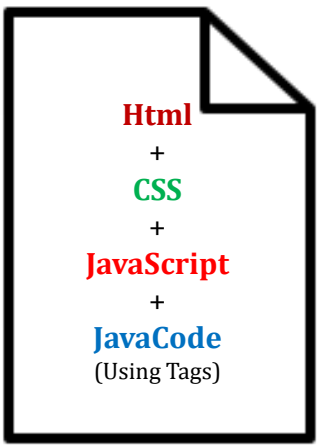
JAVA SERVER PAGES

JSP

20

- **JSP** technology is used to create web application just like Servlet technology.
- an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.
- JSP is first converted into servlet by JSP container before processing the client's request.

#web_application #DynamicWebContent
 # JAVA_into_HTML



index.jsp

Background

21

Why JSP ?

22

1. Static content are generated by java from inside servlet. Designing in servlet is difficult.
2. For every request in servlet you have to write service method which is very tiresome process.
3. Whenever modification made in static content static content then servlet need to recompiled and redeployed

#NoRedeployment #NoRe-Compilation #easy #ReduceCode

Important Tags

23

Declaration Tag

It is used to declare variables and Methods

```
<%!        %>
```

```
<%!
int var=10;
%>
```

Scriplets

It is used to Java Code

```
<%         %>
```

```
<%
java code
%>
```

Expression Tag

It evaluates and convert the expression to a string

```
<%=        %>
```

```
<% =
n1 = n1+n2
%>
```


Demonstration

24

Directives Tags –page

25

The page directive is used to provide instructions to the container.
To apply instruction on entire page.
Allow to add anywhere but by convention, page directives are coded at the top of the JSP page

Syntax

```
<%@
page attribute="value"
%>
```

attributes

Buffer, autoFlush, contentType, errorPage, isErrorPage, extends, import, info, isThreadSafe, language, session, isELIgnored, isScriptingEnabled

Directives Tags –include

26

The **include** directive is used to include a file during the translation phase.
Ask container to merge the content of other external files with current JSP during the translation
You may code the ***include*** directives anywhere in your JSP page.

Syntax

```
<%@ include        %>
```

Example

```
<%@ include file = "header.jsp" %>
```

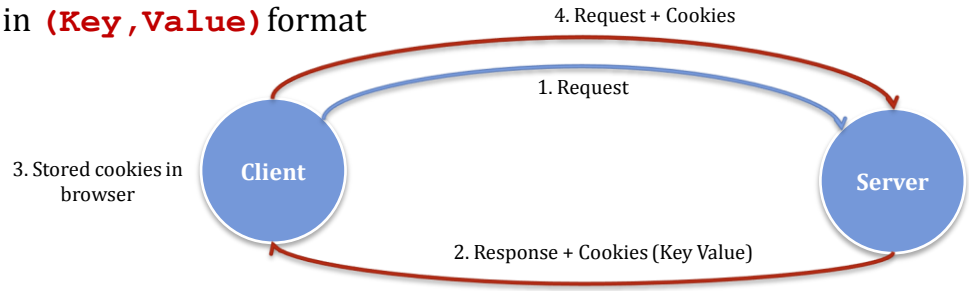
Cookies Handling

28

Cookies Handling

29

- Cookies are text files stored on the client computer and they are kept for various information tracking purposes.
- JSP transparently supports HTTP cookies using underlying servlet technology.
- Stored in **`(Key,Value)`** format



Types of Cookies

30

- There are basically two types of cookies:

  1. Persistent cookies: These cookies are also known as permanent cookies. They remain on the hard drive and persist until the user deletes them or they expire themselves.

  2. Session cookies: These cookies are also known as temporary cookies. They get deleted themselves as soon as the session ends or the browser closes.

Setting Cookies

31

- Cookie constructor with a cookie name and a cookie value, both of which are strings.
  - ***Cookie cookie = new Cookie("key","value");***
- Use **setMaxAge** to specify how long (in seconds) the cookie should be valid.
  - ***cookie.setMaxAge(60*60*24);***
- Use **response.addCookie** to add cookies in the HTTP response header as follows
  - ***response.addCookie(cookie);***

Read Cookies

32

- To read cookies, you need to create an array of *javax.servlet.http.Cookie* objects by calling the **getCookies( )** method of *HttpServletRequest*.
- Then cycle through the array, and use **getName()**and **getValue()** methods to access each cookie and associated value.

Delete Cookies

33

- To delete cookies is very simple. If you want to delete a cookie, then you simply need to follow these three steps –
  1. Read an already existing cookie and store it in Cookie object.
  2. Set cookie age as zero using the setMaxAge() method to delete an existing cookie.
  3. Add this cookie back into the response header.

Session Tracking

34

Need of Session Tracking

35

- HTTP is a "stateless" protocol
- **Stateless Problem** - each time a client retrieves a Webpage, the client opens a separate connection to the Web server and the server automatically does not keep any record of previous client request.

Methods to Session Tracking

36

Cookies

- A webserver can assign a unique session ID as a cookie to each web client and for subsequent requests from the client they can be recognized using the received cookie.

Hidden Form Fields

- web server can send a hidden HTML form field along with a unique session ID as follows
- `<input type = "hidden" name = "sessionid" value = "12345">`

Methods to Session Tracking

37

URL Rewriting

- You can append some extra data at the end of each URL.
- **http://tutorialspoint.com/file.htm;sessionid=12345**, the session identifier is attached as **sessionid = 12345**
- URL rewriting is a better way to maintain sessions and works for the browsers when they don't support cookies.

Session Object

- JSP makes use of the servlet provided HttpSession Interface.
- Dirrent methods available through the session object

Demonstration

38

JSTL

39

JSP STANDARD TAG LIBRARY

JSTL

40

- The JSP Standard Tag Library (JSTL) represents a set of tags to simplify the JSP development.

- Why JSTL ?
  - **Fast Development** JSTL provides many tags that simplify the JSP.
  - **Code Reusability** We can use the JSTL tags on various pages.
  - **No need to use scriptlet tag** It avoids the use of scriptlet tag.

JSTL Tags

Tag Name	Description
Core	The JSTL core tag provide variable support, URL management, flow control, etc. The URL for the core tag is http://java.sun.com/jsp/jstl/core. The prefix of core tag is c.
Function	The functions tags provide support for string manipulation and string length. The URL for the functions tags is http://java.sun.com/jsp/jstl/functions and prefix is fn.
Formatting	The Formatting tags provide support for message formatting, number and date formatting, etc. The URL for the Formatting tags is http://java.sun.com/jsp/jstl/fmt and prefix is fmt.
XML	The XML tags provide flow control, transformation, etc. The URL for the XML tags is http://java.sun.com/jsp/jstl/xml and prefix is x.
SQL	The JSTL SQL tags provide SQL support. The URL for the SQL tags is http://java.sun.com/jsp/jstl/sql and prefix is sql.

Setting up JSTL with Eclipse

42

- Step 1- Download two jar files
  - jstl-standard.jar
    - http://www.java2s.com/Code/Jar/j/Downloadjstlstandardjar.htm
  - jstl-1.2.jar
    - http://www.java2s.com/Code/Jar/j/Downloadjstl12jar.htm
- Step2-simply copy the JAR files in the distribution‘s 'lib' directory to your application's webapps\ROOT\WEB-INF\lib directory.

Demonstration of CORE Tags

43

List of JSTL Tags with Syntax

44

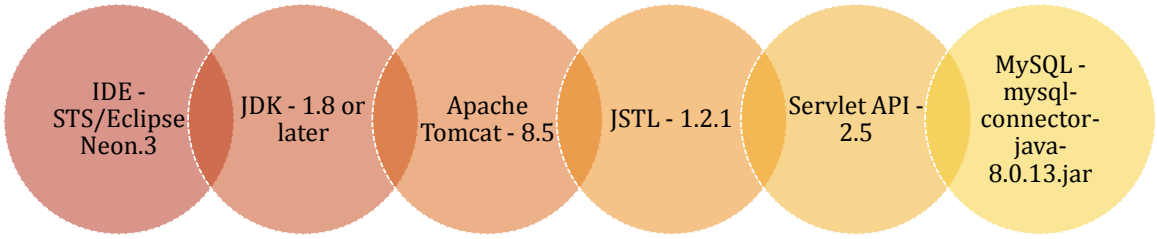
- Refer
  - https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm

JSP Servlet JDBC MySQL CRUD

45

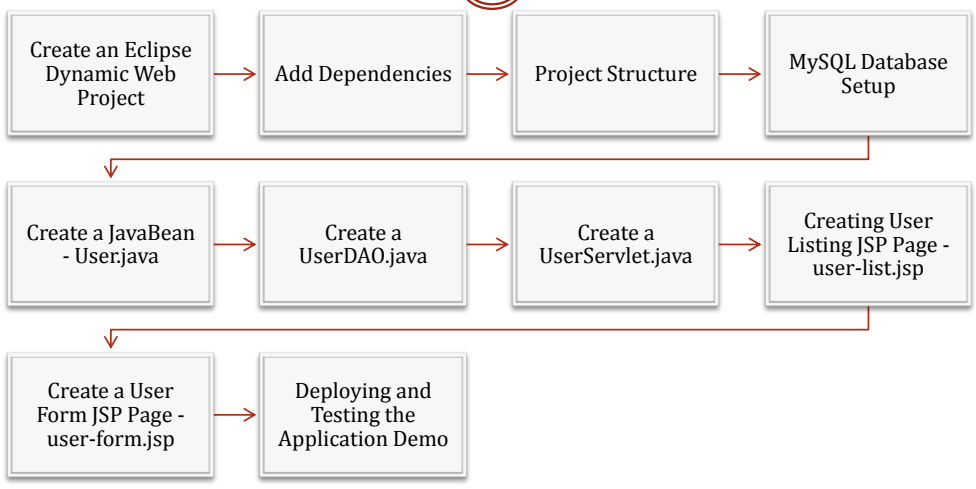
Tools and Technologies

46



Development Steps

47



More details of Project

48

- Click Here….
- Video-Click Here
  - https://youtu.be/RqiuxA_OFOk

Post Survey – Servlet, JSP, JSTL and MYSQL

49

- https://forms.gle/hZV9pC7ajqWTWsxi8

THANK YOU

50

NILESH BHANDARE
NDBHANDARE@MITAOE.AC.IN
9096142430