Module 5
Servlets

Background

- J2EE
  - To create web centric applications
  - Simplified creations
  - Versatile
  - Java based
- HTML & XML

Servlets

- small programs that execute on the server side of a web connection.

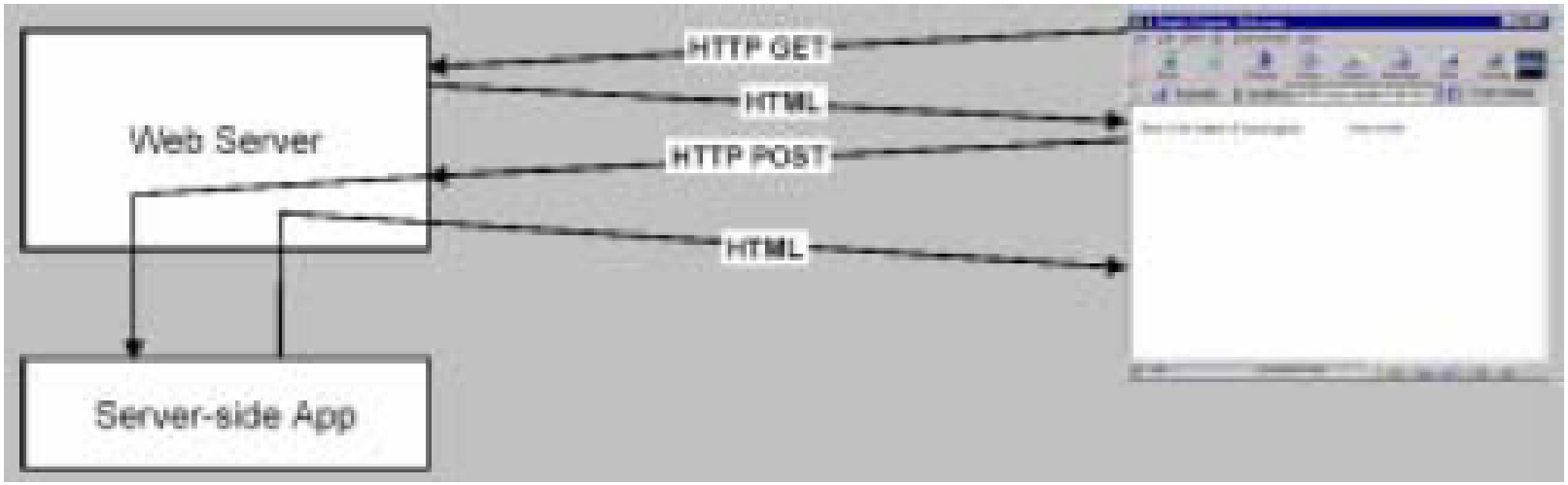
- dynamically extend the functionality of a web server

Background

- Static Pages
- Dynamic Page
  - CGI

- Disadvantage of CGI
- expensive in terms of **processor and memory resources** to create a separate process for each client request.
- It was expensive to **open and close database connections** for each client request
- **Not platform- independent**

- Architecture

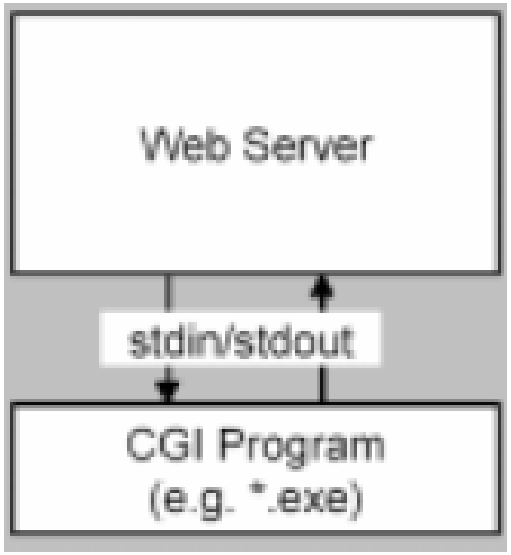


- Applications
  - Dynamic generates HTML pages
  - Access to database and/or back-end servers
  - etc.

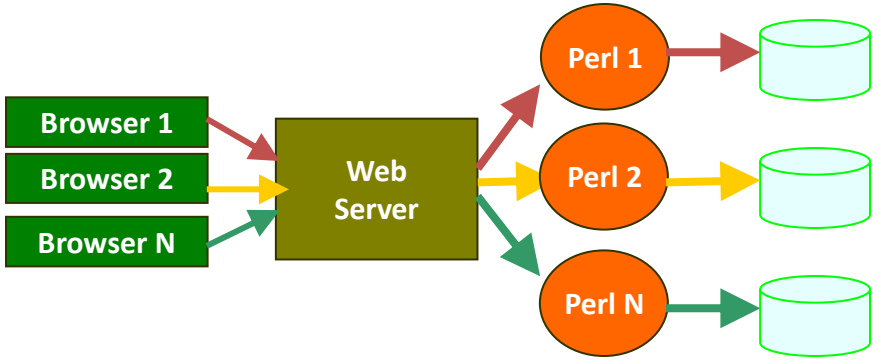
- Common Gateway Interface (CGI)
  - Basically call external program
  - Use standard input and output for data exchange
  - Programming language independent

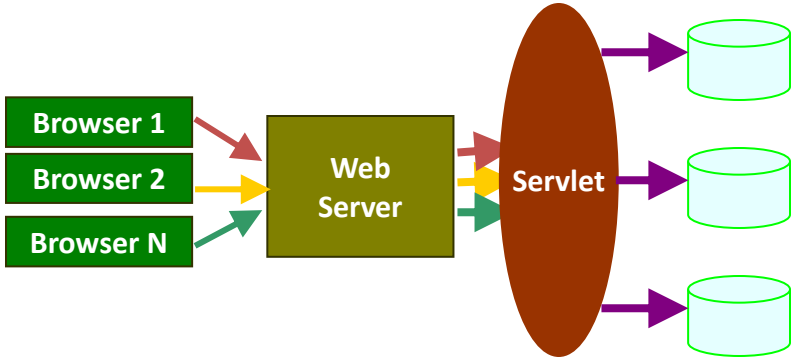
Weakness

- CGI program may not be easily portable to other platform
- Substantial overhead is incurred in starting the CGI process

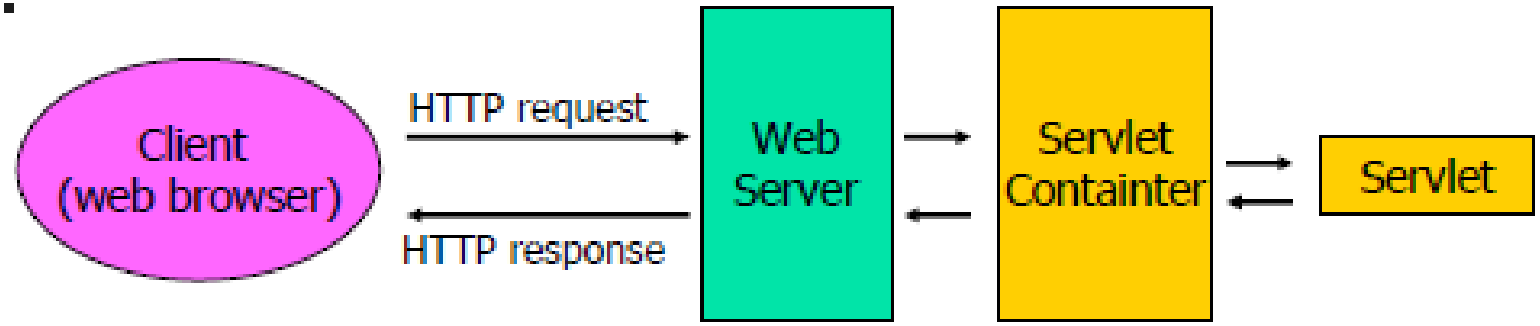


Servlets vs. CGI





- A Servlet does not run in a separate process.
- A Servlet stays in memory between requests.
- A CGI program needs to be loaded and started for each CGI request.
- There is only a single instance of a servlet which answers all requests concurrently.



- The client makes a request via HTTP
- The web server receives the requests and forwards it to the servlet
  - If the servlet has not yet been loaded, the web server loads it into the JVM and executes it
- The servlet receives the HTTP request and performs some type of process
- The servlet returns a response to the web server
- The web server forwards the response to the client

Advantages of Servlets

- Performance better: **execute within the address space of a web server.** It is not necessary to create a separate process to handle each client request.
- are **platform-independent** because they are written in Java.
- the Java security manager on the server enforces a set of restrictions to **protect the resources on a server machine.**
- the full functionality of the **Java class libraries** is available. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms.

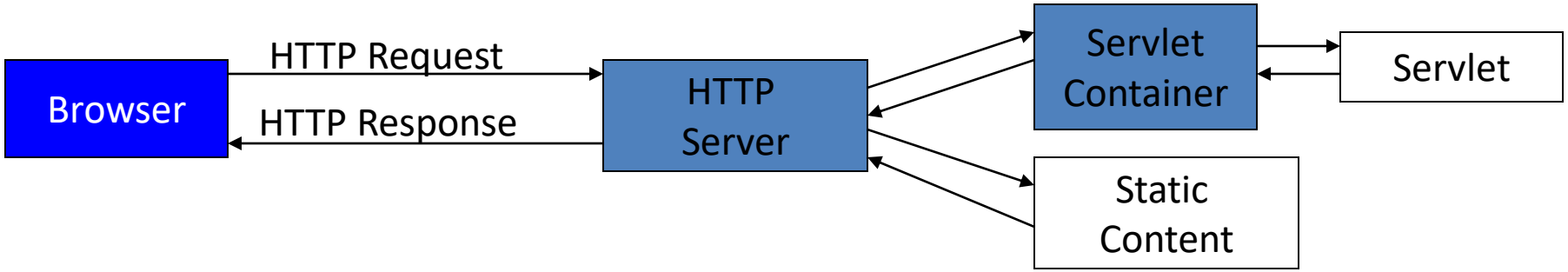
What can you build with Servlets?

- Search Engines
- E-Commerce Applications
- Shopping Carts
- Product Catalogs
- Intranet Applications
- Groupware Applications:
  - bulletin boards
  - file sharing

What is required?

- Servlets are not run in the same sense as applets and applications.
- Since they extend a web server, they require a servlet container to function.
- A servlet container is a part of a web server or application server that provides the network services over which request and response are sent.
- Contains and manages the servlets through their lifecycle.
- The container provides other services as well.

Servlet Container Architecture



Lifecycle of Servlets

- Three methods are central to the life cycle of a servlet.
  - **init( )**
  - **service( )**
  - **destroy( )**

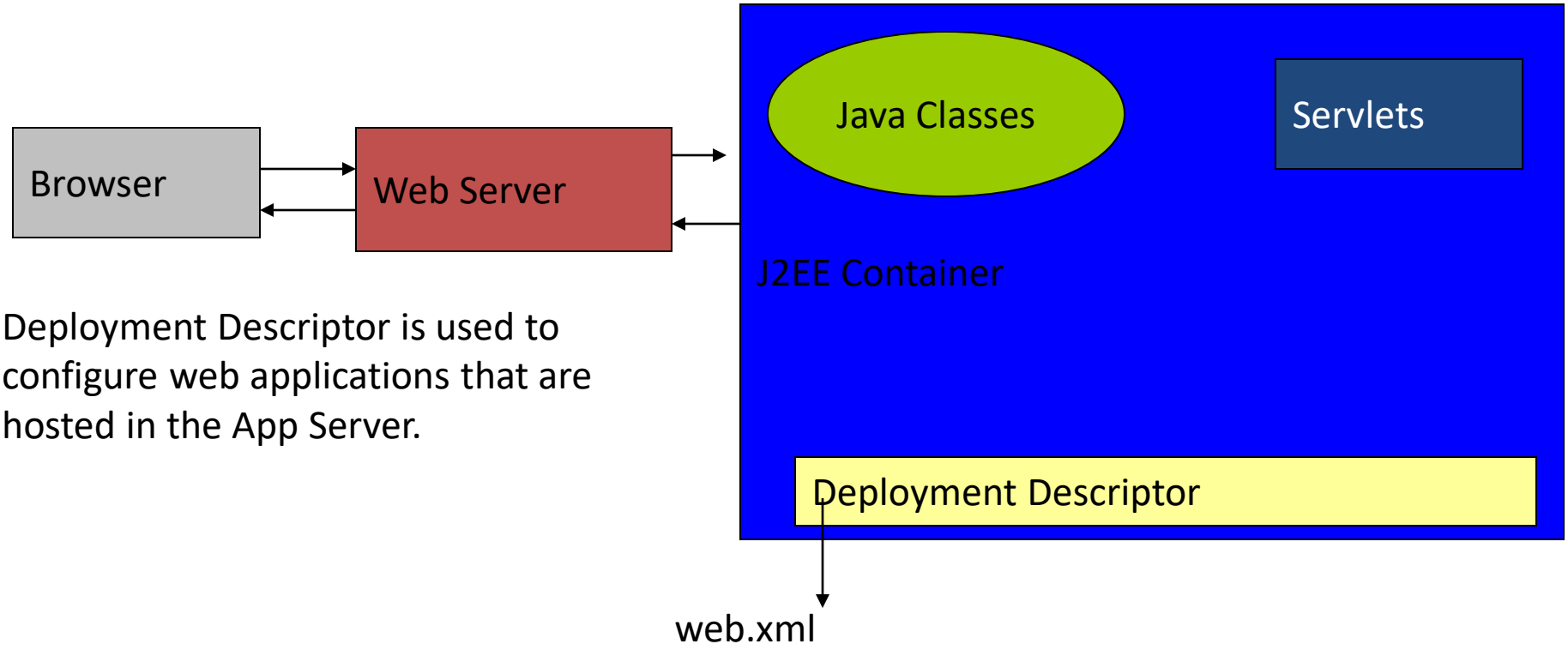
- ***implemented by every servlet***

- ***invoked at specific times by the server***

- First, a user enters a Uniform Resource Locator (URL) to a web browser. The browser then generates an HTTP request for this URL. This request is then sent to the appropriate server.
- Second, this HTTP request is received by the web server. The server maps this request to a particular servlet. The servlet is dynamically retrieved and loaded into the address space of the server.
- Third, the server invokes the **init( ) method of the servlet. This method is invoked only** when the servlet is first loaded into memory. It is possible to pass initialization parameters to the servlet so it may configure itself.

- Fourth, the server invokes the **service( ) method of the servlet. This method is called to** process the HTTP request. It may also formulate an HTTP response for the client.
- The servlet remains in the server's address space and is available to process any other HTTP requests received from clients. The **service( ) method is called for each HTTP request.**
- Finally, the server may decide to unload the servlet from its memory. The server calls the **destroy( )** method to relinquish any resources such as file handles that are allocated for the servlet.
- Important data may be saved to a persistent store. The memory allocated for the servlet and its objects can then be garbage collected.

Web Container/Servlet engine



Deployment Descriptor is used to configure web applications that are hosted in the App Server.

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

348.0/590.0MB

- New Project... Ctrl+Shift+N
- New File... Ctrl+N
- Open Project... Ctrl+Shift+O
- Open Recent Project >
- Close Project (isewebappl)
- Close Other Projects
- Close All Projects
- Open File...
- Open Recent File >
- Project Groups...
- Project Properties (isewebappl)
- Import Project >
- Export Project >
- Save Ctrl+S
- Save As...
- Save All Ctrl+Shift+S
- Page Setup...
- Print... Ctrl+Alt+Shift+P
- Print to HTML...
- Exit

service(ServletRequest re

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

375.4/590.0MB

Projects × Files

- box1
- callbyref
- charfile
- classwing
- commandline
- copyfile
- Counter
- datatypes
- demo1
- filechar
- hellobnmit
- HelloWorld
- isewebappl
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

New Project ×

Steps

1. **Choose Project**
2. ...

Choose Project

Filter:

Categories:

- Java with Maven
- Java with Gradle
- Java with Ant
  - JavaFX
  - Java Web
  - Java Enterprise
  - NetBeans Modules
- HTML5/JavaScript
- C/C++
- PHP
- Samples

Projects:

- Web Application
- Web Application with Existing Sources
- webfreeform.xml

Description:

Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.

< Back | Next > | Finish | Cancel | Help

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

436:2/590.0MB

Projects × Files

- box1
- callbyref
- charfile
- classwing
- commandline
- copyfile
- Counter
- datatypes
- demo1
- filechar
- hellobnmit
- HelloWorld
- isewebappl
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

New Project ×

Steps

1. **Choose Project**
2. ...

Choose Project

Filter:

Categories:

- Java with Maven
- Java with Gradle
- Java with Ant
  - JavaFX
  - Java Web
  - Java Enterprise
  - NetBeans Modules
- HTML5/JavaScript
- C/C++
- PHP
- Samples

Projects:

- Web Application
- Web Application with Existing Sources
- webfreeform.xml

Description:

Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.

< Back | Next > | Finish | Cancel | Help

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

505.1/590.0MB

Projects × Files

- box1
- callbyref
- charfile
- classwing
- commandline
- copyfile
- Counter
- datatypes
- demo1
- filechar
- hellobnmit
- HelloWorld
- isewebappl
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: WebApplication4

Project Location: E:\Saritha\OOC-3 sem 2022 Browse...

Project Folder: E:\Saritha\OOC-3 sem 2022\WebApplication4

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

< Back | Next > | Finish | Cancel | Help

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

605.6/654.0MB

Projects × Files

- box1
- callbyref
- charfile
- classwing
- commandline
- copyfile
- Counter
- datatypes
- demo1
- filechar
- hellobnmit
- HelloWorld
- isewebappl
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

New Web Application ×

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server: GlassFish Server (1) Add...

Java EE Version: Java EE 7 Web

Context Path: /WebApplication4

< Back | Next > | Finish | Cancel | Help

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

736.7/766.0MB

Projects × Files

- box1
- callbyref
- charfile
- classwing
- commandline
- copyfile
- Counter
- datatypes
- demo1
- filechar
- hellobnmit
- HelloWorld
- isewebappl
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

New Web Application

Steps

1. Choose Project
2. Name and Location
3. Server and Settings
4. **Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

- [ ] Spring Web MVC
- [ ] JavaServer Faces
- [ ] Struts 1.3.10

< Back | Next > | Finish | Cancel | Help

isewebappl - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

582.0/766.0MB

Projects × Files

- WEB-INF
  - index.html
- Source Packages
  - <default package
    - servtrial2.java
- Test Packages
- Libraries
- Test Libraries
- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
  - Libraries
  - Configuration Files
- webapptrial1
- wrap
- wrapexample

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

index.html ×

Source History

```
<!DOCTYPE html>
<!--
Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
Click nbfs://nbhost/SystemFileSystem/Templates/JSP Servlet/Html.html to edit this template
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div>TODO write content</div>
    </body>
</html>
```

WebApplication4 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

434.9/898.0MB

Projects × Files

- WEB-INF
  - index.html
- Source Packages
  - <default package
    - servtrial2.java
- Test Packages
- Libraries
- Test Libraries
- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
  - Libraries
  - Configuration Files
- webapptrial1
- wrap
- wrapexample

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

index.html ×

Source History

```
1  <!DOCTYPE html>
2  <!--
3  Click nbfs://nbhos
4  Click nbfs://nbhos
5  -->
6  <html>
7      <head>
8          <title>TODO
9          <meta char
10         <meta name
11     </head>
12     <body>
13         <div>TODO
14     </body>
15 </html>
16
```

New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project: WebApplication4

Filter:

Categories:

- Web
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Dependency I
- Java

File Types:

- JSP
- JSF Page
- Servlet
- Filter
- Web Application Listener
- WebSocket Endpoint
- HTML
- XHTML
- Cascading Style Sheet
- JavaScript File

Description:

Creates a new servlet class. A servlet is a server-side Java class which runs within a web server.

< Back | Next > | Finish | Cancel | Help

WebApplication4 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

608:6/898:0MB

Projects × Files

- WEB-INF
  - index.html
- Source Packages
  - <default package
    - servtrial2.java
- Test Packages
- Libraries
- Test Libraries
- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
  - Libraries
  - Configuration Files
- webapptrial1
- wrap
- wrapexample

Navigator ×

Members <em...

- servtrial3 :: GenericServlet
  - servtrial3()
  - service(ServletRequest re

index.html ×

Source History

```
1  <!DOCTYPE html>
2  <!--
3  Click nbfs://nbhos
4  Click nbfs://nbhos
5  -->
6  <html>
7      <head>
8          <title>TODO
9          <meta char
10         <meta name=
11     </head>
12     <body>
13         <div>TODO
14     </body>
15 </html>
16
```

New Servlet

Steps

1. Choose File Type
2. **Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name: servlettest1

Project: WebApplication4

Location: Source Packages

Package:

Created File: E:\Saritha\OOC-3 sem 2022\WebApplication4\src\java\servlettest1.java

⚠ Warning: It is highly recommended that you do not place Java classes in the default packag

< Back | Next > | Finish | Cancel | Help

WebApplication4 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

261.1/898.0MB

Projects × Files

- WEB-INF
  - index.html
- Source Packages
  - <default package>
    - servtrial2.java
- Test Packages
- Libraries
- Test Libraries
- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
    - <default package>
      - servlettest1.java
  - Test Packages
  - Libraries
  - Test Libraries

Navigator ×

Members <em...

- servlettest1 :: HttpServlet
  - servlettest1()
  - doGet(HttpServletReque
  - doPost(HttpServletRequ
  - getServletInfo() : String
  - processRequest(HttpServ

index.html × servlettest1.java ×

Source History

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
4   */
5
6  import java.io.IOException;
7  import java.io.PrintWriter;
8  import javax.servlet.ServletException;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 /**
14  *
15  * @author shreyas
16  */
17 public class servlettest1 extends HttpServlet {
18
19     /**
20      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
21      * methods.
22      *
23      * @param request servlet request
24      * @param response servlet response
25      * @throws ServletException if a servlet-specific error occurs
26      * @throws IOException if an I/O error occurs
27      */
28     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
29             throws ServletException, IOException {
30         response.setContentType("text/html;charset=UTF-8");
31         try ( PrintWriter out = response.getWriter()) {
32             /* TODO output your page here. You may use following sample code. */
33             out.println("<!DOCTYPE html>");
34             out.println("<html>");
35             out.println("<head>");
36             out.println("<title>Servlet servlettest1</title>");
```

webapptrial1 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

298.3/898.0MB

Projects × Files

- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
    - <default package>
      - servlettest1.java
  - Test Packages
  - Libraries
  - Test Libraries
  - Configuration Files
- webapptrial1
  - Web Pages
  - Source Packages
    - <defau
      - ser
  - Test Packa
  - Libraries

Navigator ×

Members

- servlettrial1
  - servlettr
  - service(S

Olmethods3

index.html × servlettest1.java × servlettrial1.java ×

Source History

```
4   */
5
6   import jakarta.servlet.GenericServlet;
7   import jakarta.servlet.ServletException;
8   import jakarta.servlet.ServletRequest;
9   import jakarta.servlet.ServletResponse;
    import jakarta.servlet.http.HttpServlet;
    import jakarta.servlet.http.HttpServletRequest;
    import jakarta.servlet.http.HttpServletResponse;
13  import java.io.IOException;
14  import java.io.PrintWriter;

    shreyas

    ss servlettest1 extends GenericServlet {

    d service (ServletRequest request, ServletResponse response) throws ServletException, IOException

    se.setContentType("text/html");
    riter pw=response.getWriter();
    ntln("<B>Welcome to 3 sem ISE students to the world of Servlets");
    se();
```

Open	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Compile File	F9
Run File	Shift+F6
Debug File	Ctrl+Shift+F5
Profile File	
Test File	Ctrl+F6
Debug Test File	Ctrl+Shift+F6
Profile Test File	
Add	
Delete	Delete
Save As Template...	
Find Usages	Alt+F7
Refactor	>
BeanInfo Editor...	
File Members	Ctrl+F12
File Hierarchy	Alt+F12
History	>

WebApplication4 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

362.3/898.0MB

Projects | Files

- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
    - <default package>
      - servlettest1.java
  - Test Packages
  - Libraries
  - Test Libraries
  - Configuration Files
- webapptrial1
  - Web Pages
  - Source Packages
    - <default package>
      - servlettrial1.java
  - Test Packages
  - Libraries

Navigator

Members <em...

- servlettest1 :: GenericServlet
  - servlettest1()
  - service(ServletRequest re

index.html | servlettest1.java | servlettrial1.java

Source | History

```
 */

import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

/**
 *
 * @author shreyas
 */
public class servlettest1 extends Gener

public void service (ServletRequest rec                    eption, IOException
{
    response.setContentType("text/html");
    PrintWriter pw=response.getWriter();
    pw.println("<B>Welcome to 3 sem ISE students to the world of Servlets");
    pw.close();
}
}
```

Set Servlet Execution URI

Select servlet execution URI, optionally add some request parameters :
e.g. **/flowerServlet?flower=rose&color=red**

/servlettest1

OK | Cancel

WebApplication4 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

542.2/898.0MB

Projects × Files

- Configuration Files
- WebApplication4
  - Web Pages
  - Source Packages
    - <default package>
      - servlettest1.java
  - Test Packages
  - Libraries
  - Test Libraries
  - Configuration Files
- webapptrial1
  - Web Pages
  - Source Packages
    - <default package>
      - servlettrial1.java
  - Test Packages
  - Libraries

Navigator ×

Members <em...

- servlettest1 :: GenericServlet
  - servlettest1()
  - service(ServletRequest re

index.html × servlettest1.java × servlettrial1.java ×

Source History

```
 */

import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

/**
 *
 * @author shreyas
 */
public class servlettest1 extends GenericServlet {

public void service (ServletRequest request, ServletResponse response) throws ServletException, IOException
{
    response.setContentType("text/html");
    PrintWriter pw=response.getWriter();
    pw.println("<B>Welcome to 3 sem ISE students to the world of Servlets");
    pw.close();
}
}
```

OImethods3.java Javadoc Sources Output

Type here to search

http://localhost:48278/WebApplication4/servlettest1

Search...

localhost | localhost

Welcome to 3 sem ISE students to the world of Servlets

```
import java.io.IOException;
import java.io.PrintWriter;


public class servlettest1 extends GenericServlet {

public void service (ServletRequest request, ServletResponse
response) throws ServletException, IOException
{
   response.setContentType("text/html");
   PrintWriter pw=response.getWriter();
   pw.println("<B>Welcome to 3 sem ISE students to the world of
Servlets");
   pw.close();
}
}
```

getWriter( ) method obtains a **PrintWriter**. Anything written to this stream is sent to the client as part of the HTTP response. Then **println( )** is used to write some simple HTML source code as the HTTP response.

establishes the MIME type of the HTTP response. In this program, the MIME type is text/html. This indicates that the browser should interpret the content as HTML source code

The Servlet API

- Two packages contain the classes and interfaces that are required to build servlets.
- These are
- **javax.servlet and javax.servlet.http.**
- **Not part of Java core packages**

The javax.servlet Package

Interface	Description
Servlet	Declares life cycle methods for a servlet.
ServletConfig	Allows servlets to get initialization parameters.
ServletContext	Enables servlets to log events and access information about their environment.
ServletRequest	Used to read data from a client request.
ServletResponse	Used to write data to a client response.

Class	Description
GenericServlet	Implements the **Servlet** and **ServletConfig** interfaces.
ServletInputStream	Provides an input stream for reading requests from a client.
ServletOutputStream	Provides an output stream for writing responses to a client.
ServletException	Indicates a servlet error occurred.
UnavailableException	Indicates a servlet is unavailable.

- **The Servlet Interface- init(), service(), destroy().**

Method	Description
void destroy( )	Called when the servlet is unloaded.
ServletConfig getServletConfig( )	Returns a **ServletConfig** object that contains any initialization parameters.
String getServletInfo( )	Returns a string describing the servlet.
void init(ServletConfig *sc*) throws ServletException	Called when the servlet is initialized. Initialization parameters for the servlet can be obtained from *sc*. An **UnavailableException** should be thrown if the servlet cannot be initialized.
void service(ServletRequest *req*, ServletResponse *res*) throws ServletException, IOException	Called to process a request from a client. The request from the client can be read from *req*. The response to the client can be written to *res*. An exception is generated if a servlet or IO problem occurs.

The ServletConfig Interface

The **ServletConfig interface allows a servlet to obtain configuration data when it is loaded.**

Method	Description
ServletContext getServletContext( )	Returns the context for this servlet.
String getInitParameter(String *param*)	Returns the value of the initialization parameter named *param*.
Enumeration getInitParameterNames( )	Returns an enumeration of all initialization parameter names.
String getServletName( )	Returns the name of the invoking servlet.

The ServletContext Interface

- The ServletContext interface enables servlets to obtain information about their environment.

Method	Description
Object getAttribute(String *attr*)	Returns the value of the server attribute named *attr*.
String getMimeType(String *file*)	Returns the MIME type of *file*.
String getRealPath(String *vpath*)	Returns the real path that corresponds to the virtual path *vpath*.
String getServerInfo( )	Returns information about the server.
void log(String *s*)	Writes *s* to the servlet log.
void log(String *s*, Throwable *e*)	Writes *s* and the stack trace for *e* to the servlet log.
void setAttribute(String *attr*, Object *val*)	Sets the attribute specified by *attr* to the value passed in *val*.

The ServletRequest Interface : The ServletRequest interface enables a servlet to obtain information about a client request.

Method	Description
Object getAttribute(String *attr*)	Returns the value of the attribute named *attr*.
String getCharacterEncoding( )	Returns the character encoding of the request.
int getContentLength( )	Returns the size of the request. The value –1 is returned if the size is unavailable.
String getContentType( )	Returns the type of the request. A **null** value is returned if the type cannot be determined.
ServletInputStream getInputStream( ) throws IOException	Returns a **ServletInputStream** that can be used to read binary data from the request. An **IllegalStateException** is thrown if **getReader( )** has already been invoked for this request.
String getParameter(String *pname*)	Returns the value of the parameter named *pname*.
Enumeration getParameterNames( )	Returns an enumeration of the parameter names for this request.
String[ ] getParameterValues(String *name*)	Returns an array containing values associated with the parameter specified by *name*.
String getProtocol( )	Returns a description of the protocol.
BufferedReader getReader( ) throws IOException	Returns a buffered reader that can be used to read text from the request. An **IllegalStateException** is thrown if **getInputStream( )** has already been invoked for this request.
String getRemoteAddr( )	Returns the string equivalent of the client IP address.
String getRemoteHost( )	Returns the string equivalent of the client host name.
String getScheme( )	Returns the transmission scheme of the URL used for the request (for example, "http", "ftp").
String getServerName( )	Returns the name of the server.
int getServerPort( )	Returns the port number.

<u>The ServletResponse Interface :</u> The ServletResponse interface enables a servlet to formulate a response for a client.

Method	Description
String getCharacterEncoding( )	Returns the character encoding for the response.
ServletOutputStream getOutputStream( ) throws IOException	Returns a **ServletOutputStream** that can be used to write binary data to the response. An **IllegalStateException** is thrown if **getWriter( )** has already been invoked for this request.
PrintWriter getWriter( ) throws IOException	Returns a **PrintWriter** that can be used to write character data to the response. An **IllegalStateException** is thrown if **getOutputStream( )** has already been invoked for this request.
void setContentLength(int *size*)	Sets the content length for the response to *size*.
void setContentType(String *type*)	Sets the content type for the response to *type*.

The GenericServlet Class

- The GenericServlet class provides implementations of the basic life cycle methods for a servlet.
- GenericServlet implements the Servlet and ServletConfig interfaces.

The ServletInputStream Class

- The **ServletInputStream class extends InputStream.**
- **implemented by the servlet container**
- provides an input stream that a servlet developer can use to read the data from a client request.
- It defines the default constructor.

The ServletOutputStream Class

- The **ServletOutputStream class extends OutputStream.**
- **It is implemented by the servlet** container and provides an output stream that a servlet developer can use to write data to a client response.
- A default constructor is defined.
- It also defines the **print( ) and println( )** methods, which output data to the stream.

The Servlet Exception Classes

- **javax.servlet defines two exceptions.**
  - **ServletException, which indicates that a servlet** problem has occurred.
  - **UnavailableException, which extends ServletException.** It indicates that a servlet is unavailable.

WebApplication6 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

675.7/810.0MB

Projects x | Files

- Libraries
- Test Libraries
- Configuration Files
- WebApplication6
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages
    - <default package>
      - newservlet.java
  - Test Packages
  - Libraries
  - Test Libraries
  - Configuration Files
- webapptrial1
  - Web Pages
    - WEB-INF
    - index.html

Navigator x

- html

newservlet.java x | index.html x | newservlet.java x | index.html x | currdate.java x

Source | History

```
1  <!DOCTYPE html>
2  <!--
3  Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
4  Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Html.html to edit this template
5  -->
6  <html>
7      <head>
8          <title>TODO supply a title</title>
9          <meta charset="UTF-8">
10         <meta name="viewport" content="width=device-width, initial-scale=1.0">
11     </head>
12     <body>
13         <form
14
15 action="newservlet" method="post">
16 <table>
17 <tr>
18 <td><B>Employee</td>
19 <td><input type="text" name= "empname"  size= "25" value=""></td>
20 </tr>
21 <tr>
22 <td><B>Phone</td>
23 <td><input type="text" name="phonenum" size="25" value=""></td>
24 </tr>
25 </table>
26 <input type="submit" value="Submit">
27     </body>
28 </html>
29
```

Javadoc Output

29:1 INS Unix (L

Submit

Reading Servlet Parameters

```
<html>
   <head>
      <title>TODO supply a title</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form

action="newservlet" method="post">
<table>
<tr>
<td><B>Employee</td>
<td><input type="text" name= "empname"  size= "25" value=""></td>
</tr>
<tr>
<td><B>Phone</td>
<td><input type="text" name="phonenum" size="25" value=""></td>
</tr>
</table>
<input type="submit" value="Submit">
   </body>
</html>
```

```
import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
```

```
public class newservlet extends GenericServlet {
public void service(ServletRequest request,ServletResponse response)throws
ServletException, IOException {

// Get print writer.
PrintWriter pw = response.getWriter();

// Get enumeration of parameter names.
Enumeration e = request.getParameterNames();

// Display parameter names and values.
while(e.hasMoreElements()) {
String pname = (String)e.nextElement();
pw.print(pname + " = ");
String pvalue = request.getParameter(pname);
pw.println(pvalue);
}

pw.close();

}        }
```

WebApplication6 - Apache NetBeans IDE 12.6

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Files

- Libraries
- Test Libraries
- Configuration Files
- WebApplication6
  - Web Pages
    - WEB-INF
    - index.html
  - Source Packages
    - <default package>
      - newservlet.java
  - Test Packages
  - Libraries
  - Test Libraries
  - Configuration Files
- webapptrial1
  - Web Pages
    - WEB-INF
    - index.html

newservlet.java - Navigator

Members <empty>

- newservlet :: GenericServlet
  - newservlet()
  - service(ServletRequest request, ServletResponse

newservlet.java | index.html | newservlet.java | index.html | currdate.java

Source History

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 */

import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
/*import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;*/

/**
 *
 * @author shreyas
 */
public class newservlet extends GenericServlet {
public void service(ServletRequest request,ServletResponse response)throws ServletException, IOException {
// Get print writer.
PrintWriter pw = response.getWriter();
// Get enumeration of parameter names.
Enumeration e = request.getParameterNames();
// Display parameter names and values.
while(e.hasMoreElements()) {
String pname = (String)e.nextElement();
pw.print(pname + " = ");
String pvalue = request.getParameter(pname);
pw.println(pvalue);


}
pw.close();



}        }
```

Javadoc Output

41:13

TODO supply a title

localhost:48278/WebApplication6/

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea... | Five Free

Employee

Phone

Submit

TODO supply a title

localhost:48278/WebApplication6/

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea...

Employee saritha

Phone 456789

Submit

localhost:48278/WebApplication6/newservlet

localhost:48278/WebApplication6/newservlet

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea...

```
empname = saritha
phonenum = 456789
```

The javax.servlet.http Package

Interface	Description
HttpServletRequest	Enables servlets to read data from an HTTP request.
HttpServletResponse	Enables servlets to write data to an HTTP response.
HttpSession	Allows session data to be read and written.
HttpSessionBindingListener	Informs an object that it is bound to or unbound from a session.

Class	Description
Cookie	Allows state information to be stored on a client machine.
HttpServlet	Provides methods to handle HTTP requests and responses.
HttpSessionEvent	Encapsulates a session-changed event.
HttpSessionBindingEvent	Indicates when a listener is bound to or unbound from a session value, or that a session attribute changed.

The **HttpServletRequest interface** enables a servlet to obtain information about a client request.

Method	Description
String getAuthType( )	Returns authentication scheme.
Cookie[ ] getCookies( )	Returns an array of the cookies in this request.
long getDateHeader(String *field*)	Returns the value of the date header field named *field*.
String getHeader(String *field*)	Returns the value of the header field named *field*.
Enumeration getHeaderNames( )	Returns an enumeration of the header names.
int getIntHeader(String *field*)	Returns the **int** equivalent of the header field named *field*.
String getMethod( )	Returns the HTTP method for this request.
String getPathInfo( )	Returns any path information that is located after the servlet path and before a query string of the URL.
String getPathTranslated( )	Returns any path information that is located after the servlet path and before a query string of the URL after translating it to a real path.
String getQueryString( )	Returns any query string in the URL.
String getRemoteUser( )	Returns the name of the user who issued this request.
String getRequestedSessionId( )	Returns the ID of the session.
String getRequestURI( )	Returns the URI.
StringBuffer getRequestURL( )	Returns the URL.
String getServletPath( )	Returns that part of the URL that identifies the servlet.
HttpSession getSession( )	Returns the session for this request. If a session does not exist, one is created and then returned.
HttpSession getSession(boolean *new*)	If *new* is **true** and no session exists, creates and returns a session for this request. Otherwise, returns the existing session for this request.
boolean isRequestedSessionIdFromCookie( )	Returns **true** if a cookie contains the session ID. Otherwise, returns **false**.
boolean isRequestedSessionIdFromURL( )	Returns **true** if the URL contains the session ID. Otherwise, returns **false**.
boolean isRequestedSessionIdValid( )	Returns **true** if the requested session ID is valid in the current session context.

The **HttpServletResponse interface** : The **HttpServletResponse interface enables a servlet to formulate an HTTP response to a** client. Several constants are defined. These correspond to the different status codes that can be assigned to an HTTP response.

EX: **SC_OK indicates that the HTTP request** succeeded

SC_NOT_FOUND indicates that the requested resource is not available.

Method	Description
void addCookie(Cookie *cookie*)	Adds *cookie* to the HTTP response.
boolean containsHeader(String *field*)	Returns **true** if the HTTP response header contains a field named *field*.
String encodeURL(String *url*)	Determines if the session ID must be encoded in the URL identified as *url*. If so, returns the modified version of *url*. Otherwise, returns *url*. All URLs generated by a servlet should be processed by this method.
String encodeRedirectURL(String *url*)	Determines if the session ID must be encoded in the URL identified as *url*. If so, returns the modified version of *url*. Otherwise, returns *url*. All URLs passed to **sendRedirect( )** should be processed by this method.

Method	Description
void sendError(int *c*) throws IOException	Sends the error code *c* to the client.
void sendError(int *c*, String *s*) throws IOException	Sends the error code *c* and message *s* to the client.
void sendRedirect(String *url*) throws IOException	Redirects the client to *url*.
void setDateHeader(String *field*, long *msec*)	Adds *field* to the header with date value equal to *msec* (milliseconds since midnight, January 1, 1970, GMT).
void setHeader(String *field*, String *value*)	Adds *field* to the header with value equal to *value*.
void setIntHeader(String *field*, int *value*)	Adds *field* to the header with value equal to *value*.
void setStatus(int *code*)	Sets the status code for this response to *code*.

The HttpServlet Class

- extends the GenericServlet

Method	Description
void doDelete(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP DELETE request.
void doGet(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP GET request.
void doHead(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP HEAD request.
void doOptions(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP OPTIONS request.
void doPost(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP POST request.
void doPut(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP PUT request.
void doTrace(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Handles an HTTP TRACE request.
long getLastModified(HttpServletRequest *req*)	Returns the time (in milliseconds since midnight, January 1, 1970, GMT) when the requested resource was last modified.
void service(HttpServletRequest *req*, HttpServletResponse *res*) throws IOException, ServletException	Called by the server when an HTTP request arrives for this servlet. The arguments provide access to the HTTP request and response, respectively.

Handling HTTP Requests and Responses

1.Handling HTTP GET Requests

```
<html>
  <head>
    <title>Color Selection</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form  action="ColorGetServlet" method="get">
<B>Color:</B>
<select name="color" size="1">
<option value="Red">Red</option>
<option value="Green">Green</option>
<option value="Blue">Blue</option>
</select>
<br><br>
<input type=submit value="Submit">
</form>

  </body>
</html>
```

```
public class ColorGetServlet extends GenericServlet {

public void service(ServletRequest request, ServletResponse response)throws
ServletException, IOException

{
String color = request.getParameter("color");
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<B>The selected color is: ");
pw.println(color);
pw.close();
}
}
```

```
public class ColorGetServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException

{
String color = request.getParameter("color");
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<B>The selected color is: ");
pw.println(color);
pw.close();
}
}
```

Color Selection

localhost:48278/WebApplication7/

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea...

Color: Red

Red
Green
Blue

Submit

localhost:48278/WebApplication7/ColorGetServlet?color=Red

localhost:48278/WebApplication7/ColorGetServlet?color=Red

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea...

The selected color is: Red

2. Handling HTTP POST Requests

```
<html>
   <head>
      <title>Color Selection</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form  action="ColorGetServlet" method="post" >
<B>Color:</B>
<select name="color" size="1">
<option value="Red">Red</option>
<option value="Green">Green</option>
<option value="Blue">Blue</option>
</select>
<br><br>
<input type=submit value="Submit">
</form>

   </body>
</html>
```

```
public class ColorGetServlet extends GenericServlet {

public void service(ServletRequest request, ServletResponse response)throws
ServletException, IOException

{
String color = request.getParameter("color");
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<B>The selected color is: ");
pw.println(color);
pw.close();
}
}
```

```
public class ColorGetServlet extends HttpServlet {

 public void doPost(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException

{
String color = request.getParameter("color");
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<B>The selected color is: ");
pw.println(color);
pw.close();
}
}
```

localhost:48278/WebApplication7/ColorGetServlet

localhost:48278/WebApplication7/ColorGetServlet

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea...

The selected color is: Red

doGet() and doPost()

- **doGet() shall be used when small amount of data and insensitive data like a query has to be sent as a request.**
  ->doPost() shall be used when comparatively large amount of sensitive data has to be sent. Examples are sending data after filling up a form or sending login id and password

- *In doGet Method the parameters are appended to the URL and sent along with header information In doPost, parameters are sent in separate line in the body*
- *Maximum size of data that can be sent using doget is 240 bytes There is no maximum size for data*
- *Parameters are not encrypted Parameters are encrypted*
- *DoGet method generally is used to query or to get some information from the server Dopost is generally used to update or post some information to the server*
- *DoGet is faster if we set the response content length since the same connection is used. Thus increasing the performance DoPost is slower compared to doGet since doPost does not write the content length*
- *DoGet should be idempotent. i.e. doget should be able to be repeated safely many times This method does not need to be idempotent. Operations requested through POST can have side effects for which the user can be held accountable, for example, updating stored data or buying items online.*
- *DoGet should be safe without any side effects for which user is held responsible*

The Cookie Class

- ***The Cookie class encapsulates a cookie.***
- ***A cookie is stored on a client and contains state information.***
- ***Cookies are valuable for tracking user activities.***

  For example, user visits an online store. A cookie can save the user's name, address, and other information. The user does not need to enter this data each time he or she visits the store.

- ***A servlet can write a cookie to a user's machine via the addCookie( ) method of the HttpServletResponse interface.***

- The information that is saved for each cookie includes the following:

  - ***The name of the cookie***
  - ***The value of the cookie***
  - ***The expiration date of the cookie***
  - ***The domain and path of the cookie***

- There is one constructor for Cookie. It has the signature shown here:

Cookie(String name, String value)

Here, the name and value of the cookie are supplied as arguments to the constructor.

Method	Description
Object clone( )	Returns a copy of this object.
String getComment( )	Returns the comment.
String getDomain( )	Returns the domain.
int getMaxAge( )	Returns the maximum age (in seconds).
String getName( )	Returns the name.
String getPath( )	Returns the path.
boolean getSecure( )	Returns **true** if the cookie is secure. Otherwise, returns **false**.
String getValue( )	Returns the value.
int getVersion( )	Returns the version.
void setComment(String *c*)	Sets the comment to *c*.
void setDomain(String *d*)	Sets the domain to *d*.
void setMaxAge(int *secs*)	Sets the maximum age of the cookie to *secs*. This is the number of seconds after which the cookie is deleted.
void setPath(String *p*)	Sets the path to *p*.
void setSecure(boolean *secure*)	Sets the security flag to *secure*.
void setValue(String *v*)	Sets the value to *v*.
void setVersion(int *v*)	Sets the version to *v*.

The **HttpServletRequest interface** enables a servlet to obtain information about a client request.

Method	Description
String getAuthType( )	Returns authentication scheme.
Cookie[ ] getCookies( )	Returns an array of the cookies in this request.
long getDateHeader(String *field*)	Returns the value of the date header field named *field*.
String getHeader(String *field*)	Returns the value of the header field named *field*.
Enumeration getHeaderNames( )	Returns an enumeration of the header names.
int getIntHeader(String *field*)	Returns the **int** equivalent of the header field named *field*.
String getMethod( )	Returns the HTTP method for this request.
String getPathInfo( )	Returns any path information that is located after the servlet path and before a query string of the URL.
String getPathTranslated( )	Returns any path information that is located after the servlet path and before a query string of the URL after translating it to a real path.
String getQueryString( )	Returns any query string in the URL.
String getRemoteUser( )	Returns the name of the user who issued this request.
String getRequestedSessionId( )	Returns the ID of the session.
String getRequestURI( )	Returns the URI.
StringBuffer getRequestURL( )	Returns the URL.
String getServletPath( )	Returns that part of the URL that identifies the servlet.
HttpSession getSession( )	Returns the session for this request. If a session does not exist, one is created and then returned.
HttpSession getSession(boolean *new*)	If *new* is **true** and no session exists, creates and returns a session for this request. Otherwise, returns the existing session for this request.
boolean isRequestedSessionIdFromCookie( )	Returns **true** if a cookie contains the session ID. Otherwise, returns **false**.
boolean isRequestedSessionIdFromURL( )	Returns **true** if the URL contains the session ID. Otherwise, returns **false**.
boolean isRequestedSessionIdValid( )	Returns **true** if the requested session ID is valid in the current session context.

The **HttpServletResponse interface** : The **HttpServletResponse interface enables a servlet to formulate an HTTP response to a** client. Several constants are defined. These correspond to the different status codes that can be assigned to an HTTP response.

EX: **SC_OK indicates that the HTTP request** succeeded

SC_NOT_FOUND indicates that the requested resource is not available.

Method	Description
void addCookie(Cookie *cookie*)	Adds *cookie* to the HTTP response.
boolean containsHeader(String *field*)	Returns **true** if the HTTP response header contains a field named *field*.
String encodeURL(String *url*)	Determines if the session ID must be encoded in the URL identified as *url*. If so, returns the modified version of *url*. Otherwise, returns *url*. All URLs generated by a servlet should be processed by this method.
String encodeRedirectURL(String *url*)	Determines if the session ID must be encoded in the URL identified as *url*. If so, returns the modified version of *url*. Otherwise, returns *url*. All URLs passed to **sendRedirect( )** should be processed by this method.

Method	Description
void sendError(int *c*) throws IOException	Sends the error code *c* to the client.
void sendError(int *c*, String *s*) throws IOException	Sends the error code *c* and message *s* to the client.
void sendRedirect(String *url*) throws IOException	Redirects the client to *url*.
void setDateHeader(String *field*, long *msec*)	Adds *field* to the header with date value equal to *msec* (milliseconds since midnight, January 1, 1970, GMT).
void setHeader(String *field*, String *value*)	Adds *field* to the header with value equal to *value*.
void setIntHeader(String *field*, int *value*)	Adds *field* to the header with value equal to *value*.
void setStatus(int *code*)	Sets the status code for this response to *code*.

Using Cookies

File	Description
AddCookie.htm	Allows a user to specify a value for the cookie named **MyCookie**.
AddCookieServlet.java	Processes the submission of **AddCookie.htm**.
GetCookiesServlet.java	Displays cookie values.

The HTML source code contains a text field in which a value can be entered. There is also a submit button on the page. When this button is pressed, the value in the text field is sent to **AddCookieServlet** via an HTTP POST request.

The source code for **AddCookieServlet.java gets the** value of the parameter named "data". It then creates a **Cookie object that has the name** "MyCookie" and contains the value of the "data" parameter. The cookie is then added to the header of the HTTP response via the **addCookie( ) method. A feedback message is then** written to the browser.

The source code for servlet2 **invokes** the **getCookies( ) method to read any cookies that are included in the HTTP GET request. The** names and values of these cookies are then written to the HTTP response. Observe that the **getName( ) and getValue( ) methods are called to obtain this information.**

```
<html>
   <head>
      <title>Working on Cookie</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <center>
<form
method="post"
action="AddCookieServlet">
<input type=text name="name" size=25 value="">
<B>Enter your Branch:</B>
<input type=text name="branch" size=25 value="">
<input type=submit value="Submit"></form>

   </body>
</html>
```

The HTML source code contains a text field in which a value can be entered. There is also a submit button on the page. When this button is pressed, the value in the text field is sent to AddCookieServlet via an HTTP POST request.

```
public class AddCookieServlet extends HttpServlet {

   public void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException,
      IOException {
// Get parameter from HTTP request.
String data = request.getParameter("name");
// Create cookie.
Cookie cookie1 = new Cookie("Name", data);
cookie1.setMaxAge(5);

data = request.getParameter("branch");
Cookie cookie2 = new Cookie("Branch", data);
cookie2.setMaxAge(5);
// Add cookie to HTTP response.
response.addCookie(cookie1);
response.addCookie(cookie2);
// Write output to browser.
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<B> Dear "+cookie1.getValue()+" your branch is set to : "+cookie2.getValue());
pw.print("<form action='servlet2' method='post'>");
  pw.print("<input type='submit' value='go'>");
   pw.print("</form>");
```

The source code for AddCookieServlet.java gets the value of the parameter named "data". It then creates a Cookie object that has the name "MyCookie" and contains the value of the "data" parameter. The cookie is then added to the header of the HTTP response via the addCookie( ) method. A feedback message is then written to the browser.
Another submit button is created

```
public class servlet2 extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)throws
     ServletException , IOException{

   response.setContentType("text/html");
   PrintWriter out = response.getWriter();

   Cookie ck[]=request.getCookies();
   out.print("Hello "+ck[0].getValue()+ " the details of your cookie are: ");

   out.println("<B>");
for(int i = 0; i < ck.length; i++) {
String name = ck[i].getName();
String value = ck[i].getValue();
out.println("<Br>Data = " + name + "; value = " + value);
}

   out.close();  }
}
```

The source code for GetCookiesServlet.java invokes the getCookies( ) method to read any cookies that are included in the HTTP GET request. The names and values of these cookies are then written to the HTTP response. Observe that the getName( ) and getValue( ) methods are called to obtain this information.

Working on Cookie

localhost:48278/cookietrial/

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl | Collaboration, Tea... | Five Free C

Enter your Name: Dr. Saritha Chakrasali **Enter your Branch:** ISE Submit

localhost:48278/cookietrial/AddCookieServlet

localhost:48278/cookietrial/AddCookieServlet

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi...

Dear Dr. Saritha Chakrasali your branch is set to : ISE

go

localhost:48278/cookietrial/servlet2

localhost:48278/cookietrial/servlet2

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi... | dhl

Hello Dr. Saritha Chakrasali the details of your cookie are:
Data = Name; value = Dr. Saritha Chakrasali
Data = Branch; value = ISE

Eclipse GlassFish 6.2.1 - Error report

localhost:48278/cookietrial/servlet2

Google Slides | How schools can ra... | Welcome to bnmit.... | Bookmarks | Cisco Webex Meeti... | Important Life Skills... | DSPs advance medi...

HTTP Status 500 - Internal Server Error

---

type Exception report

message Internal Server Error

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
java.lang.NullPointerException: Cannot load from object array because "ck" is null
```

note The full stack traces of the exception and its root causes are available in the Eclipse GlassFish 6.2.1 logs.

Eclipse GlassFish 6.2.1

The **HttpSession interface :**The HttpSession interface enables a servlet to read and write the state information that is associated with an HTTP session.

Method	Description
Object getAttribute(String *attr*)	Returns the value associated with the name passed in *attr*. Returns **null** if *attr* is not found.
Enumeration getAttributeNames( )	Returns an enumeration of the attribute names associated with the session.
long getCreationTime( )	Returns the time (in milliseconds since midnight, January 1, 1970, GMT) when this session was created.
String getId( )	Returns the session ID.
long getLastAccessedTime( )	Returns the time (in milliseconds since midnight, January 1, 1970, GMT) when the client last made a request for this session.
void invalidate( )	Invalidates this session and removes it from the context.
boolean isNew( )	Returns **true** if the server created the session and it has not yet been accessed by the client.
void removeAttribute(String *attr*)	Removes the attribute specified by *attr* from the session.
void setAttribute(String *attr*, Object *val*)	Associates the value passed in *val* with the attribute name passed in *attr*.

The HttpSessionEvent Class

- HttpSessionEvent encapsulates session events.
- It ***extends EventObject*** and is generated when a change occurs to the session.
- It defines this **constructor**:

HttpSessionEvent(HttpSession session)

- Here, *session is the source of the event.*
- **Method**

HttpSession getSession( )

- It returns the session in which the event occurred.

The HttpSessionBindingEvent Class

- The HttpSessionBindingEvent class extends **HttpSessionEvent**.
- It is generated when a listener is bound to or unbound from a value in a HttpSession object.
- It is also generated when an attribute is bound or unbound. Here are its constructors:

HttpSessionBindingEvent(HttpSession session, String name)

HttpSessionBindingEvent(HttpSession session, String name, Object val)

- Here, *session is the source of the event, and name is the name associated with the object that is* being bound or unbound.

The HttpSessionBindingListener Interface

- The **HttpSessionBindingListener interface is implemented by objects that need to be** notified when they are bound to or unbound from an HTTP session. The methods that are invoked when an object is bound or unbound are
- void valueBound(HttpSessionBindingEvent *e)*
- void valueUnbound(HttpSessionBindingEvent *e)*

Here, *e is the event object that describes the binding.*

Session Tracking

- A session can be created via the **getSession( )** method of HttpServletRequest.
- An **HttpSession object** is returned.
- This object can **store a set of bindings** that associate names with objects.
- The setAttribute(), getAttribute(), getAttributeNames( ), and removeAttribute( )

  methods of HttpSession manage these bindings.
- It is important to note **that session state is shared among all the servlets that are associated with a particular client.**

```
<html>
   <head>
      <title>Display date and time</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   </head>
   <body>
      <form

action="currdate" method="post">
         <input type="submit" value="Current Date "/>
      </form>
         </body>
</html>
```

```
public class currdate extends HttpServlet {

  public void doPost(HttpServletRequest request, HttpServletResponse
response)  throws ServletException, IOException {
                        // Get the HttpSession object.
HttpSession hs = request.getSession(true);

                        // Get writer.
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.print("<B>");

                        // Display date/time of last access.
Date date = (Date)hs.getAttribute("date");
if(date != null) {
pw.print("Last access: " + date + "<br>");
}

                        // Display current date/time.
date = new Date();
hs.setAttribute("date", date);
pw.println("Current date: " + date);   }   }
```

File Edit View History Bookmarks Tools Help

http://localho...ss/DateServlet

localhost:8080/servletsclass/DateServlet

Search

Most Visited Home Getting Started Latest Headlines

Current date: Sat Oct 03 04:58:50 IST 2015

File Edit View History Bookmarks Tools Help

http://localho...ss/DateServlet × http://localho...ss/DateServlet × +

localhost:8080/servletsclass/DateServlet

Search

Most Visited Home Getting Started Latest Headlines

Last access: Sat Oct 03 04:58:50 IST 2015
Current date: Sat Oct 03 04:59:12 IST 2015