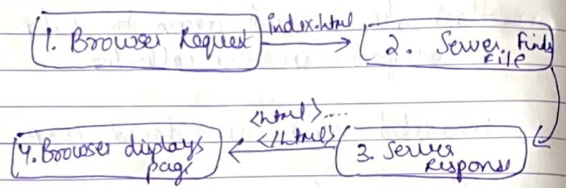
HTTP is a protocol that clients and servers use on the web to communicate

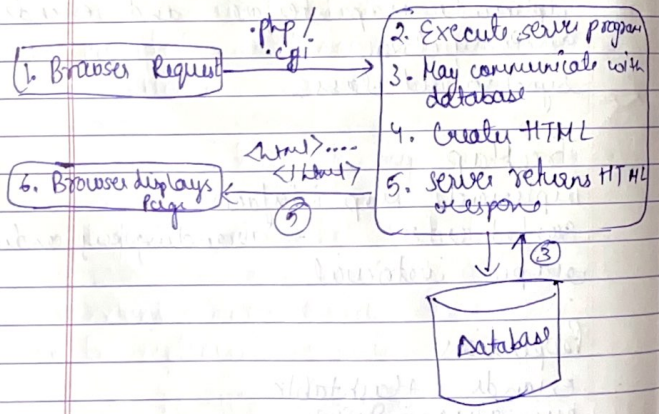
Date.
Page No.

Server-side programming

① Static HTTP transaction
- Browser requests for index.html
Server responds with HTML file



② Dynamic HTTP transaction
- Browsers requests OrderServlet.class, server runs program that creates HTML
Server returns HTML to browsers.



Apache HTTP server
Microsoft Internet Information Services.
eg. Java System Web server.

Page No.

Web Server
→ A computer having server software installed within it which serves up web pages.
→ A program that uses client/server model and World Wide Web's HyperText Transfer Protocol (HTTP)
→ Responsible for accepting HTTP requests from clients and serving HTTP responses which are web pages such as HTML doc.

A web component is a software entity that runs on a web server
→ Servlets
→ Java Server Pages (JSPs)
} J2EE specification defines two web components

CGI (Common Gateway Interface)
CGI technology enables the web server to call an external program and pass HTTP request information to the external program to process the request
For each request, it starts a new process

Disadvantages → If clients ↑es, it takes more time for sending the response

For each request, it starts a process and the web servers is limited to start processes.
It uses platform dependent lang — C, C++, perl etc

Date.
Page No.

There are many advantages of servlets over CGI.

→ web container creates threads for handling the multiple requests to the servlet.

• Threads have many benefits over Processes
→ They share a common area
→ They are lightweight
→ Cost of communication are low.

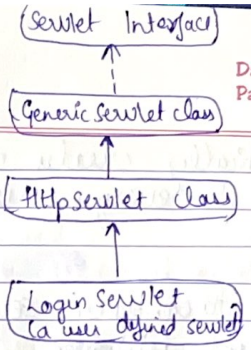
Servlets are robust, portable, secure and have better performance.

Servlet architecture Overview
Java Servlet API are in packages javax.servlet and javax.servlet.http
• Provides interfaces and classes for writing servlets

All servlet must implement servlet interface
• Defines life cycle methods

Extend Generic Servlet class
• To implement generic services

Extend HttpServlet class
• To implement HTTP specific services



Date.
Page No.

Web Deployment Descriptor - web.xml
An XML file is a deployment descriptor it allows
- Mapping from URIs to application resources
- Initialization parameters
- Security constraints
- registration of listeners and filters.

Web Containes
Web components and their containes run on J2EE servers
• Provides Execution environment for servlets and JSPs of a web application
• Manages execution of JSP and servlet components for J2EE app.

Roles →
1) Provides an easy way for servlets to talk to web server.
2) Controls lifecycle of servlets

Date.
Page No.

3) Automatically creates a new Java thread for every servlet request it receives

4) Enables to configure security in XML deployment descriptor

5) Does JSP processing

Handling Servlet requests

Container creates 2 objects on receiving a request for a servlet HttpServletRequest and HttpServletResponse

Finds right servlet based on URL in the request

Creates a thread for that request

Passes request and response objects to the servlet thread.

Calls servlet's service() method

The service() method in turn calls doGet and doPost based on type of request

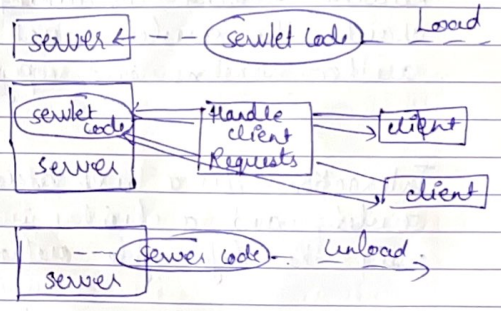
The doGet Method generates dynamic page and captures it in response object

Converts response object into an HTTP response on completion of thread.

Sends this response to the client

Finally deletes the request and response objects.

Life cycle of servlet



Servlet Interface

→ init(ServletConfig)
Initialize the servlet. Runs once before any requests can be serviced

→ service(ServletRequest, ServletResponse)
Processes a single request from the client

Date.
Page No.

→ destroy()
This method releases all the resources

→ getServletConfig()
Returns a servlet config object and this object contains any initialization parameters and startup configuration information for this servlet.

→ getServletInfo()
Returns a string containing information about the servlet, such as its author, version, and copyright

Interaction b/w a web browser, a servlet, and a client is controlled using the life cycle methods

init() service() destroy()

The init() and destroy() methods will be called only once during the life time of your Servlet

The service() and it's broken down methods (doGet(), doPost()) will be called as many times as requests are received for them by web container

Date.
Page No.

Two versions of init method

init() → use this when servlet does not need any specific initialization

init(ServletConfig) – use this when servlet needs to check specific settings before completing initialization.

Every time the server receives an incoming request for a servlet, it (web container) generates a new thread and calls the service method.

The service method checks the HTTP request type and calls the appropriate doXXX method

Role
→ To extract info from the request
→ Access external resources
→ Populate the response based on the info

Server may unload a servlet instance
→ if the servlet has been idle
→ if server shuts down
→ if web app is undeployed

Date.
Page No.

Before it removes the servlet, it calls destroy methods only once

Before servlets gets destroyed, this method helps in

→ cleanup operations such as closing database connections

→ Halting background details

→ Releasing other resources such as io streams

Syntax: public void destroy ()

Web Server	Application Server
It comprises of web container only	It comprises of web container and EJB container
Useful for static content	useful for dynamic content
Utilizes less resources	Utilizes more resources
Multithreading is supported	Multithreading is not supported

Date.
Page No.

Capacity is lower	Capacity is higher
HTML and HTTP protocols are used	GUI as well as HTTP and RPC/RMI protocols are used.
eg ~~Web server~~ Nginx, Apache HTTP server	Glassfish, JBoss

How Servlet works

The server checks if the servlet is requested for first time

If yes, web container does the following task

- loads the servlet class
- instantiates the servlet class.
- calls the init method passing the ServletConfig object.

else

- calls the service method passing request and response objects

The web container calls the destroy method when it needs to remove the servlet such as at time of stopping server or undeploying the project.

Date.
Page No.

WAR file (web archive)
A war file contains all the contents of a web application. It reduces the time duration for transferring file.

The war file combines all the files into a single unit. So it takes less time while transferring file from client to server.

To create a war file, jar tool of JDK is needed

-c is used to create the war file

jar -cvf ____.war*

-c used to create file
-v generate the verbose output
-f specify the archive file name

* signifies all the files of the directory (including subdirectory)

Date.
Page No.

ServletConfig interface
Provided to a servlet upon initialization by the web container

Single read only interface
String getInitParameter(String name)
Enumeration getInitParameterNames()
String getServletName()

can also access ServletContext

ServletContext interface
Allows a servlet to communicate with the servlet container

Access container-managed resources, dispatch requests, write to logs

Defines a set of methods that a servlet uses to communicate with its servlet container
eg to get the MIME type of a file, dispatch requests, or write to a log file.

Key
This is one context "web app" per JVM

ServletContext object is contained within ServletConfig object, which the servlet container provides the servlet when the

Date.
Page No.

servlet is initialized

Resources such as index.html can be accessed through web server or by servlet

→ request.getContextPath()
to identify its context Path

→ getResource() and getResourceAsStream (request.getContextPath() + "index.html")

To retrieve context-wide initialization parameters, servlet uses getInitParameter() and getInitParameterNames()

To access a range of information about the local environment, shared with other servlets in same servlet context, servlet uses getAttribute(), setAttribute(), removeAttribute(), getAttributeNames()

Date.
Page No.

RequestDispatcher Interface

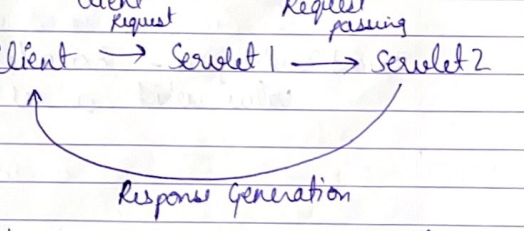
The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the way of servlet collaboration.

Used in order to FORWARD or INCLUDE a request from one servlet to another

RequestDispatcher.forward(request, response)
RequestDispatcher.include(request, response)

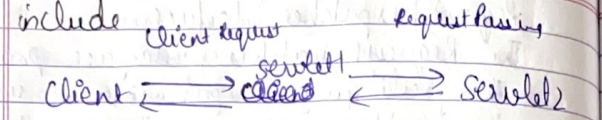
Both these methods take ServletRequest and ServletResponse object as an argument

forward



Forwards a request from a servlet to another resource on the server

include



Includes the content of a resource in the response.

The getRequestDispatcher() method of ServletRequest interface returns the object of RequestDispatcher

first.html - a form containing a text field and a submit button

FirstServlet.java - accepts user name and forwards it to SecondServlet

SecondServlet.java - Extracts the username value which is set in FirstServlet

Generic Servlet	HttpServlet
A class that implements Servlet, ServletConfig and Serializable interface that provide the implementation of all methods of these interface except the service method	A class which extends the GenericServlet class and implements Serializable interface that provides HTTP specific methods like doGet, doPost etc
Protocol-independent	Protocol-dependent
service method is abstract	service method is non-abstract
subclass of ServletI	subclass of GenericServlet
defined by javax.servlet package	defined by javax.servlet.http package
not commonly used	commonly used

HTTP request

Date.
Page No.

The request sent by the computer to a web server, contains all sorts of potentially interesting info is known as HTTP requests.

GET
ask to get the resource at the requested URL

POST
asks the server to accept the body info attached. It is like GET request with extra info send with the request

HEAD
asks for only the header part of whatever a GET would return. Just like GET but with no body.

PUT
Says to put enclosed info at the requested URL

DELETE
Says to delete the resource at the requested URL

Cookies

Date.
Page No.

- javax.servlet.http.Cookie.

A cookie is a small bit of textual information sent to the client by a web server and web server can later read it back from the browser.

The process of using cookies in servlets
- Servlets sends a cookie with its response to the client
- The client saves the cookie
- The client returns a cookie back with subsequent requests

Uses of cookies
- Identifying a user during an e-commerce session
- Avoiding username & password
- Customizing a website

Limitations
- a cookie size is limited to 4KB
- supports 20 cookies per website
- supports 30 cookies in total

Creating cookies - use a constructor
- Cookie (String name, String value)

Date.
Page No.

Adding cookies to a response -
Assume response is an HttpServletResponse
• response.addCookie (cookie1)

Retrieving cookies from a request -
Assume request is an HttpServletRequest
• request.getCookies ()
// returns array of Cookie & null.

Cookie [] cookies = request.getCookies()
String name = cookies[i].getName();
String value = cookies[i].getValue();

Two types of cookies
Non-persistent
It is valid for single session only.
It is removed each time when user closes the browser

Persistant
It is valid for multiple session. It is not removed each time when user closes the browser. It is removed only if user logout or signout

Date.
Page No.

Advantage
Simplest technique of maintaining the state
Cookie are maintained at client side.

Disadvantage
It will not work if cookie is disabled from the browser

Only textual information can be set in Cookie object

Gmail uses cookie technique for login.
If you disable the cookie, gmail won't work.

→ public void setMaxAge (int expiry)
set the max age of the cookies in seconds

→ public String getName()
returns the name of the cookie. The name cannot be changed after creation

→ public String getValue()
returns the value of the cookie

→ public void setName (String name)
changes the name of the cookie

Before servlets CGI (Common gateway Interface) scripting language was common as a server-side programming language

Date.
Page No. 190

Java based platform independent → Servlets → robust and scalable

they are

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page.

Servlet is an API that provides many interfaces and classes including documentation

Servlet is an interface that must be implemented for creating any servlet

Servlet is a class that extends the capabilities of the servers and responds to the incoming requests It can respond to any requests.

Servlet is a web component that is deployed on the server to create a dynamic web page.

Qq

{Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse etc}

Advantages

1) Better performance: because it creates a thread for each request

- A Java class that runs on a web server and dynamically handles client requests.
- It extends the functionality of a web server by receiving client requests and dynamically generating a response.

Date.
Page No. 191

2) Portability
3) Secure
4) Robust: JVM manages servlets, so there is no need to worry about Memory leak, garbage collection etc.

Life Cycle of a Servlet

The web container maintains the life cycle of a servlet instance.

1) Servlet class is loaded
The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.

2) Servlet instance is created
The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.

3) init() method is invoked
The web container calls the init method only once after creating the

servlet instance. The init method is used to initialize the servlet. It is the life cycle method of the javax.servlet.Servlet interface.

4) Service method is invoked
The web container calls the service method each time when request for the servlet is received. If servlet is not initialized it follows the first three steps then calls the service method. If servlet is initialized, it calls the service method.

5) destroy method is invoked
The web container calls the destroy method before removing the servlet instance from the service.

There are three ways to create the servlet

1) By implementing the Servlet interface
2) By inheriting the GenericServlet class
3) By inheriting the HttpServlet class

The HttpServlet class is widely used to create the servlet because it provides methods to handle http requests such as doGet(), doPost, doHead(), etc.

```
package com.pract.test;
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

public class Demo extends HttpServlet
{
    public void doGet(HttpServletRequest req,
            HttpServletResponse res) throws
                ServletException, IOException
    {
        String str = req.getParameter("H");
        PrintWriter out = res.getWriter();

        out.println(str.toUpperCase());
        out.println(str + "Hello");

        out.close();
    }
}
```

→ For compiling the servlet, jar file is required to be loaded.

servlet-api.jar Apache Tomcat

→ The deployment descriptor is an xml file, from which web container gets the information about the servlet to be invoked.

```
<web-app>
<servlet>
<servlet-name> abc </servlet-name>
<servlet-class> car.pret.test.Demo </servlet-class>
</servlet>

<servlet-mapping>
<servlet-name> abc </servlet-name>
<url-pattern> /xyz </url-pattern>
</servlet-mapping>
</web-app>
```



```
<html>
<head>
<title> — </title>
</head>
<body>
<form action = "xyz"  method = "post">
    Enter your name <input type = "text"
                         name = "n" />
    <br>
    <input type = "submit"  value = "OK">
</form>
</body>
</html>
```

(limited data can be send) (insensitive data)

note: The doGet() method is used for getting the information from the server (used for static contents)

The doPost() method is used for sending the information to the server (used for dynamic contents)
→ unlimited data can be send
→ sensitive data

note: The service() method of servlet is invoked to inform the Servlet about the client requests. This method uses ServletRequest object to collect the data requested by the client

This method uses ServletResponse object to generate the output content.

Diff B/w

~~Applets~~ Servlets

Methods doGet() and doPost() in HttpServlet class receives appropriate client request, and formats a response using 2 arguments

Objects are created by servlet container

An HttpServletRequest Object - encapsulates data from the client

An HttpServletResponse Object - encapsulate response to the client

HttpServletRequest Interface (client to servlet)

It provides methods that allows to retrieve incoming information.

Methods to read parameters from a form
getParameter(String pname)
getParameterNames()
getParameterValues(String pname)

HttpServletResponse Interface (servlet to client)

allow to specify outgoing info.

Methods: getWriter
& setContentType
sendRedirect

Get	Post
limited data can be sent	large amount of data can be sent
Not secure data is expose in URL bar	Secure data is not expose in URL bar
can be bookmarked	cannot be bookmarked
Idempotent. Second request will be ignored until response of first request is delivered	non-idempotent
More efficient	less efficient