MODULE-8
SERVLETS

Course Topics

- → Module 1
  - » Introduction to Java
- → Module 2
  - » Data Handling and Functions
- → Module 3
  - » Object Oriented Programming in Java
- → Module 4
  - » Packages and Multi-threading
- → Module 5
  - » Collections
- → Module 6
  - » XML
- → Module 7
  - » JDBC
- → Module 8
  - » **Servlets**
- → Module 9
  - » JSP
- → Module 10
  - » Hibernate
- → Module 11
  - » Spring
- → Module 12
  - » Spring, Ajax and Design Patterns
- → Module 13
  - » SOA
- → Module 14
  - » Web Services and Project

Objectives

At the end of this module, you will be able to:

→ Understand Client Server Architecture

→ Understand Server types

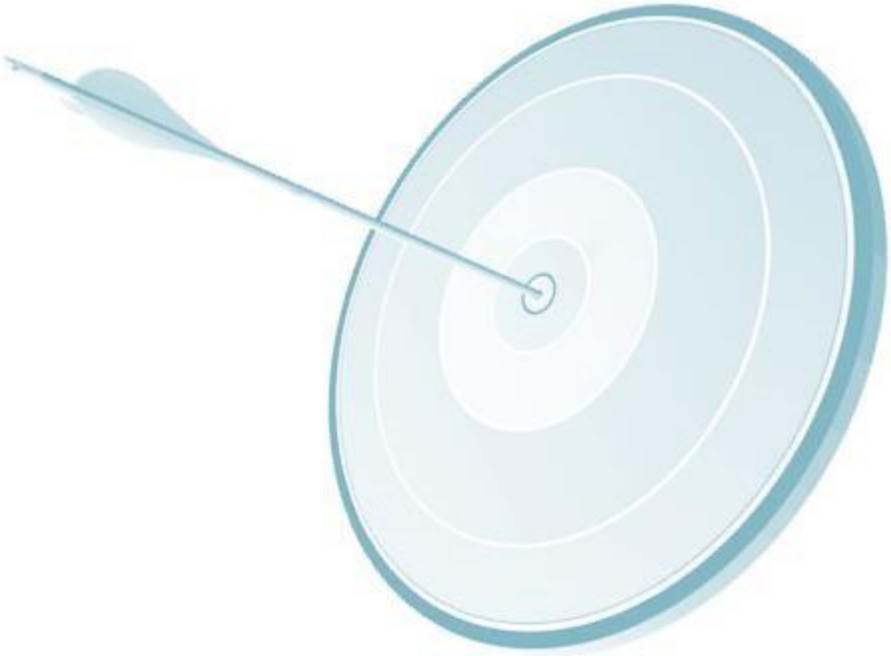
→ Use Http and its methods

→ Understand servlet types and exceptions

→ Implement servlets

→ Understand and Implement Session Tracking

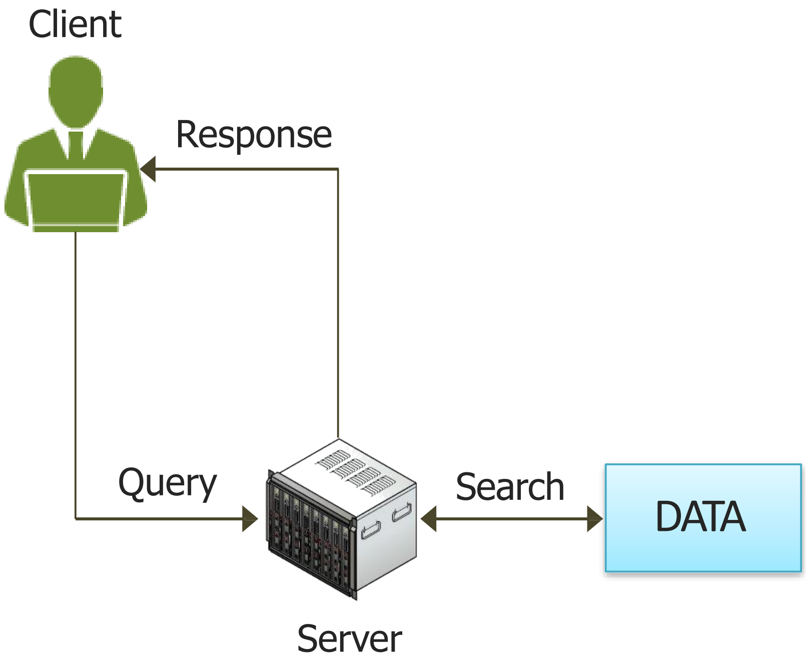
→ Use Filters



Internet

Internet is global system of interconnected computers which uses standard protocol TCP/IP which connects billions of computers across the world.

Client Server Technology



→ In the client server technology, client requests for a service. This request is reached to server through TCP/IP.

→ Upon receiving the request server interprets the request and provides the required service to the client.

→ Server and client can be on the same machine.

→ Server and client can be in different machines. In this case, Server can serve many clients at a time.

Annie’s Question

Can you give a practical example of Client Server Technology?

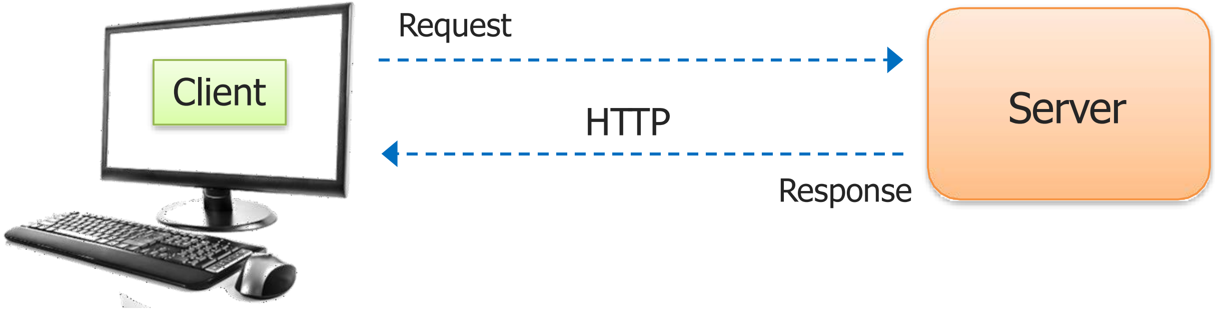
Annie’s Answer

One printer can be connected to many computers. This printer server will print from all the computers when requested for.

Web Technology

→ Client sends the request to the server on internet.

→ Client uses HTTP protocol to send the request.



Client

Request

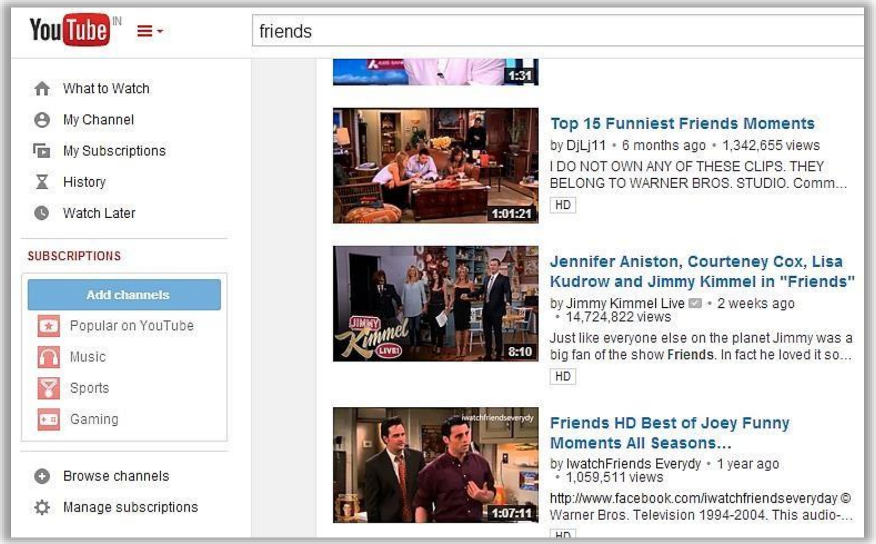
HTTP

Response

Server

Web Technology - Example

YouTube, Amazon.com etc., uses web technology. When the user logs into this site and clicks on the selected item, server recognizes what the user/client needs and responds to the client request by displaying the desired information/image/page.



friends

- What to Watch
- My Channel
- My Subscriptions
- History
- Watch Later

SUBSCRIPTIONS

Add channels

- Popular on YouTube
- Music
- Sports
- Gaming

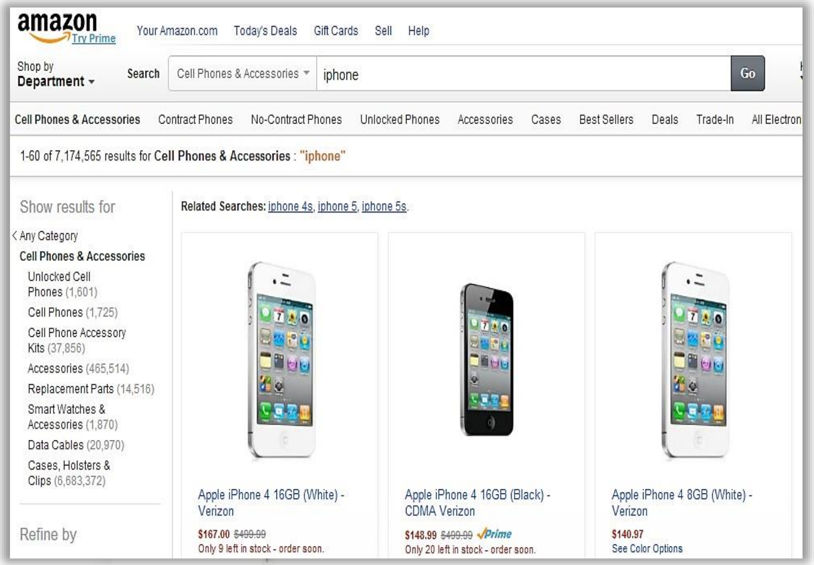
- Browse channels
- Manage subscriptions

1:31

Top 15 Funniest Friends Moments
by DjLj11 • 6 months ago • 1,342,655 views
I DO NOT OWN ANY OF THESE CLIPS. THEY BELONG TO WARNER BROS. STUDIO. Comm...
HD
1:01:21

Jennifer Aniston, Courteney Cox, Lisa Kudrow and Jimmy Kimmel in "Friends"
by Jimmy Kimmel Live • 2 weeks ago • 14,724,822 views
Just like everyone else on the planet Jimmy was a big fan of the show Friends. In fact he loved it so...
HD
8:10

Friends HD Best of Joey Funny Moments All Seasons...
by IwatchFriends Everydy • 1 year ago • 1,059,511 views
http://www.facebook.com/iwatchfriendseveryday © Warner Bros. Television 1994-2004. This audio-...
HD
1:07:11



amazon Try Prime

Your Amazon.com | Today's Deals | Gift Cards | Sell | Help

Shop by Department

Search: Cell Phones & Accessories | iphone | Go

Cell Phones & Accessories | Contract Phones | No-Contract Phones | Unlocked Phones | Accessories | Cases | Best Sellers | Deals | Trade-In | All Electron

1-60 of 7,174,565 results for **Cell Phones & Accessories** : "iphone"

Show results for

- ‹ Any Category
- **Cell Phones & Accessories**
  - Unlocked Cell Phones (1,601)
  - Cell Phones (1,725)
  - Cell Phone Accessory Kits (37,856)
  - Accessories (465,514)
  - Replacement Parts (14,516)
  - Smart Watches & Accessories (1,870)
  - Data Cables (20,970)
  - Cases, Holsters & Clips (6,683,372)

Refine by

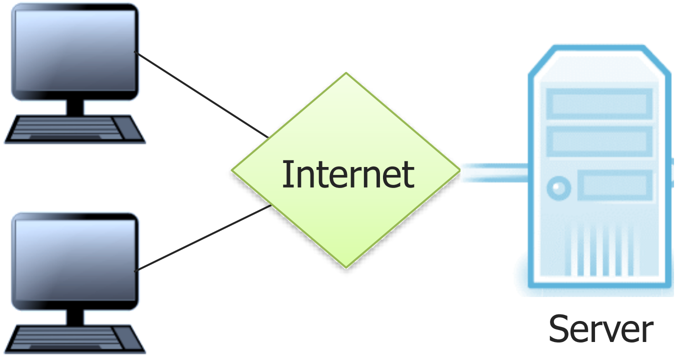
Related Searches: iphone 4s, iphone 5, iphone 5s.

Apple iPhone 4 16GB (White) - Verizon
$167.00 ~~$499.99~~
Only 9 left in stock - order soon.

Apple iPhone 4 16GB (Black) - CDMA Verizon
$148.99 ~~$499.99~~ Prime
Only 20 left in stock - order soon.

Apple iPhone 4 8GB (White) - Verizon
$140.97
See Color Options

Internet Server

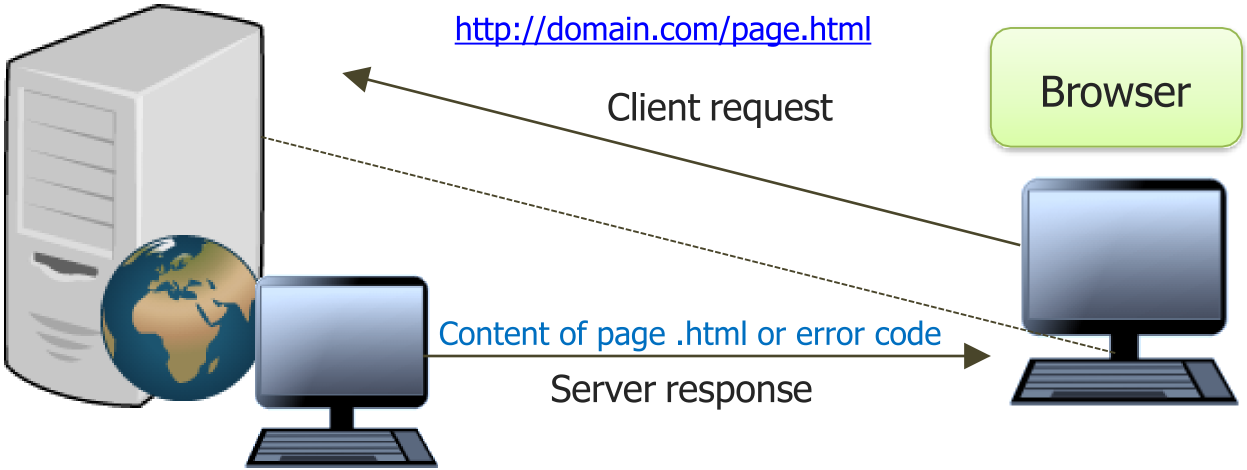


Browser is a client which sends the request to the server. This request is transmitted on internet to the server.

Server is a software which receives the request and passes it on to the corresponding program running on the server software.

For example, we can login to google server and select any program of the google like gtalk, Gmail, blog, translator etc., based on our request, google server will call the corresponding program.

Client - Server Architecture of the Internet



http://domain.com/page.html

Browser

Client request

Content of page .html or error code

Server response

Server Types

There are two types of servers:

→ Web server
→ Application server

Web server takes the http request and responds to the client. Some of the web servers are Apache Tomcat, IIS etc.

Application server is an extension of Web server. In addition to the web server, it provides the following facilities:

→ Load balancing
→ Resource management (Connection pooling)
→ State management

Some of the Application servers are Jboss, Web Logic, Web Sphere, GlassFish etc.

HTTP

HTTP stands for Hyper Text Transfer Protocol.

Standard HTTP port is 80.

HTTPS stands for HTTP + Secured

If the client/browser uses https protocol then data from the client/browser will be encrypted and sent on internet to the server. Server decrypts the data, interprets and serves the client.

HTTP - Methods

Some of the HTTP methods are:

→ GET – To retrieve information from the given server.

→ HEAD – Same as GET but transfers only header section.

→ POST – For sending the data to the server.

→ DELETE – To delete a resource.

→ PUT – The PUT method requests that the enclosed entity be stored under the supplied Request-URI.

→ TRACE – The TRACE method is used to invoke a remote, application-layer loop-back of the request message.

→ CONNECT – This specification reserves the method name CONNECT for use with a proxy that can dynamically switch to being a tunnel.

Difference between GET and POST

GET	POST
GET will display parameters as part of URL.	POST will send the user's data/parameters as part of request body.
GET is used to send Text information.	POST can be used to send large files and BLOB also.
DoGet() is used for GET.	DoPost() will be used for POST at server side.
GET is less secured as it is seen in URL	POST is secured as it will be sent as part of Request body.
GET URL can be stored in browser history.	POST will not be stored in browser history.

Where and How do we use GET and POST?

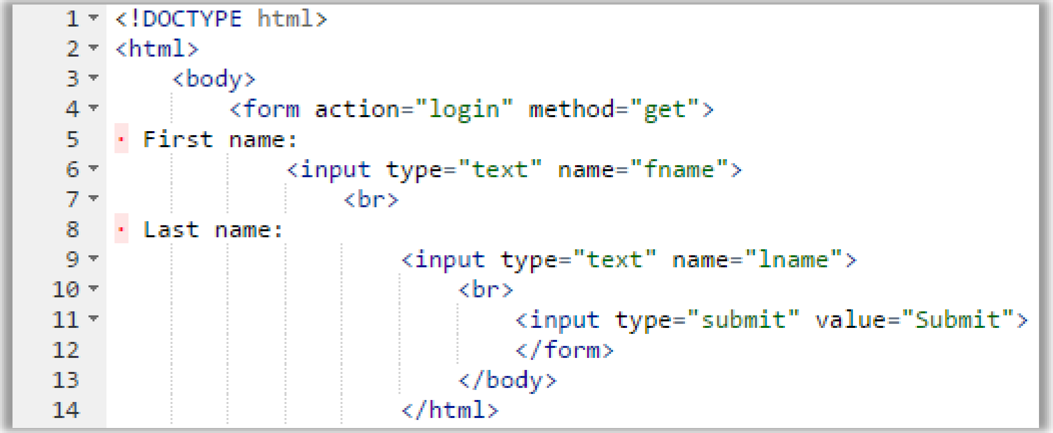
Where do we use GET and POST?

All places where the client data needs to be sent to the server on Internet, GET and POST methods can be used.

How do we use GET and POST?

When GET or POST is used, the client data is transmitted on internet and will be sent to the server. Server receives data in Request object. Request object content is extracted to get the data given by the client.

GET - Example

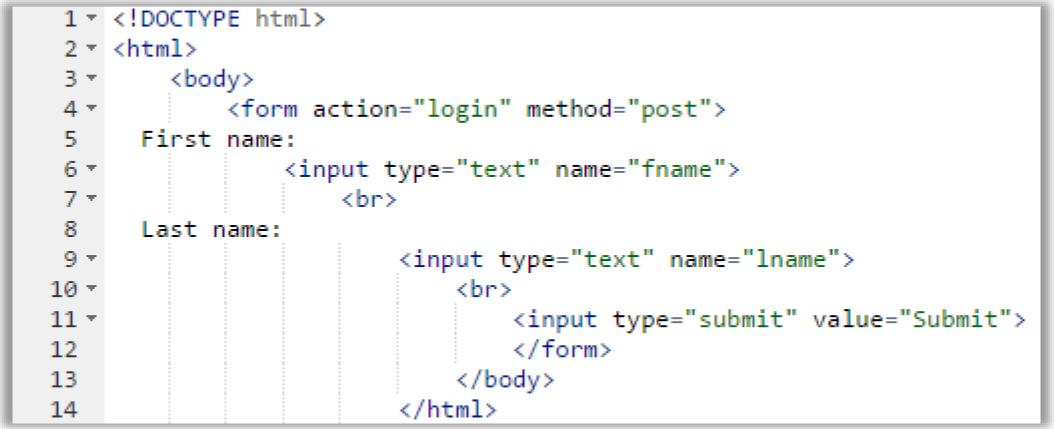


```
<!DOCTYPE html>
<html>
    <body>
        <form action="login" method="get">
  First name:
            <input type="text" name="fname">
                <br>
  Last name:
                    <input type="text" name="lname">
                        <br>
                            <input type="submit" value="Submit">
                            </form>
                        </body>
                    </html>
```

→ This HTML code displays First Name and Last Name edit boxes and submit button.

→ When the submit button is clicked, data entered in first name and last name will be sent to the server as part of URL and will execute login servlet.

Post - Example



```
<!DOCTYPE html>
<html>
    <body>
        <form action="login" method="post">
  First name:
            <input type="text" name="fname">
                <br>
  Last name:
                    <input type="text" name="lname">
                        <br>
                            <input type="submit" value="Submit">
                            </form>
                        </body>
                    </html>
```

Above HTML code works similar to previous HTML code but only the difference is method of sending the data from client is using POST rather than GET.

This will send the client data in the message body rather than as part of URL.

Annie’s Question

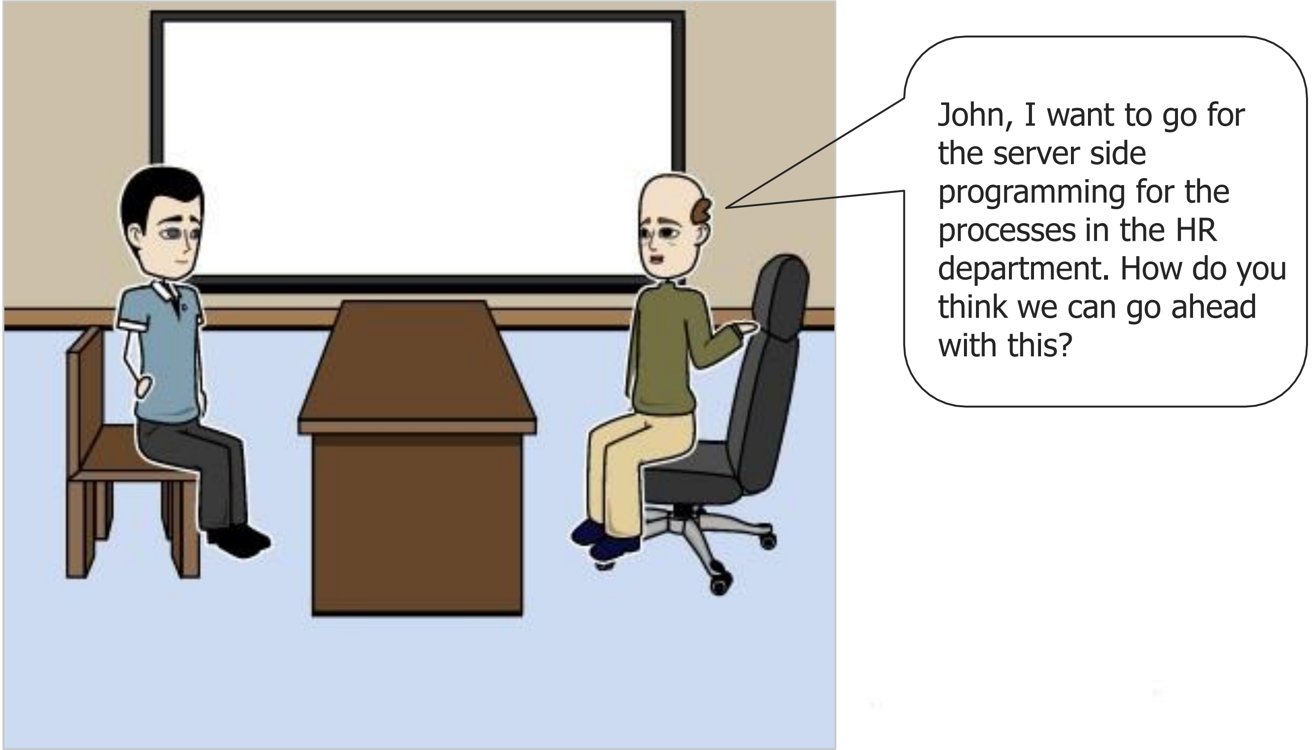
Can you think of one practical scenario where GET and POST is required?

Annie’s Answer

POST is used when we need to send confidential data. For example, entering login information in gmail, POST can be used. GET is used in all other cases.

Servlet

John to Program Processes…

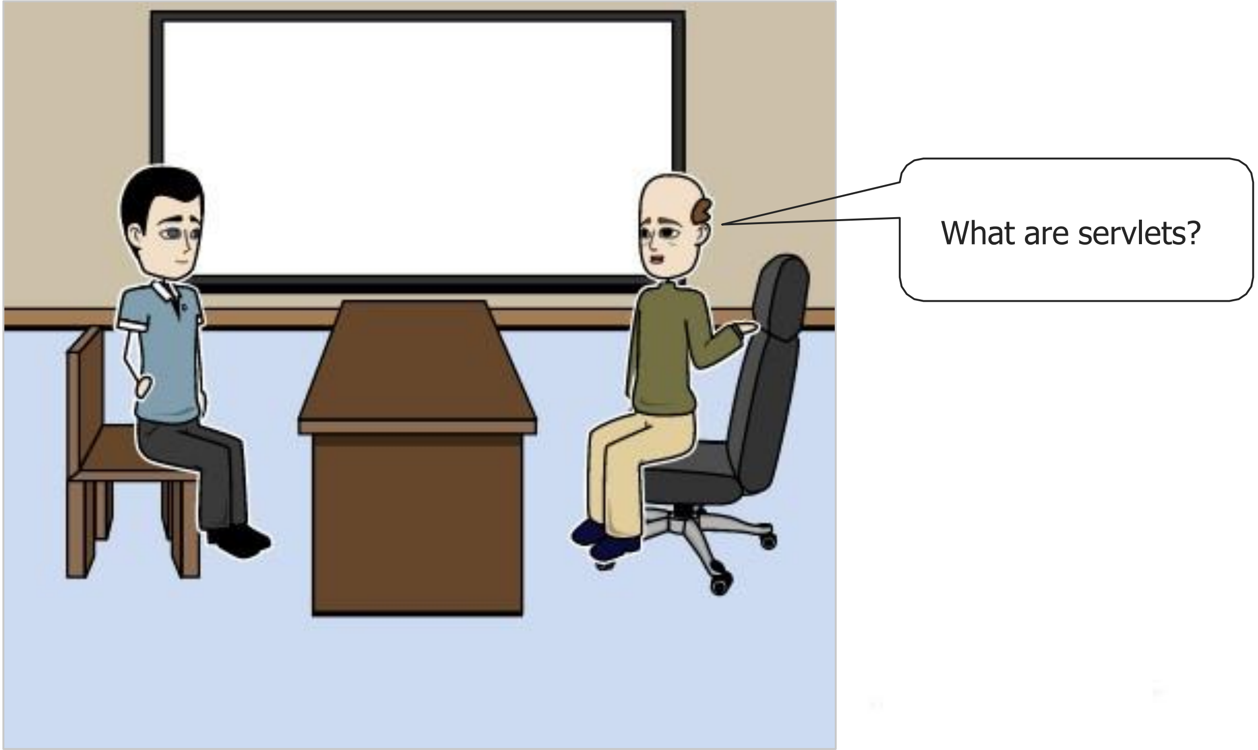


John, I want to go for the server side programming for the processes in the HR department. How do you think we can go ahead with this?

John to Program Processes…

Sir, we can use Servlet programs. We can develop the screen using HTML. For the computation we can use Servlet programming. Also, we can use Mysql server as the RDBMS and connect it with Servlets.

John to Program Processes...



John explains Servlet

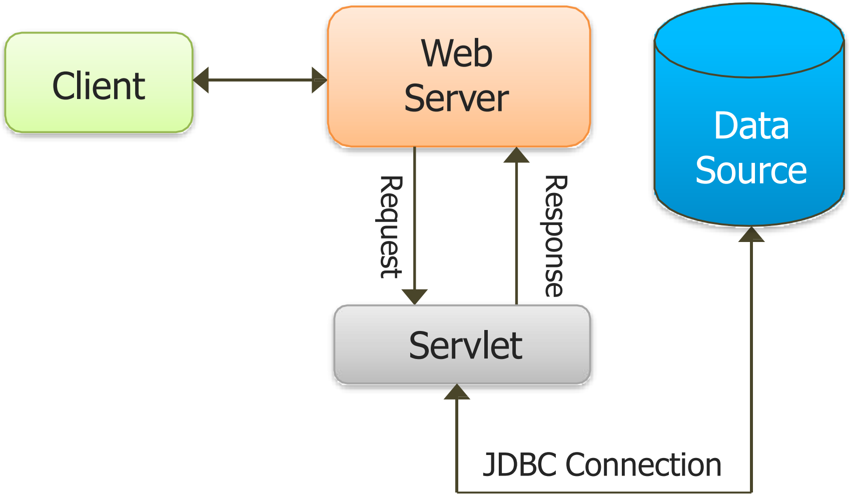
Servlet is a Java technology to create dynamic web pages.

Why Servlets?

Why Servlets?

In CGI, for each new request a new process has to be created. With this memory consumption becomes more and the performance of the system goes down. To avoid this issue Java came up with servlets.

What is a Servlet?



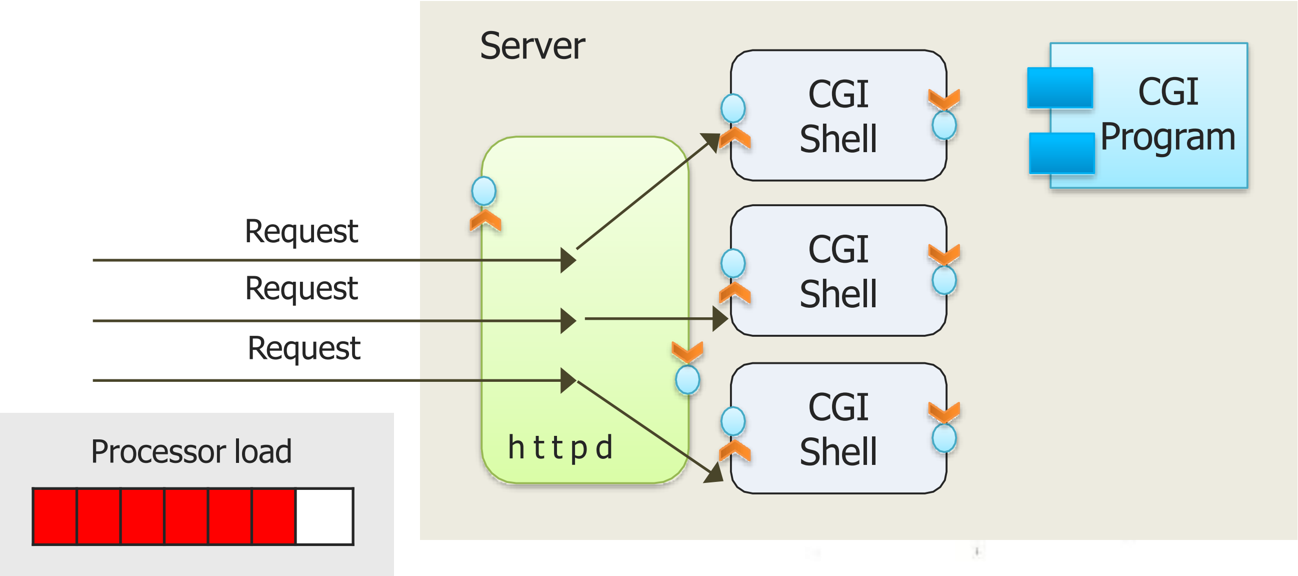
→ Servlet is a Java technology to create dynamic web pages.

→ Servlet is a web component which is deployed on a web server/app server to create dynamic web pages.

→ When the client sends the request, server receives it and sends to the corresponding servlet program. Servlet program interprets the request and responds accordingly.

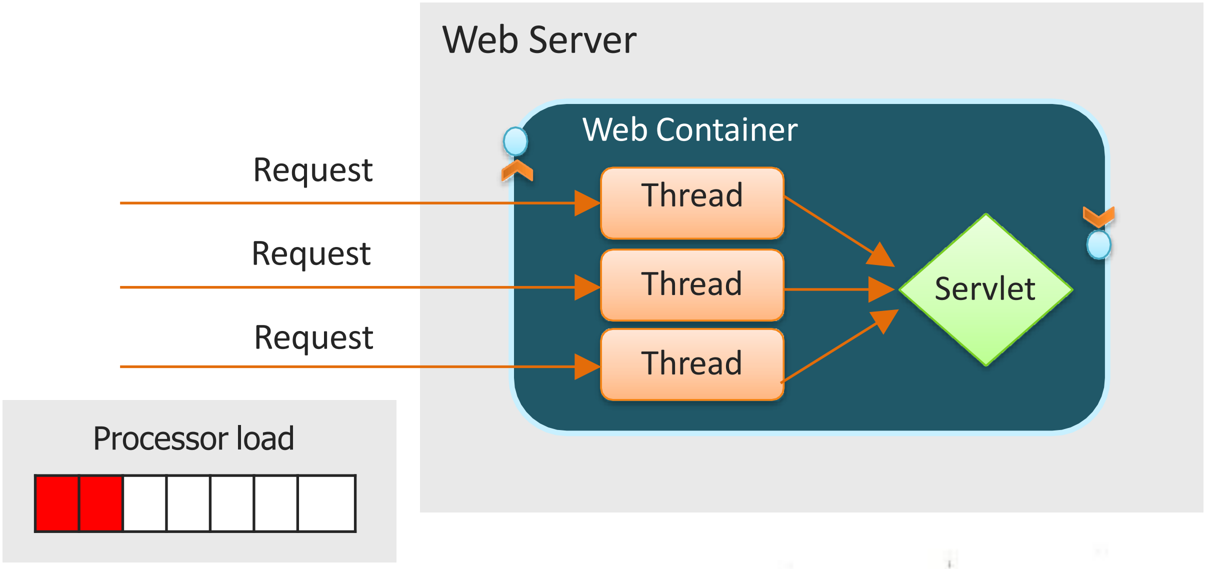
Why Servlets are used over CGI?

For every request from the client a new CGI program is created in the server system. With this the memory consumption is very high and performance level is very low. There are high chances of system crash too.



Why Servlets are used over CGI? (contd.)

To resolve this issue, servlets are used. Servlet engine or web server creates a new thread for each client request instead of creating a new program for every request.



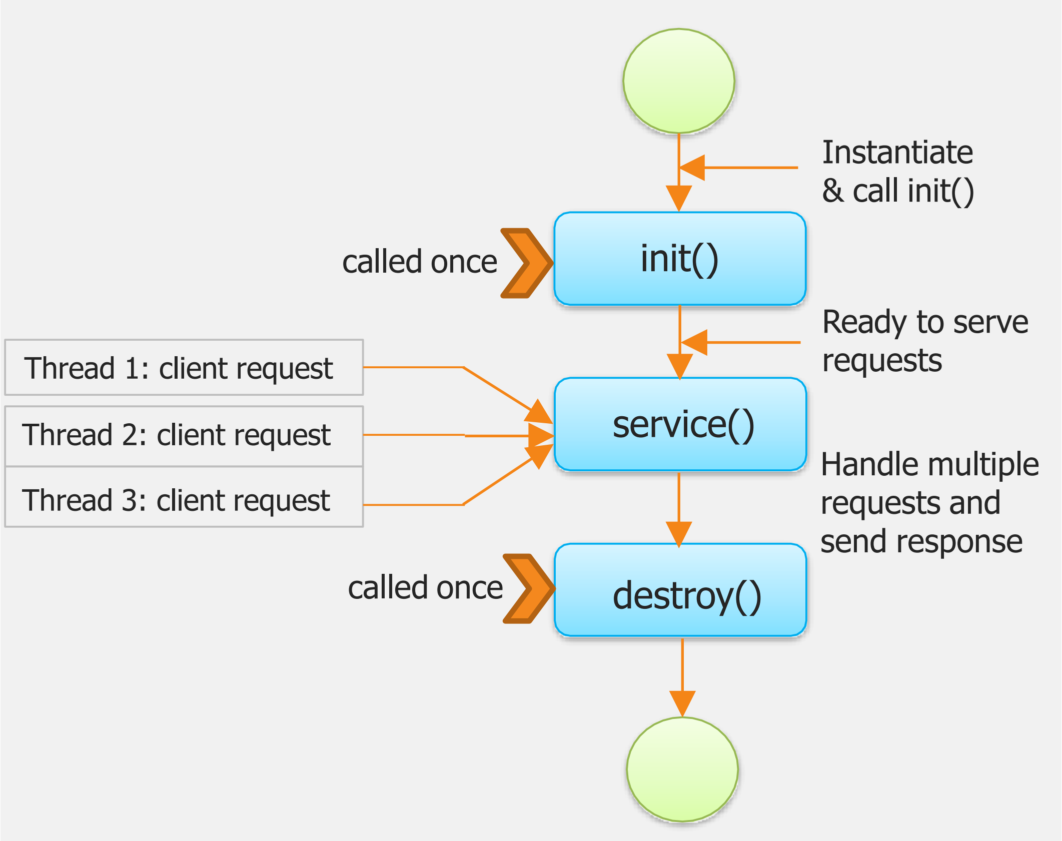
Annie’s Question

What benefit we will get if a new thread is created for each request rather than creating a new program for each request?

Annie’s Answer

When a new thread is created for each request, memory is saved and server can serve many clients and performance of the project is very good.

Servlet Life Cycle



After the servlet program is done, it has to be deployed in Apache tomcat.

When the tomcat comes up, servlet is loaded in the memory. At this time init() method is called.

This is for initializing the attributes of the servet. It is like constructor.

Service(), when any client request comes service() method is invoked.

Destroy() method is called when servlet is about to be removed from memory.

Servlet

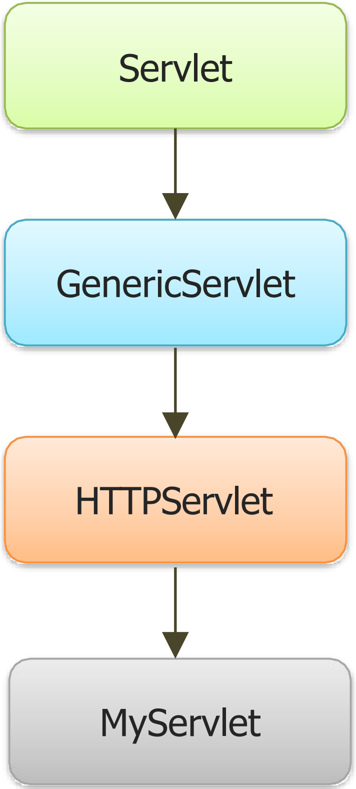
→ Servlet is a interface.

→ Servlet interface provides init(), service() and destroy() method to implement.

→ GenericServlet() implements servlet interface.

→ HttpServlet extends GenericServlet and provides HTTP methods like get, post etc.

Servlet Types



Servlet Configuration with Eclipse

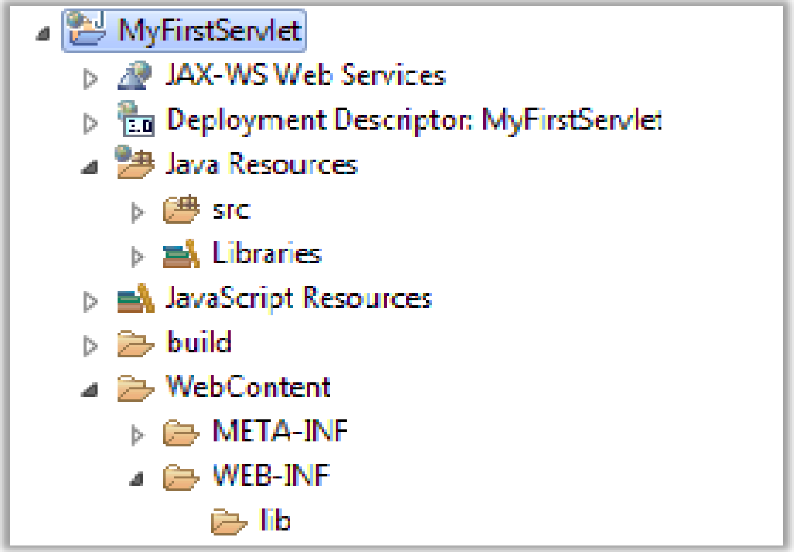
Download Apache tomcat 7.0

Open Eclipse → Click on Window menu → Preferences → Server → Runtime Environment → Add → Select the tomcat version 7.0 → provide the directory of tomcat installed directory → ok.

Project Creation in Eclipse

Ensure you are on J2EE perspective.

Click on File → New → Dynamic Web Project.
Enter the name of the project and click on finish.
If the project name is given as MyFirstServlet, Eclipse will create the following directories by default:



- MyFirstServlet
  - JAX-WS Web Services
  - Deployment Descriptor: MyFirstServlet
  - Java Resources
    - src
    - Libraries
  - JavaScript Resources
  - build
  - WebContent
    - META-INF
    - WEB-INF
      - lib

Information on Directories of Servlet Project

Src → To place Java/servlet files.

WebContent → To place html and jsp files.

WEB-INF → to place xml files.

Lib → All the libraries which are required for the project.

Annie’s Question

What is the difference between HTML and XML?

Annie’s Answer

XML carries the data and HTML displays the data.

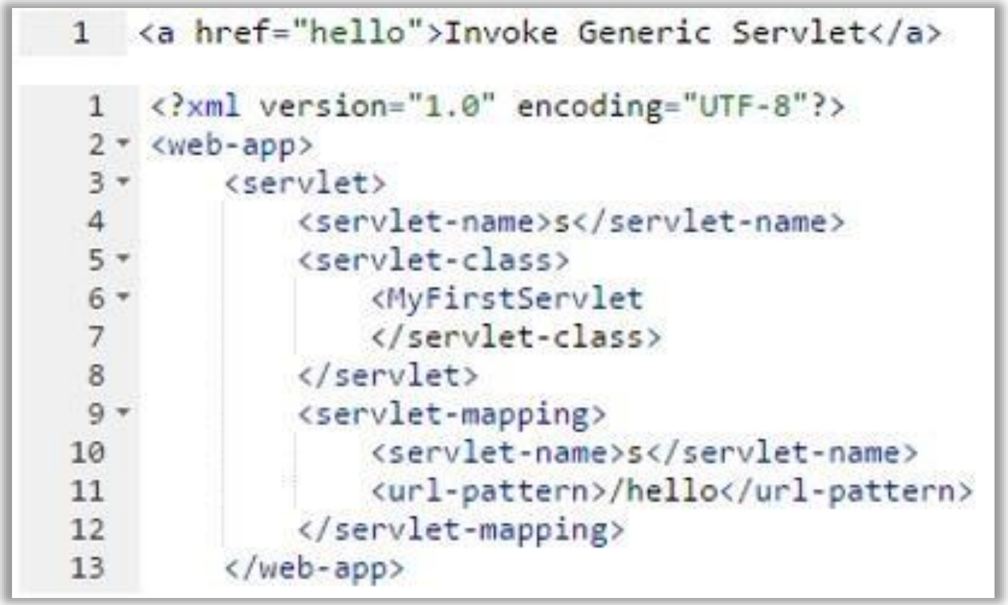
Servlet Exception

Servlet has the following two exceptions:

ServletException: Defines a servlet exception that a servlet throws while processing the client request.

UnavilableException: Defines a servlet exception that is thrown by a servlet, when a servlet is unavailable.

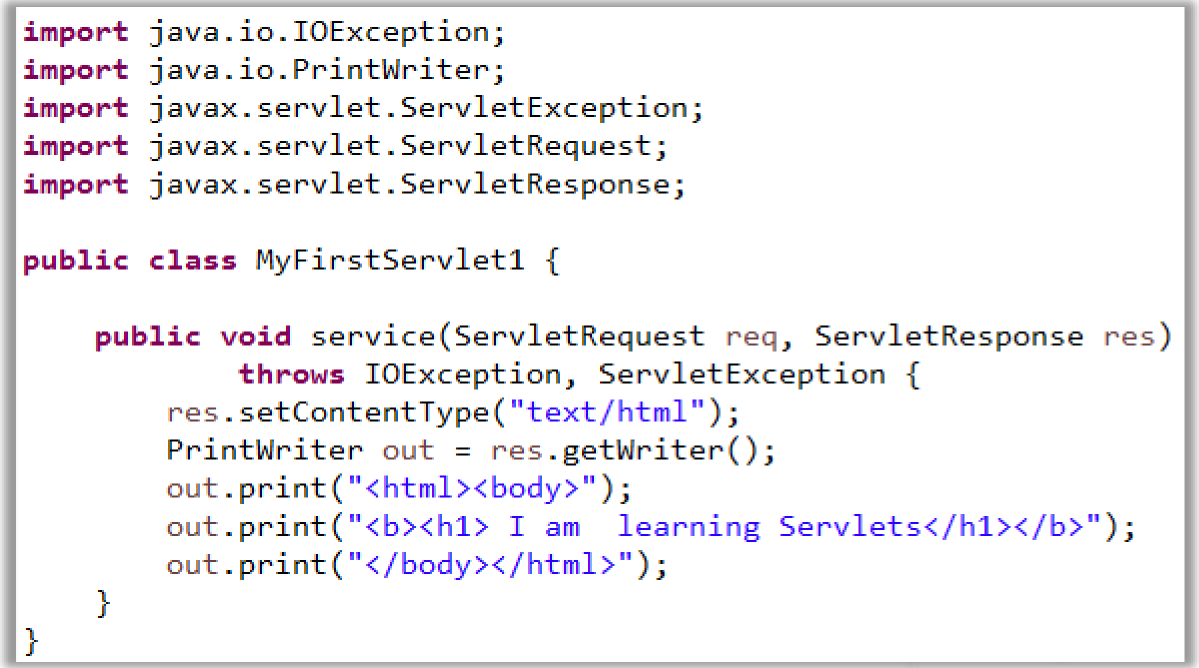
Servlet Program



```
<a href="hello">Invoke Generic Servlet</a>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <servlet>
        <servlet-name>s</servlet-name>
        <servlet-class>
            <MyFirstServlet
            </servlet-class>
        </servlet>
        <servlet-mapping>
            <servlet-name>s</servlet-name>
            <url-pattern>/hello</url-pattern>
        </servlet-mapping>
    </web-app>
```

GenericServlet Program



```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;

public class MyFirstServlet1 {

    public void service(ServletRequest req, ServletResponse res)
            throws IOException, ServletException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.print("<html><body>");
        out.print("<b><h1> I am  learning Servlets</h1></b>");
        out.print("</body></html>");
    }
}
```

Explanation of Generic Servlet Program

Index.html provides a hyperlink.

In Web.xml,

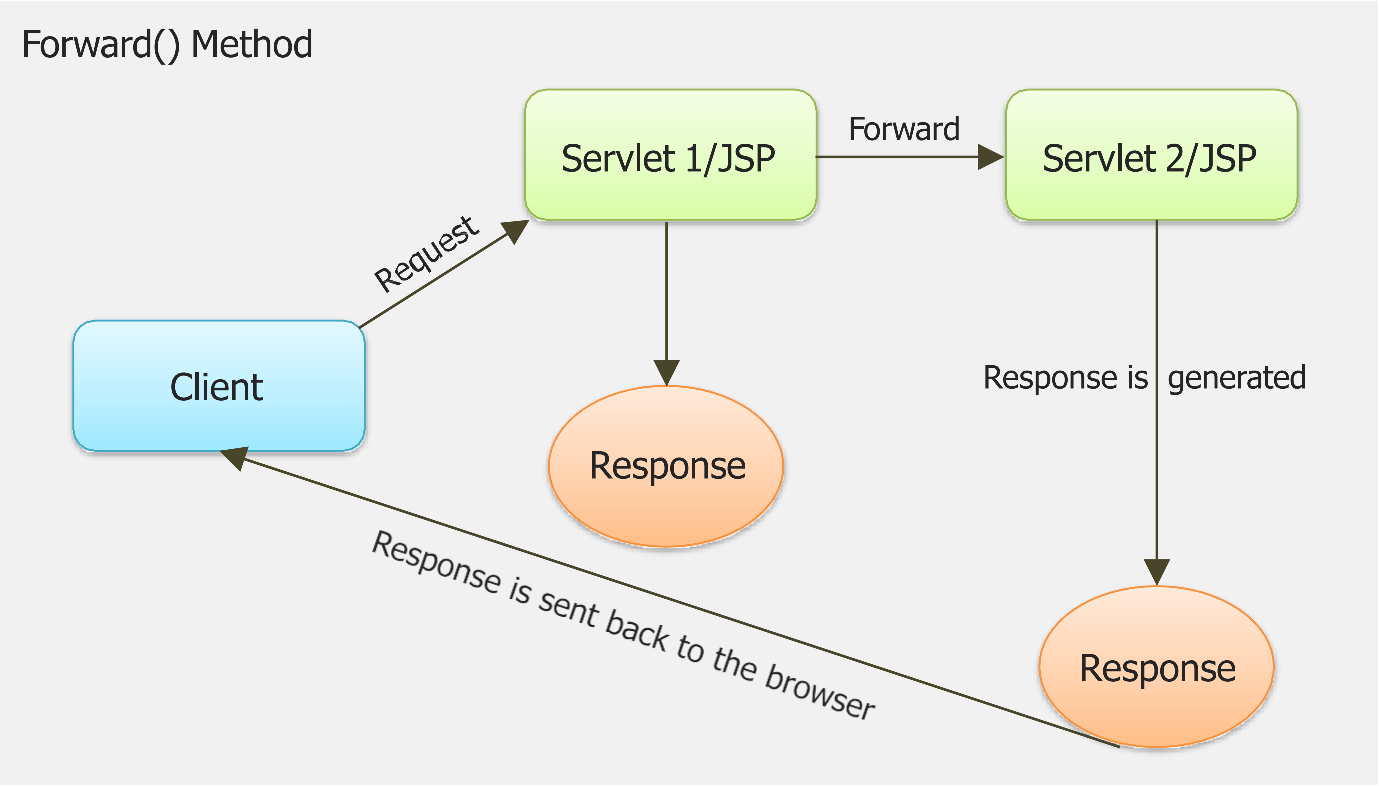
→ first you need to check <servlet-mapping>
→ <url-pattern> is mapped to a <Servlet-name>
→ Here when /hello is mapped to Servlet by name Name

Next look at <servlet> section. Here servlet name is mapped to servlet class. Here S is mapped to the class MyFirstServlet.

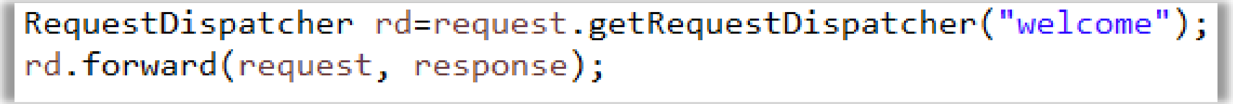
The program understands like this: URL Pattern → Servlet Name → Servlet Class...

Hence when /hello is given as part of URL, it executes MyFirstServlet.java class.

Forward and include of RequestDispatcher

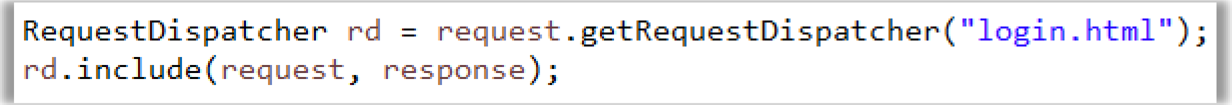


Request Dispatcher Example



```
RequestDispatcher rd=request.getRequestDispatcher("welcome");
rd.forward(request, response);
```

Dispatcher will load welcome servlet and forwards the control to it.

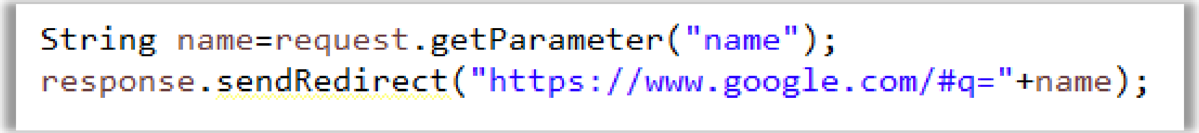


```
RequestDispatcher rd = request.getRequestDispatcher("login.html");
rd.include(request, response);
```

RequestDispatcher will load login.html and includes in the current page.

SendRedirect

SendRedirect() will redirect the control to a servlet/jsp/html.



```
String name=request.getParameter("name");
response.sendRedirect("https://www.google.com/#q="+name);
```

This code will take a search string and fires the query to google.com and displays the result on screen.

Session Tracking

It is required for a server to know the user and needs to maintain the state of the user as server will be addressing/serving multiple clients. For example, which user has logged in.

Knowing the user's information is called session tracking.

There are 4 ways for session tracking:

→ Cookies

→ URL Rewriting

→ Hidden form field

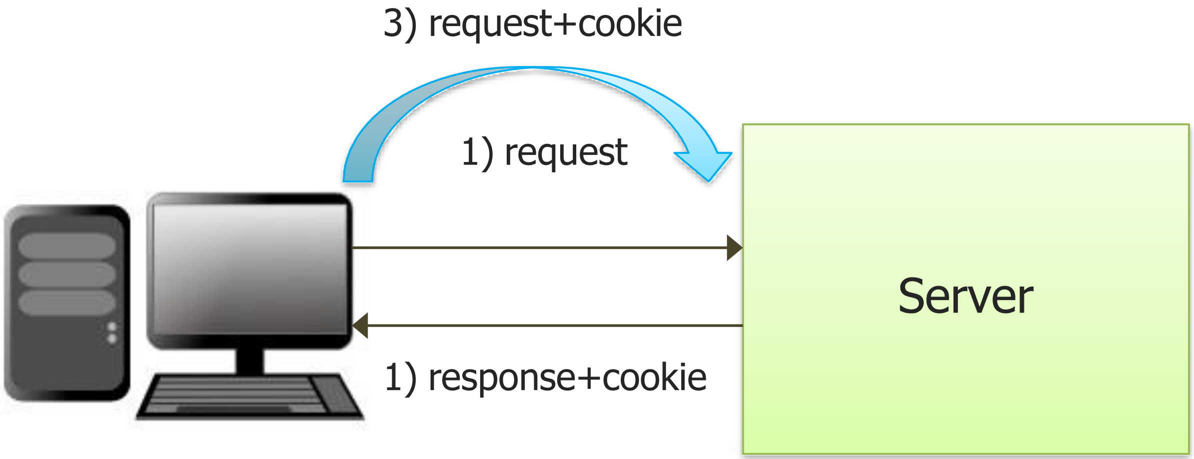
→ HttpSession object

Session Tacking – Cookies

“Cookies” is a small piece of information which is stored on client by the server.

Advantages: Maintained at client side.

Dis-advantages: Can’t be used if browser disables the cookies.



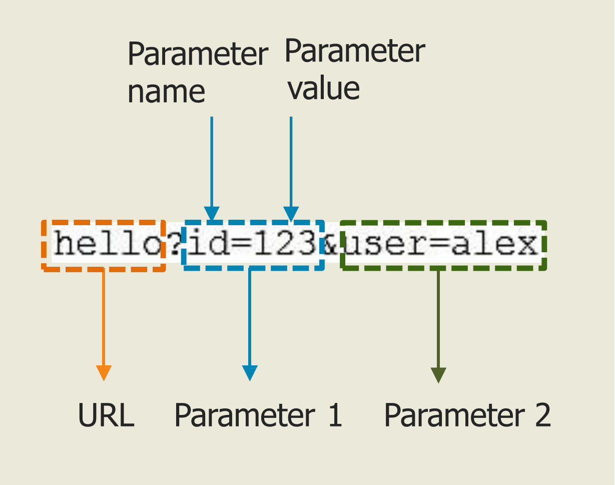
3) request+cookie

1) request

Server

1) response+cookie

Session Tacking – URL Rewrite



In URL Rewriting method, we append the user's data and call the next servlet. It is like

Http://localhost:8080/MyServlet?name=”james”

We are calling MyServlet by name James as the parameter for the field name.

Advantages: This method still works even when the cookies are disabled.

Dis-advantages: It can only text information.

Session Tracking: Hidden form Fields

This session tracking method uses Hidden keyword of HTML to send the information to the next servlet.

Advantages: This method will work even if the cookies are disabled.

Dis-advantages: To send the data to an another servlet, we need to have an extra form to be submitted.

Session Tracking – HttpSession

This session tracking method uses HttpSession object to store the state of the user.

```
HttpSession session=request.getSession();
```

```
session.setAttribute("name",”Alex”);
```

The above given code will gets the current session and stores in the session Object.

SetAttribute() is used to store the name Alex in the variable name.

In the entire session, we can use session.getAttribute(“name”), it will return the name Alex.

HttpSession is one of the best method to store the user's information.

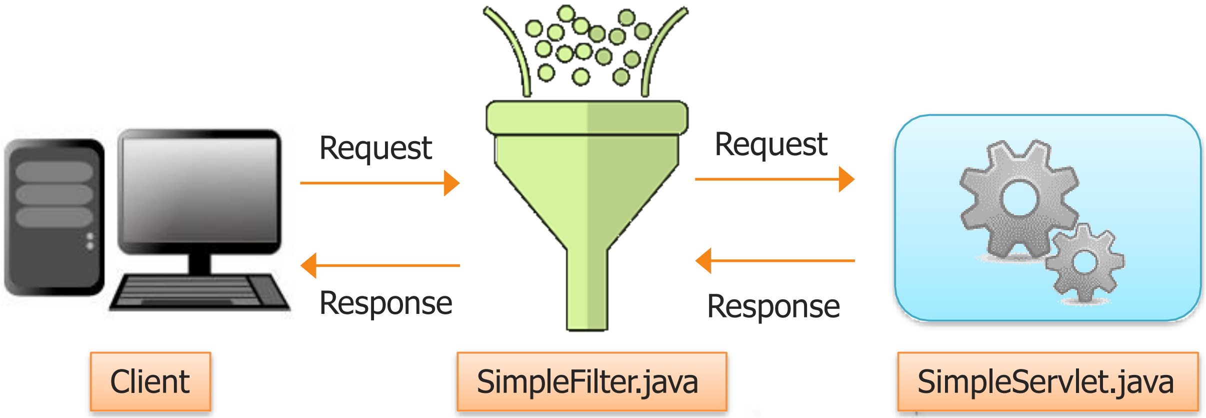
Filters

→ Filters are used to filter the data before actual processing begins.

→ Filters can be used for pre-processing and post-processing.

→ Filters are used for input validation, encryption/decryption, logging etc.

→ When a client sends request to the server with some data, control first goes to filters. Filters perform the necessary task in doFilter() and then forwards the control to another filter or a servlet.



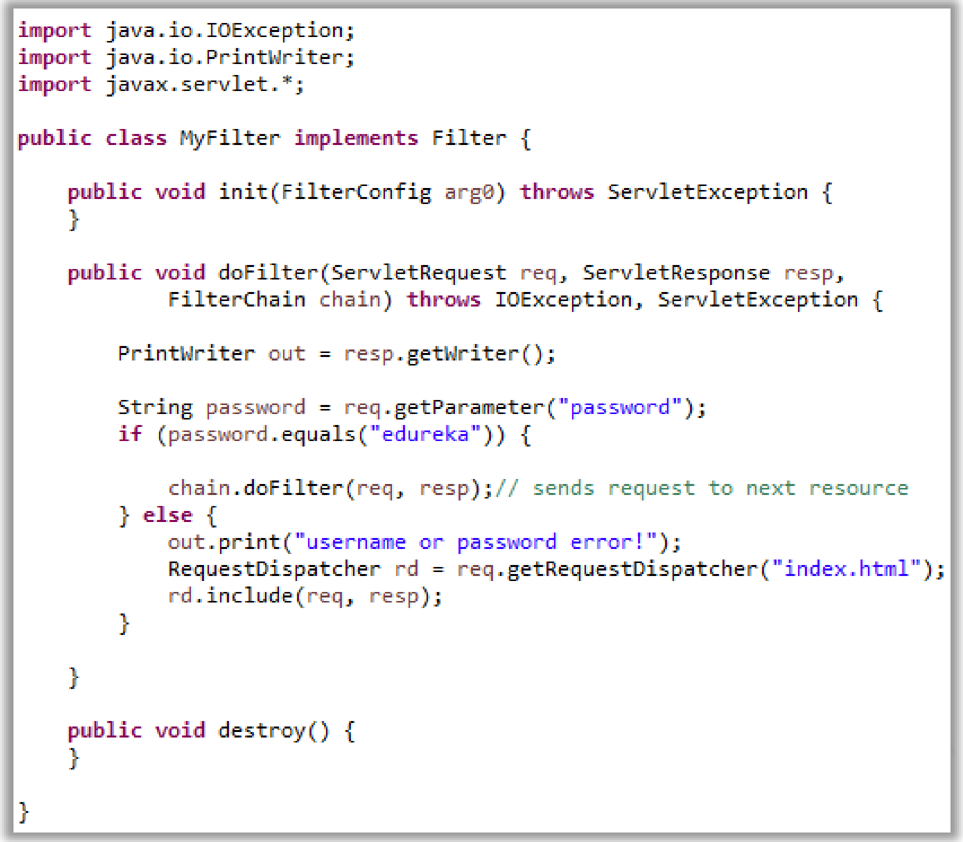
Features of Filters

→ Authentication and Authorization.

→ Logging information into the log files.

→ Compression and decompression before passing the data to the actual servlet.

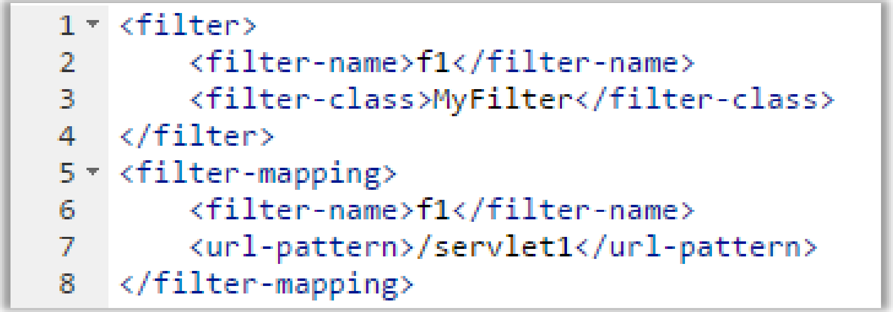
Example: Filters



```java
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;

public class MyFilter implements Filter {

    public void init(FilterConfig arg0) throws ServletException {
    }

    public void doFilter(ServletRequest req, ServletResponse resp,
            FilterChain chain) throws IOException, ServletException {

        PrintWriter out = resp.getWriter();

        String password = req.getParameter("password");
        if (password.equals("edureka")) {

            chain.doFilter(req, resp);// sends request to next resource
        } else {
            out.print("username or password error!");
            RequestDispatcher rd = req.getRequestDispatcher("index.html");
            rd.include(req, resp);
        }

    }

    public void destroy() {
    }

}
```

Code Explanation

Like servlet mapping in web.xml, filter mapping also has to be provided as given below:



```
<filter>
    <filter-name>f1</filter-name>
    <filter-class>MyFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>f1</filter-name>
    <url-pattern>/servlet1</url-pattern>
</filter-mapping>
```

Before the servlet gets executed, doFilter() of MyFilter.java will be executed.

In doFilter(), user password which is given by the user is checked with edureka. If the password is edureka then control is forwarded to the next resource else login page is displayed to the user to re-enter the valid user id and password.

QUESTIONS

Assignment

→ Develop login screen using HTML. When the user clicks on submit button validate the user id and password using filters.

If it is invalid id or password then display the login screen else display a servlet with given user id and password. If the user id and password is valid then store the user id as cookie on the client side.

Agenda for the Next Class

In the next module, you will be able to:

→ Understand JSP and its Architecture

→ Understand and Use JSP tags (Scripts, declarative, expression)

→ Use Implicit Objects

→ Implement JSP Directives

→ Use JSP and JDBC together

Survey

Your feedback is important to us, be it a compliment, a suggestion or a complaint. It helps us to make the course better!

Please spare few minutes to take the survey after the webinar.

Thank you!