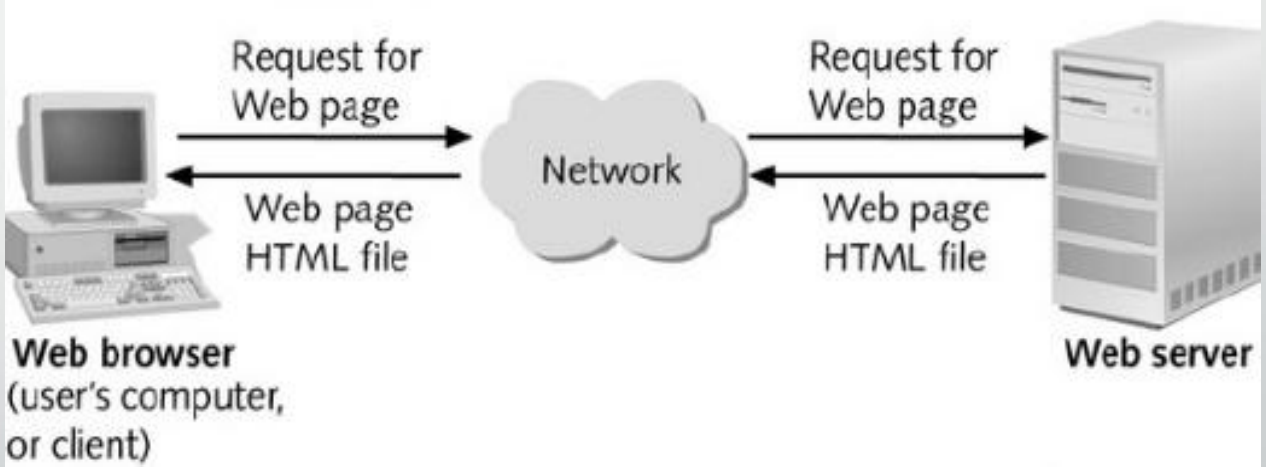
Web architecture

In a most simplified language a full fledged web applications consists of three basic components.

- Client (Browsers)
- Web Servers
- Data Servers



Web architecture

Client Side – GUI:

- The user interface is always rendered on a browser
- Technologies: HTML, CSS, Javascript

Web Server:

- web servers are computers that deliver web pages.
- web server has an IP address and most of the times a domain name.
- Any computer can be turned into web server.

Data Server:

- SQL Server
- Oracle Server etc.

Basic Web Application - Requirements:

- Local web server
- A browser
- Basic knowledge of HTML, CSS, Javascript, server side scripting language

HTTP- Hyper Text Transfer Protocol

- HTTP is an application-level protocol for distributed, collaborative, hypermedia information systems.
- HTTP is a request/response standard of a client and a server.
- Typically, an HTTP client initiates a request and server respond to that request

HTTP (Hypertext Transfer Protocol)

- Used for the vast majority of transactions on World Wide Web
- Every web page you visit is transmitted using HTTP prefix
  - http://
- Basically stateless, i.e., does not require the server to retain information or status about each user for the duration of multiple requests
- Relies on client, usually a web browser, to make a request and a server to send a response
  - HTTP client opens a socket connection to an HTTP server
  - Issues an HTTP request
  - Receives an HTTP response (stream of data)
- Connections for the protocol last long enough for one transaction, i.e., one or more request/response pairs

Why we need Servlets?

- Static Page: a page which is pre-made
- Dynamic Page: which builds on user request

- On server we have helper application (web container) where servlet will create a dynamic page for user

The Container

- The container, sometimes referred to as a servlet engine, is provided by a Web or an application server within which servlets run. Similar to simple Java programming, servlets are Java classes compiled to bytecode. These bytecodes are loaded dynamically into a Java technology-enabled Web server.
- A servlet container is often built into the server by default or sometimes provided as an add-on component via the server's native extension.
- The servlet classes generally interact with the server via an HTTP request/response mechanism implemented by the servlet engine.
- The primary function of the container is to contain servlet classes and manage their life cycle.
- Apart from supporting HTTP protocol, the servlet container also supports request/response based protocols, such as HTTPS, for a more secure network interaction.

The Component

- *Component* use the protocols and methods of the container to access other application components and services provided by the server.
- Servlet -> let=component
- Java Servlet is a platform-independent, container-based Web **component** used to generate dynamic content in a Web page.
- It is one of the stable technologies to share server-side resources in client-server programming.

Understanding the Java Servlet Life Cycle

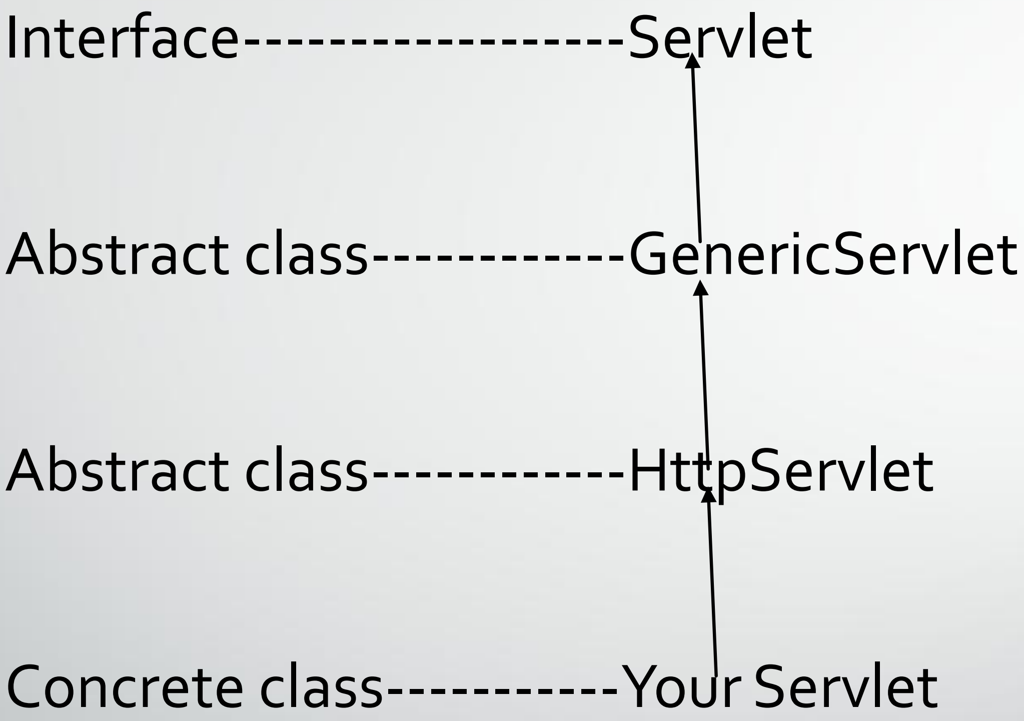
Typically, a servlet goes through the following sequence of events:

- A client makes an HTTP request to the Web server via a Web browser.
- The Web server delegates the request to the servlet container.
- Depending upon the configuration of the servlet, the container invokes the appropriate servlet class with the response/request object.
- The request object provides information about the user, parameters, and other details. Once these details are known, the servlet processes the data according to the program logic and then sends back to the client via a response object.
- The servlet container returns the control back to the Web server after request processing is finished.

Life Cycle of Servlet

1) Servlet class is loaded
2) Servlet instance is created
3) Init method is invoked
4) Service method is invoked
5) Destroy method is invoked

Classes/Interfaces/Packages



Interface------------------Servlet

Abstract class------------GenericServlet

Abstract class------------HttpServlet

Concrete class-----------Your Servlet

Package   javax.servlet.*

HttpServlet class

- Methods:
  - Protected void service (HttpServletRequest req, HttpServletResponse res)
    - Receives the request from the service method and dispatches the request to the doXXX() method depending on the incoming http request type.
  - Protected void doGet (HttpServletRequest req, HttpServletResponse res)
    - Handles the GET request
  - Protected void doPost (HttpServletRequest req, HttpServletResponse res)
    - Handles the POST request

Http Requests

- **GET:** Ask to get resource at the requested URL.
- **POST:** Asks the server to accept the body info attached. It is like GET request with extra info sent with the request.

Adding 2 numbers using servlets

```
<html>
    <body>
        <form action = “Add” method=“post”>
            Enter 1st number: <input type=“text” name=“t1>
            <br>
            Enter 2nd number: <input type=“text” name=“t2”>
            <br>
            <input type=“submit” value=“Add”>
        </form>
    </body>
</html>
```

```
Import javax.servlet.*;
Import java.io.*;

@WebServlet(urlPatterns = {"/Add"})
Public class Add extends HttpServlet
{
        public void service(HttpServletRequest req, HttpServletResponse
res)
{
        int I = Integer.parseInt(req.getParameter(“t1”);
        Int j = Integer.parseInt(req.getParameter(“t2”);
        Int k = j+j;

        PrintWriter out = res.getWriter();
        Out.println(k);
}
}
```

End.