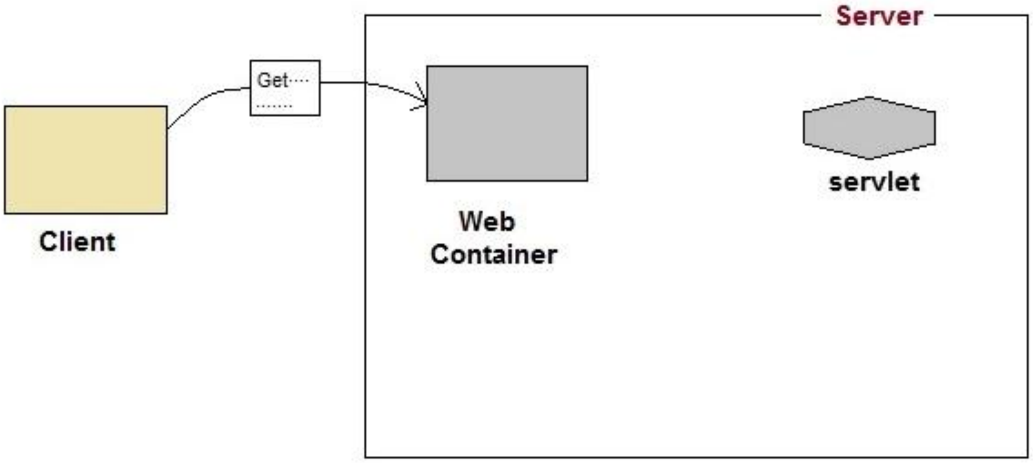
How Servlet internally works?

The web container is responsible to handle the request. Let's see how it handles the request.

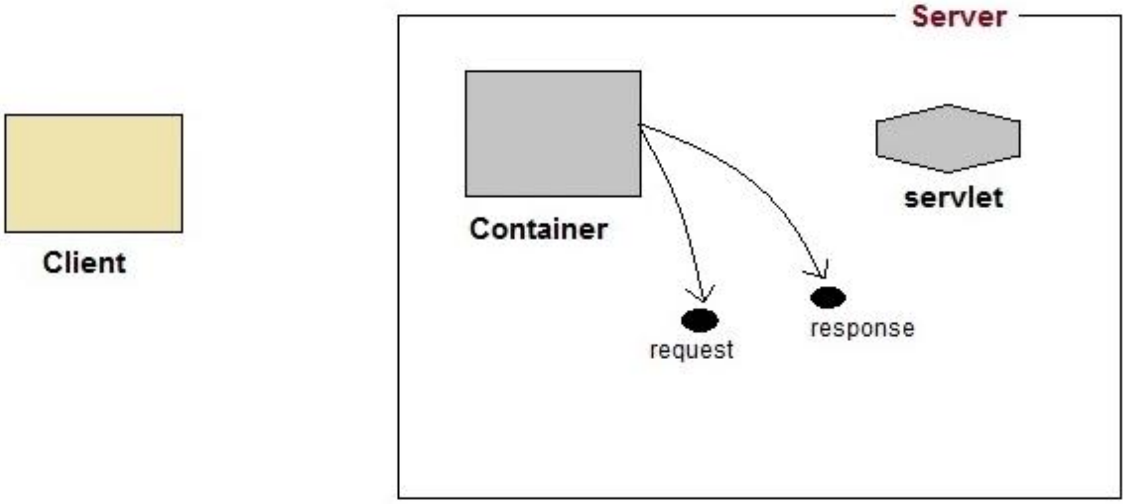
- maps the request with the servlet in the web.xml file.
- creates request and response objects for this request
- calls the service method on the thread
- The public service method internally calls the protected service method
- The protected service method calls the doGet method depending on the type of request.
- The doGet method generates the response and it is passed to the client.
- After sending the response, the web container deletes the request and response objects. The thread is contained in the thread pool or deleted depends on the server implementation.

1. When User sends request for a servlet by clicking a link that has URL to a servlets.

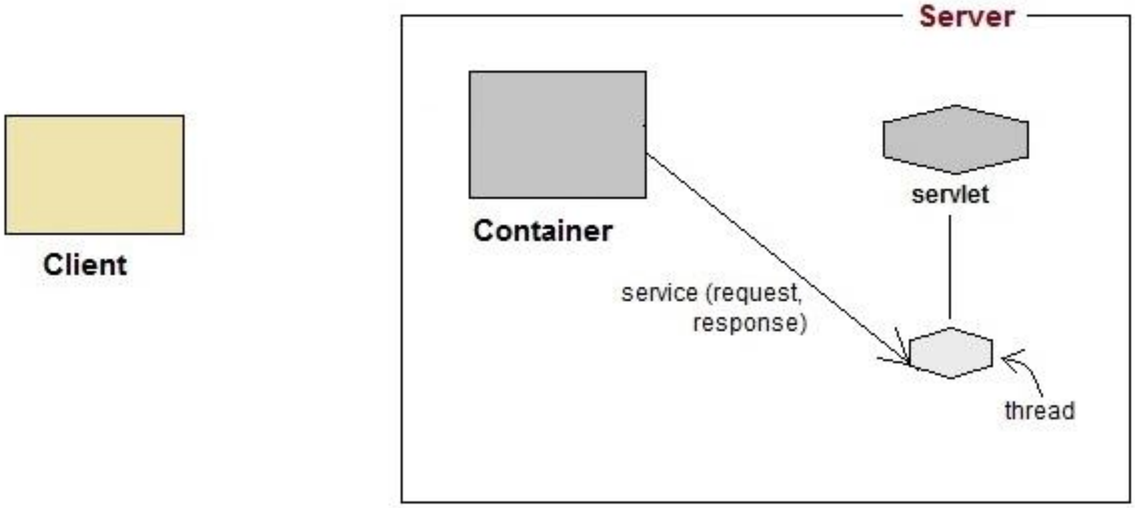


2. The container finds the servlet using deployment descriptor and creates two objects :

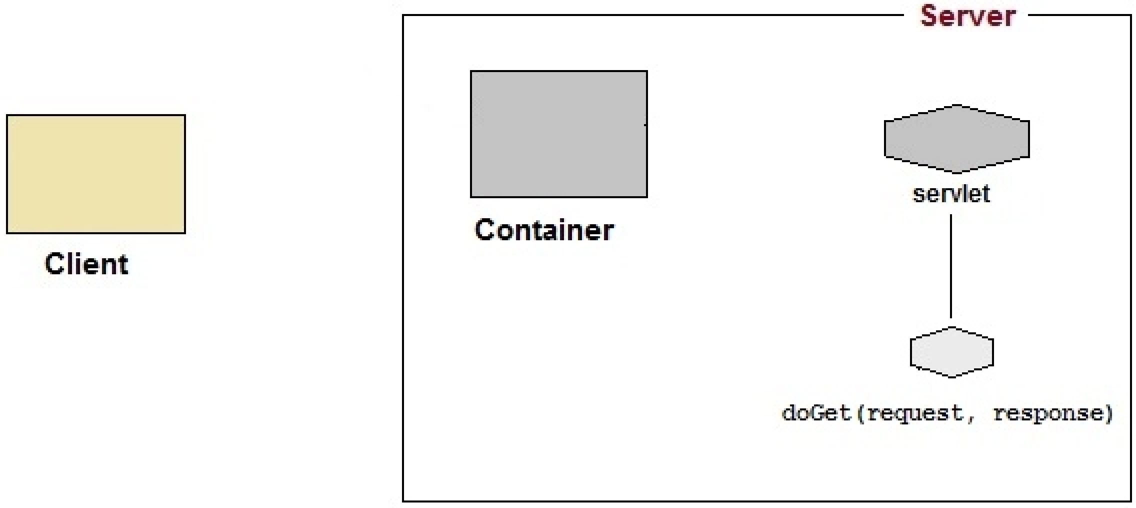
a. HttpServletRequest
b. HttpServletResponse



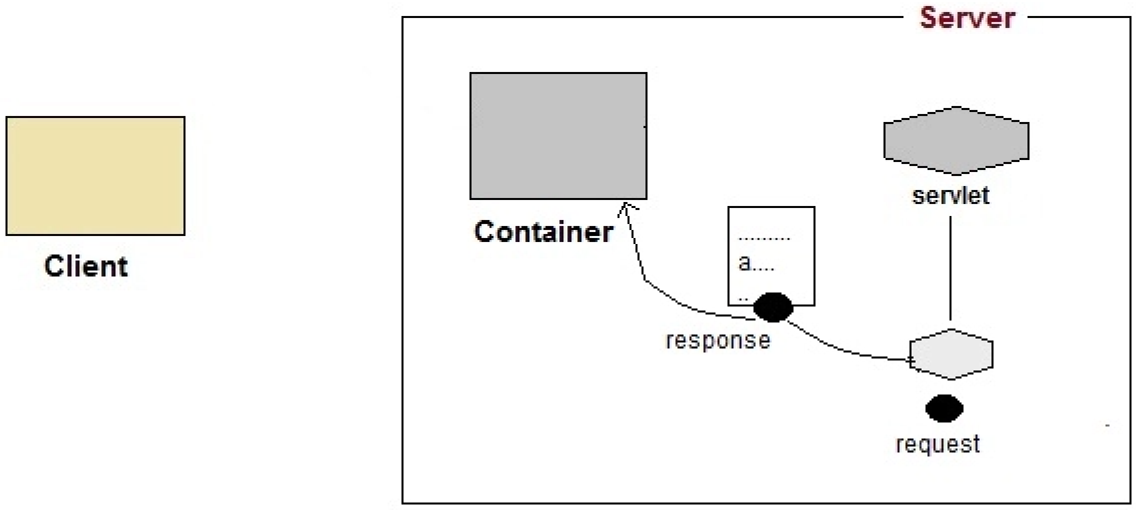
3. Then the container creates or allocates a thread for that request and calls the Servlet's service() method and passes the request, response objects as arguments.



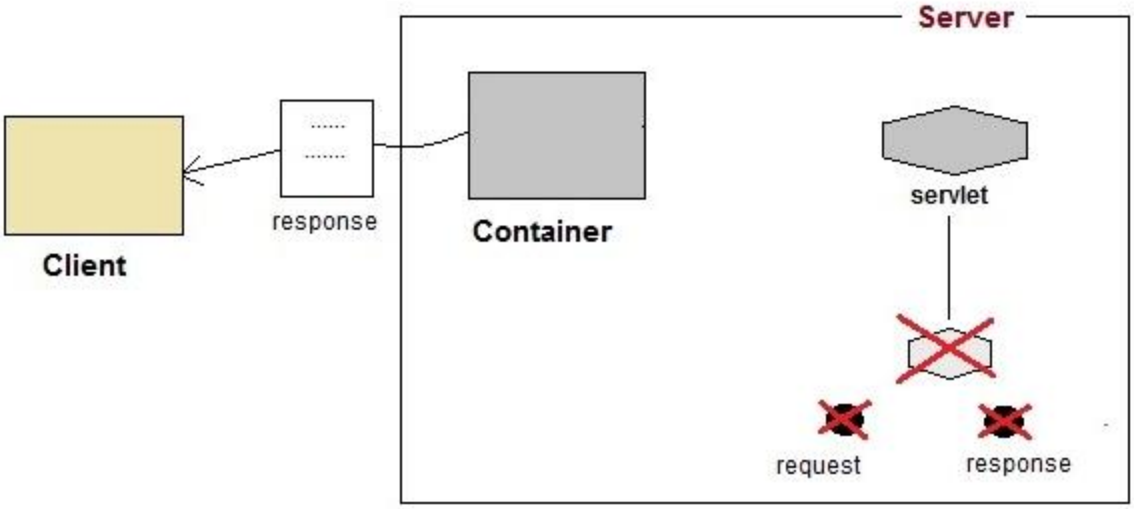
4. The service() method, then decides which servlet method, doGet() or doPost() to call, based on HTTP Request Method(Get, Post etc) sent by the client. Suppose the client sent an HTTP GET request, so the service() will call Servlet's doGet() method.



5. Then the Servlet uses response object to write the response back to the client.



6. After the service() method is completed the thread dies. And the request and response objects are ready for garbage collection.


Server
Client
response
Container
servlet
request
response

HTML + Servlet Example-

How to pass the data from HTML page to Servlet Page.

Register.jsp

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="test" method="post">
Username<input type="text" name="username"/>
Password<input type="password"name="password"/>

<input type="submit" value="submit">
</form>
</body>
</html>
```

RegistrationServlet.java

```
package com.servlet;


import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
```

```
/**
 * Servlet implementation class RegistrationServlet
 */
@WebServlet("/test")
public class RegistrationServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void service(HttpServletRequest request, HttpServletResponse
response)
                        throws ServletException, IOException {

                String username = request.getParameter("username");
                System.out.println("username>>" + username);
                String password = request.getParameter("password");
                System.out.println("password>>" + password);
        }

}
```

Note- Form action name and WebServlet(“”) name should be same.

http://localhost:8080/ServletExample/Register.jsp

then username and password displayed on screen.