

Java Servlets 3.0

Lesson 03: Introduction to Servlets API
and Ease of Development through
Annotations



Lesson Objectives

In this lesson, we will learn:

- Introduction to Servlet
- Role of Servlets in Web application design
- Advantages of Servlets
- Basic Servlet Architecture : Servlet Container
- Servlet Lifecycle
- Ease of Developing Servlets through Annotations
- Retrieving Information from HTML Page





What are Servlets?

Servlets are Java programs that extend the functionality of a Web server and capable of generating a dynamic response to a particular request using the HTTP Request / Response paradigm

Servlets are not tied to a specific client-server protocol but they are most commonly used with HTTP and the word “Servlet” is often used in the meaning of “HTTP Servlet”

- It is available and runs on all major web and application servers
- It is platform and server independent



What Can Servlets Do?

Servlets can do the following functions:

- Dynamically build and return an HTML file based on nature of client request
- Process user input passed in an HTML form and return an appropriate response
- Provide user authentication and other security mechanisms
- Interact with server resources such as databases, other applications and network files to return useful information to the client
- Automatically attach web page design elements such as headers or footers, to all pages returned by server
- Forward requests from one server to another for load balancing purpose
- Manage state information on top of the stateless HTTP



Advantages of Servlets

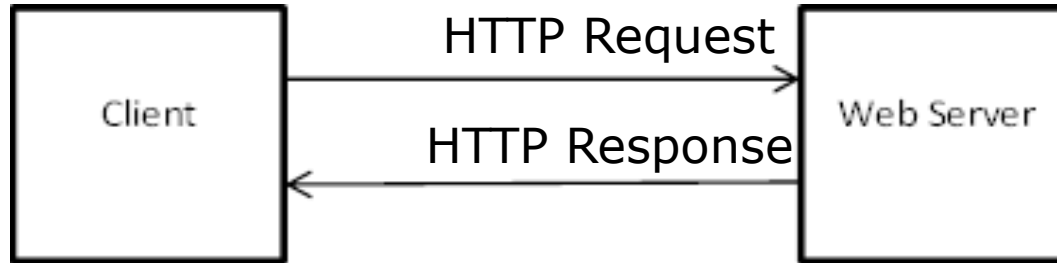
Servlets provide the following advantages:

- Crash Resistance
- Cross-Platform
- Cross-Server
- Durable
- Dynamically Loadable across the Network
- Extensible
- Multithreaded
- Protocol Independent
- Secure



What is a Web Server?

- To know what is a Servlet container we need to know what is a Web Server first



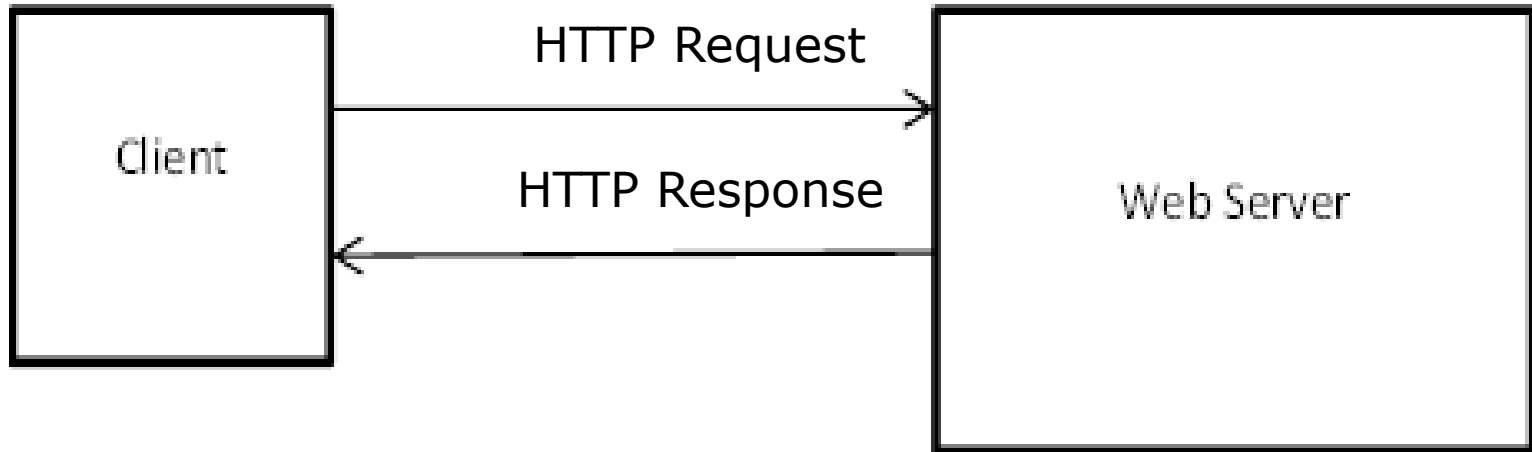
A web server uses HTTP protocol to transfer data. In a simple situation, a user types in a URL (e.g. www.servletdemos.com/static.html) in browser (a client), and get a web page to read. So what the server does is sending a web page to the client. The transformation is in HTTP protocol which specifies the format of request and response message.



Servlet Container / Servlet Engine / Web Container

What is a Servlet Container / Servlet Engine / Web Container

- Client can request only static web page from Server. If client wants to read Web page based upon input (that requires processing), basic idea of servlet container is to dynamically generate the Web page on the server side
- Servlet container is essentially a part of a web server that interacts with the servlets. Servlet Container is container for servlets





Servlet Container / Servlet Engine / Web Container

Advantages of Servlet Container

- Providing communication support between Web Components and Web Server
- Life cycle support for Web Components
- Networking support
- Enabling Web Security
- Multi threading support for Web Components



3.4: Basic Servlet Architecture : Servlet API

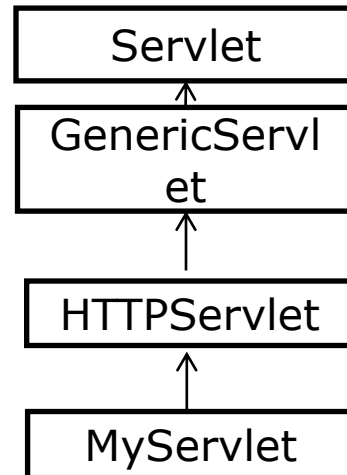
Servlet API

A servlet is an instance of a class which implements the `javax.servlet.Servlet` interface.

Servlet is web component inside Servlet Container

Most servlets extend one of the standard implementations of that interface, namely `javax.servlet.GenericServlet` or `javax.servlet.http.HttpServlet`.

In Servlet 3.0, there is `javax.servlet.annotation` package for creating servlets via annotations





Servlet Interface Life Cycle Methods

`init():`

- It is executed once when the servlet is first loaded.

`service():`

- It is called in a new thread by server for each request.

`destroy():`

- It is called when server deletes servlet instance.

These lifecycle methods are implemented in `GenericServlet` class.



3.4: Basic Servlet Architecture : Request Processing

Steps to process a Request

Web server receives HTTP request

Web server forwards the request to servlet container

The servlet is dynamically retrieved and loaded into the address space of the container, if it is not in the container

The container invokes the `init()` method of the servlet for initialization(invoked once when the servlet is loaded first time)

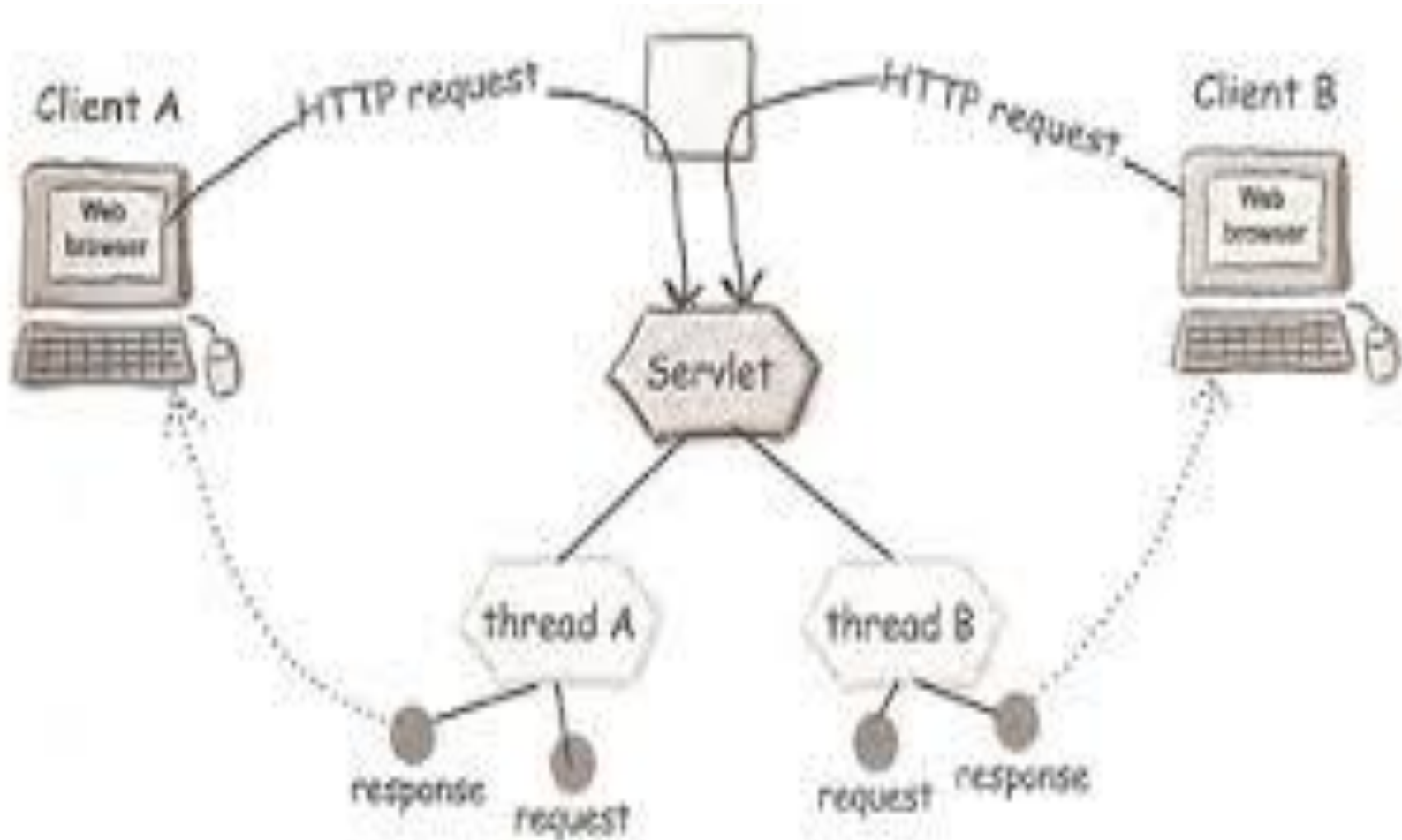
The container invokes the `service()` method of the servlet to process the HTTP request, i.e., read data in the request and formulate a response. The servlet remains in the container's address space and can process other HTTP requests

Web server return the dynamically generated results to the client



3.4: Basic Servlet Architecture : Request Processing

How Container handles a request



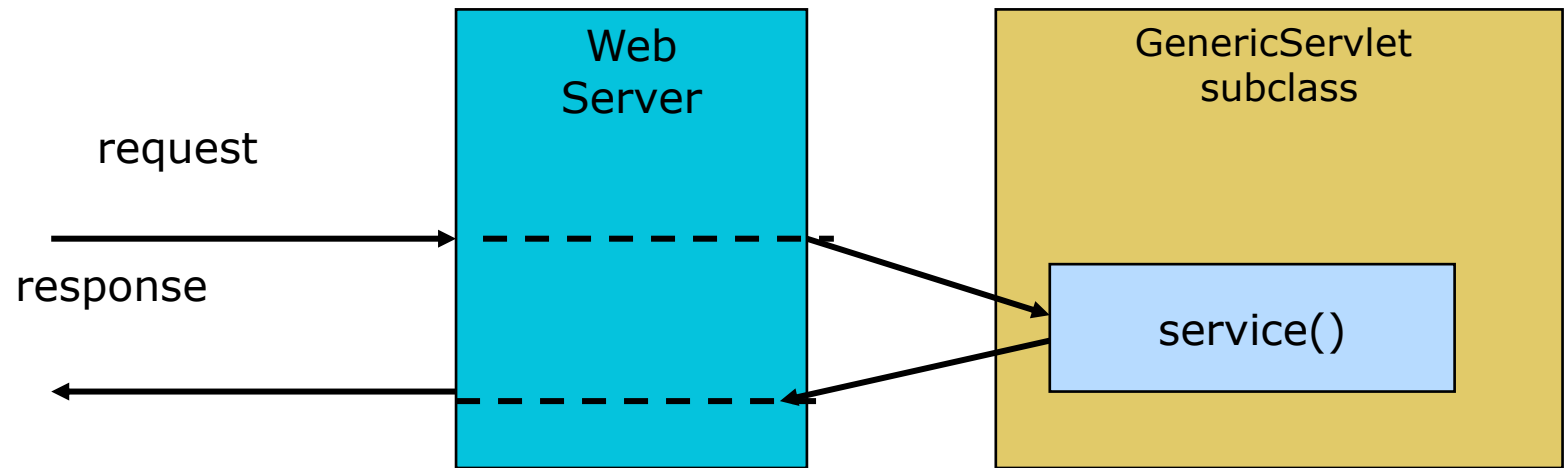


3.4: Basic Servlet Architecture : Hierarchy

GenericServlet class

javax.servlet.GenericServlet

- It is protocol independent
- It makes writing servlets easier
 - It provides simple versions of `init()` and `destroy()` and of the methods in the `ServletConfig` interface.
 - It also implements the `log` method, declared in the `ServletContext` interface.
- To write a generic servlet, override the abstract `service()`.



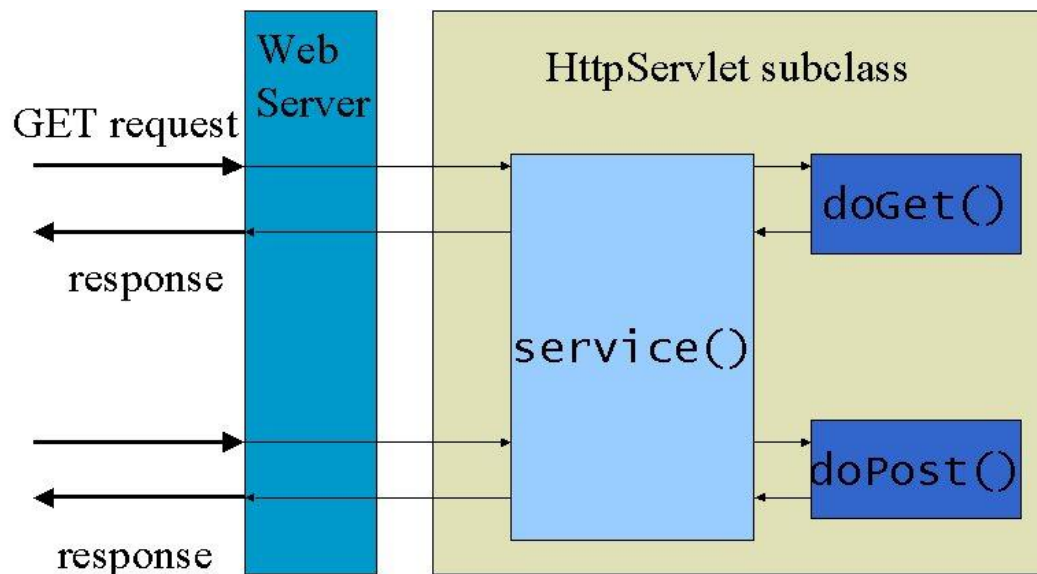


3.4: Basic Servlet Architecture : Hierarchy

HttpServlet class

`javax.servlet.http.HttpServlet:`

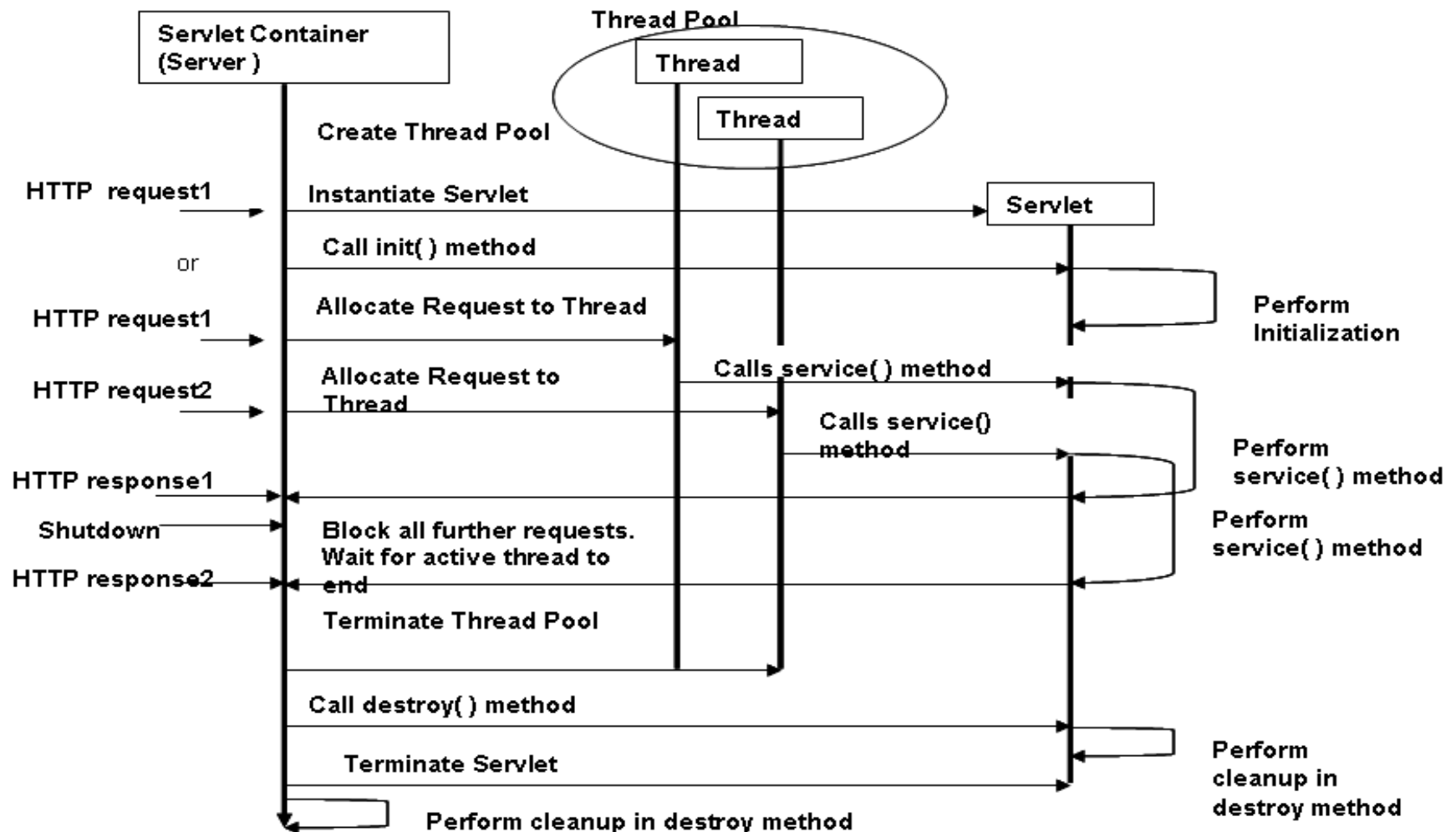
- It has a built-in HTTP protocol support.
- Its subclass must override at least one of the following methods: `doGet()`, `doPost()`, `doHead()`, `doTrace()`, `doPut()`, `doDelete()`
- One must not override the `service()` method, since it dispatches a request to the different `doXXX()` methods for different HTTP requests.





3.5 : Servlet lifecycle

Servlet Life Cycle





Annotations and their Need

Annotations can be described as metadata. These are metadata for the code written; and do not contain any business logic

They specify a standard way of defining metadata in code

Annotations are tightly coupled with the code

Application development becomes easy due to annotations



Differences between Servlets 2.0 and Servlets 3.0

Servlet 2.0	Servlets 3.0
Creation of Servlets and other components via XML mappings	Creation of Servlets and other components via annotations
No such feature	Pluggability support for 3 rd party frameworks
No such feature	Asynchronous processing of Servlets



URL Mapping of Servlets and Annotations

@WebServlet Annotation:

`javax.servlet.annotation.WebServlet` is a class-level annotation that affirms the annotated class as a servlet and holds metadata about the declared servlet

The `urlMappings` attribute is a mandatory attribute of `@WebServlet` that specifies the URL pattern that invokes this servlet

Here is the much simplified version written to the Servlet 3.0 API.

As `MyServlet` is annotated as a servlet using the `@WebServlet` annotation, it gets initialized during the startup of the web container. Note that the deployment descriptor is optional in this case.

```
@WebServlet(name = "Basic ", urlPatterns={"/MyApp"}, loadOnStartup=1)
public class MyServlet extends HttpServlet{
    public void doGet(HttpServletRequest req, HttpServletResponse res){        ....    }}
```



3.6: Ease of developing Servlets through Annotations

Writing First Servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
@WebServlet("/HelloWorld")
public class HelloWorld extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<BIG>Hello World</BIG>");
        out.println("</BODY></HTML>");
    }
}
```

Servlets are not part of standard SDK, they are part of j2ee

Use of @WebServlet annotation

Servlets normally extend HttpServlet

Setting the response type

Getting PrintWriter object to send response to client

Output to the client



3.7: Retrieving information from HTML Page

Retrieving Information

Many a times, information about the environment in which a web application is running needs to be known.

Sometimes information about the server that is executing servlets or which client is sending the request needs to be known.

Sometimes information regarding the requests that the application is handling is required to be known

We shall now see a number of methods that provide this information to servlets

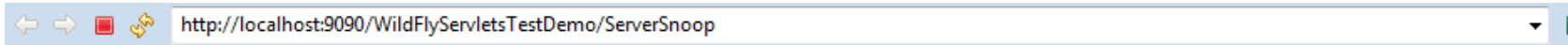


Getting Information about the Server

There are four methods that a servlet can use to learn about its server:

These are as follows:

- `public String ServletRequest.getServerName()`
- `public String ServletRequest.getServerPort()`
- `public String ServletContext.getServerInfo()`
- `public String ServletRequest.getAttribute(String name)`



```
req.getServerName(): localhost
req.getServerPort(): 9090
getServletContext().getServerInfo(): WildFly 8.1.0.Final - 1.0.15.Final
getServerInfo() name: WildFly 8.1.0.Final - 1.0.15.Final
getServletContext().getAttribute("attribute"): null
```



Getting Info about client m/c and user

Getting Information about client machine:

- `public String ServletRequest.getRemoteAddr()`: It retrieves the IP address of the client machine.
- `public String ServletRequest.getRemoteHost()`: It retrieves the hostname of the client machine.

Getting Information about User:

- `public String HttpServletRequest.getRemoteUser()`: It returns the login of the user, if the user has been authenticated, or null if not.

Lab: JSP Scripting Elements



Lab 1.1



Summary



In this lesson, we have learnt:

- Role of Servlets in web application design
- Basic Servlet Architecture
- Servlet Lifecycle
- Elements of a Web Application
- Developing Servlets
- Initializing Servlets
- Getting Information about the Server, Client and User



Review Questions



Question 1: Which Method in a HttpServlet is not recommended to be overridden?

- Option 1: init
- Option 2: destroy
- Option 3: service
- Option 4: doGet
- Option 5: doPost



Question 2: What is the superclass of HttpServlet?

- Option 1: GenericServlet
- Option 2: Servlet
- Option 3: SuperHttpServlet

Review Questions



Question 3: Servlets are:

- Option 1: Server-side components
- Option 2: Client-side components
- Option 3: Neither Client-side or Server-side components
- Option 4: Data tier components

Question 4: When a servlet receives an HTTP GET Request for the first time, which of these methods will be called?
(Assume the servlet is not preloaded)

- Option 1: init
- Option 2: doPost
- Option 3: processRequest
- Option 4: doGet
- Option 5: service





Review Questions

Question 5: What is the significance of using annotation `@WebServlet("/HelloWorld")` above `public class HelloWorld` extends `HttpServlet` `{}`

- Option 1: Instruct container to create Servlet Hello with URL Pattern `"/HelloWorld"`
- Option 2: Instruct container to create Servlet HelloWorld with URL Pattern `"/Hello"`
- Option 3: Instruct container to create Servlet HelloWorld with URL Pattern `"/HelloWorld"`
- Option 4: No significanc

