Servlets and JSP

Review-

- Overview of Web Application
- HTTP Methods and Responses

Outline

- Overview of Web Application
- HTTP Methods and Responses
- Lifecycle of Web Servlets
- Writing Servlet programs with Servlet APIs
- Reading and Processing Forms
- Handling GET/POST Requests
- Database connectivity through servlets
- Cookies and Sessions

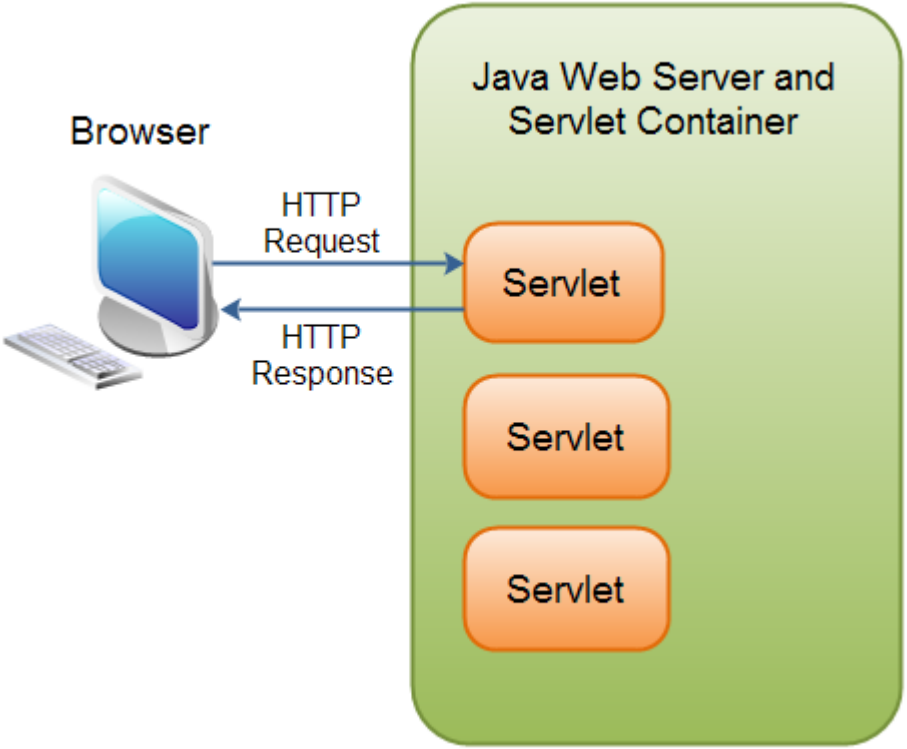
Servlet

- What is Servlet
- Features of Servlet
- Advantages of Servlet
- What is JSP
- Advantages of JSP

What is Servlet

- Servlet stands for server component, and it is a Java class program that runs on servers and extends servers' capabilities.
- It is a web component deployed on the server to create dynamic web pages. It is called a web component because it operates within a Java Enterprise Edition environment to extend the capabilities of a web server.
- Servlet can take any kind of request and process to respond in the form of an HTML page.
- They follow the Java Servlet API specification and are a key part of Java EE(Enterprise Edition).

Servlet Contd...



Features of Servlet

- Used to create web applications
- Acts as a communicator between client and server
- An application programming interface (API) that provides many interfaces and classes, including documentations
- Can generate a response to any kind of incoming request from the client
- A web container (also called a Servlet container) required to deploy and run a Servlet

Life Cycle of Servlet

Managed by Servlet Container:

- Servlet Container Role: The servlet container is responsible for managing the entire life cycle of a servlet. It ensures that servlets are loaded, initialized, can handle requests, and are eventually destroyed when no longer needed.

- javax.servlet.Servlet Interface: This interface is used by the servlet container to understand and interact with the servlet object. It provides methods that define the servlet's life cycle, allowing the container to manage its creation, initialization, service handling, and destruction.

Understanding the Servlet Life Cycle

Importance Before Creation: Before creating a servlet object, it's crucial to understand its life cycle. This knowledge helps developers ensure that the servlet behaves correctly within the container environment.

Management Insight: Understanding the life cycle provides insight into how the servlet container manages the servlet object, ensuring proper loading, initialization, request handling, and cleanup processes are followed.

Stages of the Servlet Life Cycle:

Loading a Servlet:

- The servlet class is loaded by the servlet container.

Initializing the Servlet:

- The servlet instance is created and initialized.

Request Handling:

- The servlet handles client requests.

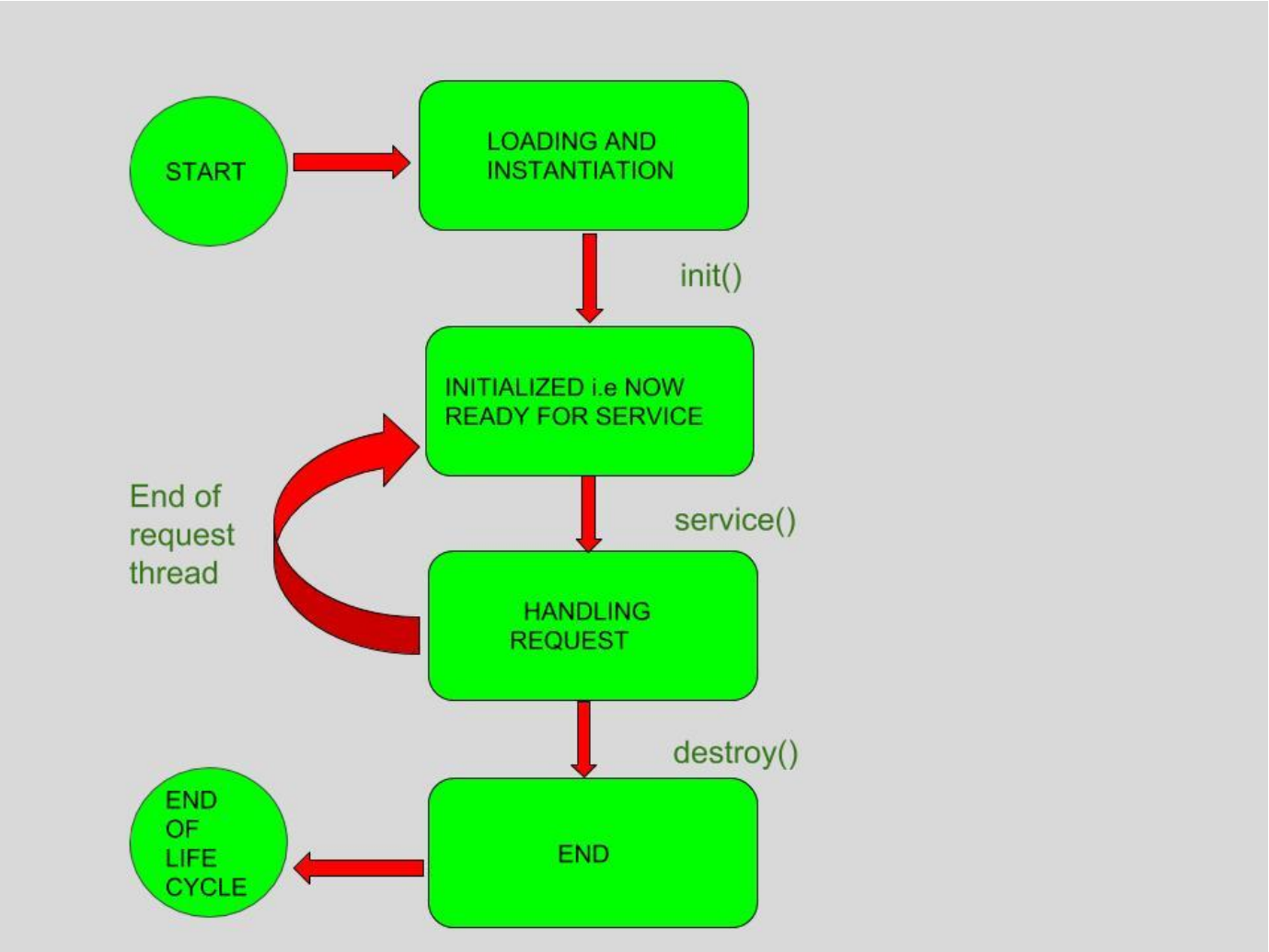
Destroying the Servlet:

- The servlet instance is destroyed.

Life Cycle Overview

- Initialization
  - Servlet container loads and initializes the servlet when the first request is received.
  - init() method is called during initialization.
- Handling Requests
  - Servlet handles multiple requests concurrently through multithreading.
  - service() method is invoked for each request, delegating to doGet() or doPost() based on the request type.
- Termina ion
  - Servlet container calls destroy() method when shutting down or reloading the web application.
  - Cleanup operations are performed in the destory() method.

Lifecycle Contd..



Methods in Servlet Life Cycle

- init(ServletConfig config) Method:
  - Initializes the servlet with the configuration data.
  - Called once during the servlet's life cycle.
- service(ServletRequest req, ServletResponse res) Method:
  - Invoked for each request.
  - Determines the type of request(GET, POST, etc) and delegates to the appropriate method(doGet(), doPost())
- destory() Method:
  - Called once during the servlet's life cycle.
  - Performs cleanup operations before the servlet is unloaded.

Writing Servlet Programs with Servlet APIs

- Import Statements:
  - Import necessary classes from the javax.servlet package.
- Extending HttpServlet:
  - Servlet classes extend HttpServlet to handle HTTP requests.
- Overriding Methods:
  - Override doGet() or doPost() methods to handle specific request types.

Advantages Of Servlet

- Better performance: Because Servlet creates a thread for every client request and not a long process, that results in better performance.
- Robust: Since JVM manages Servlet, the chances of memory leaks are rarest.
- Competent: Memory is persistent so, the invocation is very efficient.
- Elegant: The code is clean, object-oriented, modular, and intuitive.
- Strong integration: Integrate well with the server to translate file paths, conduct logging, verify authorisation, and MIME type mapping.
- Secure: Support safe programming because it uses java language.

Disadvantages of Servlet

- Time-consuming: Modification in the Servlet file is time-consuming due to reloading, recompiling, and restarting the server.
- Bulky: Writing HTML code in Servlet programming is challenging, and it makes Servlet looks bulky.
- Difficult to understand: Writing complex business logic makes the application hard to comprehend.
- Difficult to modify: Because HTML code and Java code are mixed, changes made in one code might alter the other, making modification difficult.
- Requires broader knowledge: Servlet may not be suitable for non-java developers since it requires a profound knowledge of JSP Servlet.
- Dependent: A java runtime environment requires on the server to run Servlet.

What is JSP

- Java Server Pages or JSP is a server-side technology. It is an extended version of Servlet technology developed by Sun Microsystems.
- In JSP, you write JAVA code inside HTML using JSP tags to create web pages.
- It gives more functionality than Servlet.
- A compatible web server with a Servlet container is necessary to execute JSP.

Features of JSP

- Web-based technology
- Used to create web applications
- Helps to create dynamic and platform-independent web pages
- Extended version of Servlet and allows more functionality
- A component of Java EE, and used for the simplest to the most complex applications
- Interacts smoothly with Java Servlets in a web container
- Allows you to insert Java inside the HTML

The Life Cycle Of JSP

- Translation of JSP to Servlet: Here, in the first step, the .jsp file translates to _jsp.java.
- Compilation of JSP page: The compilation of the translated Java Servlet file (_jsp.java) occurs in a Servlet class file (_jsp.class).
- Classloading: Now, the compiled Servlet class loads into the container using classloader.
- Instantiation: In this step, the web container generates an instance of that Servlet class.
- Initialisation: The container invokes the _jspinit() method. _jspinit() method is invoked only once in a life cycle after Servlet instance generation.
- Request Processing: Now, the container invokes _jspservice() method to process the request. You cannot override this method.
- Destroy: _jspdestroy() method is used to destroy the Servlet instance from use. The container invokes the _jspDestroy() method to perform any required clean up. _jspdestroy() method is invoked only once. You can override this method.

Advantages of JSP

- Easy maintenance: JSP is easy to maintain since separating business logic with presentation logic is simple.
- Servlet extension: Being the extension of Servlet technology, it can use all the features of Servlet.
- Require less coding: It allows custom tags that reduces the code and makes coding flexible.
- Fast development: There is no requirement to recompile and redeploy the project while modifying a JSP page.
- Easy integration: Supports Java API's which can be easily integrated with the HTML code.
- Easy modification: Changes can be done through the business logic page rather than changing every page that makes modification easy.

Disadvantages of JSP

- Hard to debug: Because JSP pages initially translate into Servlets before compilation, debugging or tracing errors is difficult.
- Time-consuming: Because JSP pages are compiled on the server, they take more disc space and time to store JSP pages.
- Fewer features: Since JSP output is HTML, it is not abundant in features

Difference between Servlets and JSP

Servlet	JSP
Servlet is a java code.	JSP is a HTML-based compilation code.
Writing code for servlet is harder than JSP as it is HTML in java.	JSP is easy to code as it is java in HTML.
Servlet plays a controller role in the ,MVC approach.	JSP is the view in the MVC approach for showing output.
Servlet is faster than JSP.	JSP is slower than Servlet because the first step in the JSP lifecycle is the translation of JSP to java code and then compile.
Servlet can accept all protocol requests.	JSP only accepts HTTP requests.
In Servlet, we can override the service() method.	In JSP, we cannot override its service() method.
In Servlet by default session management is not enabled, user have to enable it explicitly.	In JSP session management is automatically enabled.

Difference between Servlets and JSP (contd...)

In Servlet we have to implement everything like business logic and presentation logic in just one servlet file.	In JSP business logic is separated from presentation logic by using JavaBeansclient-side.
Modification in Servlet is a time-consuming compiling task because it includes reloading, recompiling, JavaBeans and restarting the server.	JSP modification is fast, just need to click the refresh button.
It does not have inbuilt implicit objects.	In JSP there are inbuilt implicit objects.
There is no method for running JavaScript on the client side in Servlet.	While running the JavaScript at the client side in JSP, client-side validation is used.
Packages are to be imported on the top of the program.	Packages can be imported into the JSP program (i.e, bottom , middleclient-side, or top)
It can handle extensive data processing.	It cannot handle extensive data processing very efficiently.
The facility of writing custom tags is not present.	The facility of writing custom tags is present.
Servlets are hosted and executed on Web Servers.	Before the execution, JSP is compiled in Java Servlets and then it has a similar lifecycle as Servlets.