

Lecture 10 - Java Web

Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

CSC-1004: Computational Laboratory Using Java
Course Page: [\[Click\]](#)

Outline

- Hyper Text Markup Language (HTML)
- Java Servlet



Java Servlet Introduction

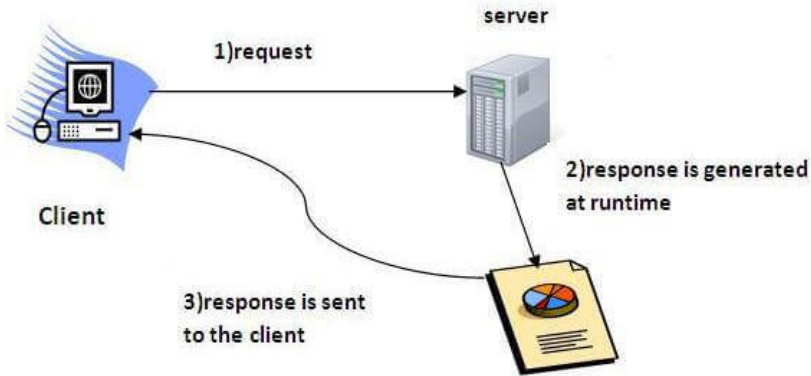
- Servlet is a technology that is used to create a web application.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to incoming requests. It can respond to any requests.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Java Servlet Introduction



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Web Terminology

- **Static Website.** The web pages are coded in HTML. The codes are fixed so the information contained in the page does not change and it looks like a printed page.

Static Website



Server



Client/Browser

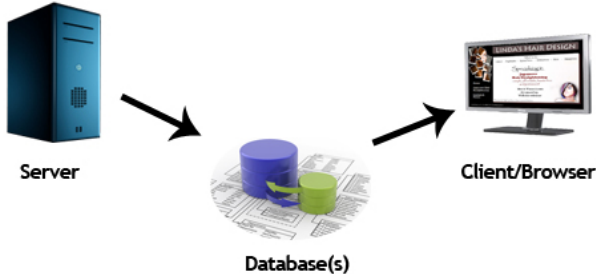


香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Web Terminology

- **Dynamic Website.** The web pages access content from a database or Content Management System (CMS). Therefore, when you alter or update the content of the database, the content of the website is also altered or updated.

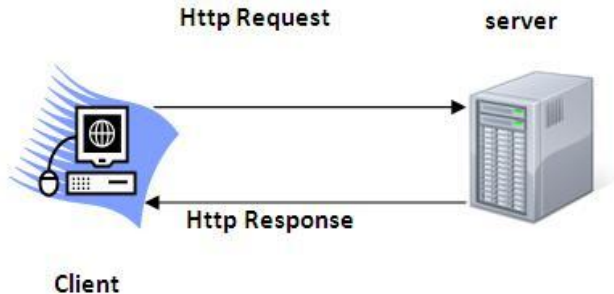
Dynamic Website



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Web Terminology

- **HTTP (Hyper Text Transfer Protocol)** is an application-level protocol for collaborative, distributed, hypermedia information systems. It is the data communication protocol used to establish communication between the client and server.



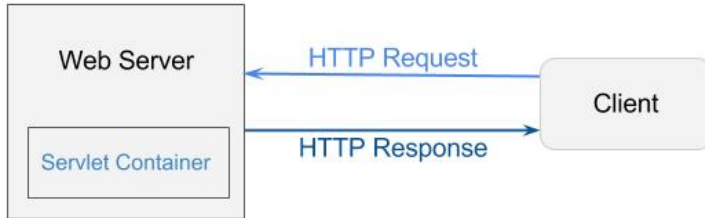
Web Terminology

- **HTTP request** is the request sent by the computer to a web server that contains all sorts of potentially interesting information.

HTTP Request	Description
GET	Asks to get the 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.
HEAD	Asks for only the header part of whatever a GET would return. Just like GET but with no body.
TRACE	Asks for the loopback of the request message, for testing or troubleshooting.
PUT	Says to put the enclosed info (the body) at the requested URL.
DELETE	Says to delete the resource at the requested URL.
OPTIONS	Asks for a list of the HTTP methods to which the thing at the request URL can respond

Web Terminology

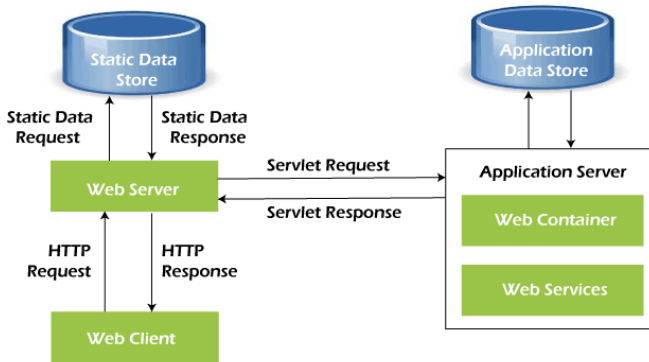
- **Servlet Container.** The client/user can request only static WebPages from the server. If the user wants to read the web pages as per input then the servlet container is used in Java.



Web Terminology

- **Web Server.** Web server contains only web or servlet container. It is a computer where the web content can be stored.

Working of web servers



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Web Terminology

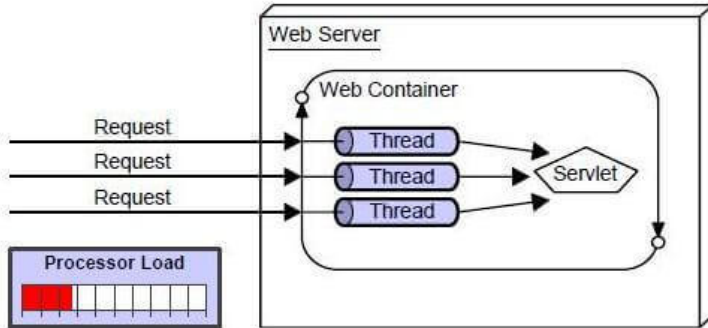
- **Web Server.** Web server contains only web or servlet container. It is a computer where the web content can be stored.
 1. If the requested web page at the client side is not found, then the web server will send the HTTP response: Error 404 Not found.
 2. When the web server searches the requested page if the requested page is found then it will send the client an HTTP response.
 3. If the client requests some other resources then the web server will contact to the application server and data is stored for constructing the HTTP response.



Web Terminology

- **Advantages of Servlet**

1. **Better performance:** because it creates a thread for each request, not process.
2. **Robust:** JVM manages Servlets, so we don't need to worry about memory leak, garbage collection, etc.



Java Servlet Interface

A Servlet interface needs to be implemented for creating any servlet. It provides 3 life cycle methods that are used to initialize the servlet, to service the requests, and to destroy the servlet and 2 non-life cycle methods.

Method	Description
public void init(ServletConfig config)	initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once.
public void service(ServletRequest request, ServletResponse response)	provides response for the incoming request. It is invoked at each request by the web container.
public void destroy()	is invoked only once and indicates that servlet is being destroyed.
public ServletConfig getServletConfig()	returns the object of ServletConfig.
public String getServletInfo()	returns information about servlet such as writer, copyright, version etc.

之大學(深圳)

University of Hong Kong, Shenzhen

Java Servlet Interface

Please check the following example:

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

public class First implements Servlet{
    ServletConfig config=null;

    public void init(ServletConfig config){
        this.config=config;
        System.out.println("servlet is initialized");
    }

    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>hello simple servlet</b>");
        out.print("</body></html>");

    }

    public void destroy(){System.out.println("servlet is destroyed");}
    public ServletConfig getServletConfig(){return config;}
    public String getServletInfo(){return "copyright 2007-1010";}

}
```



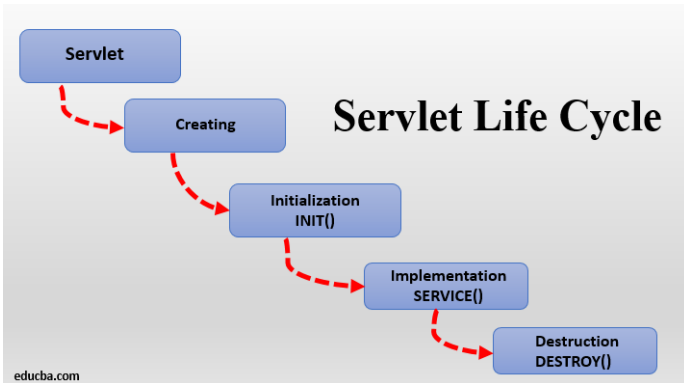
香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Life Cycle of a Servlet

- Servlet class is loaded.

The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.

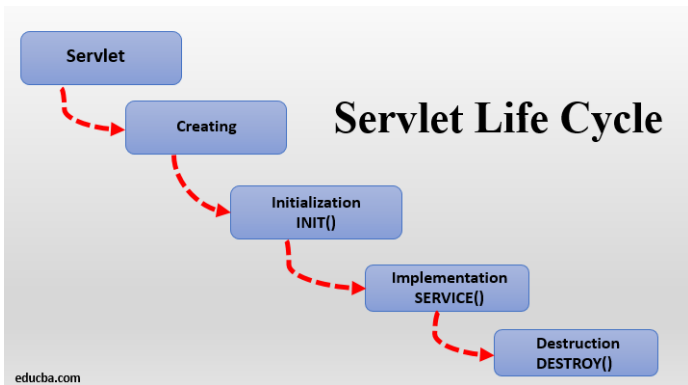


香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Life Cycle of a Servlet

- **Servlet instance is created.**

The web container creates the instance of a servlet after loading the servlet class.
The servlet instance is created only once in the servlet life cycle.

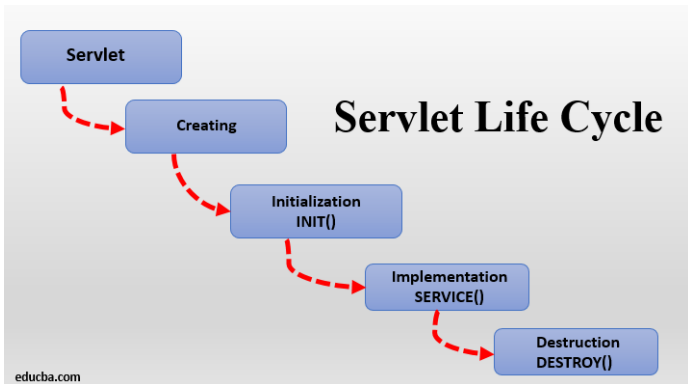


香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Life Cycle of a Servlet

- **Init method is invoked.**

The web container calls the init method only once after creating the servlet instance. The init method is used to initialize the servlet.



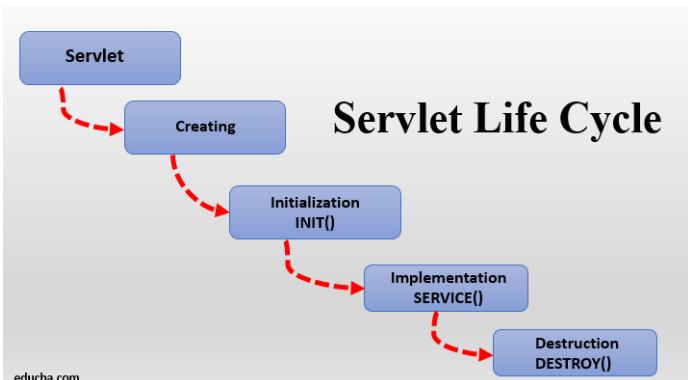
香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Life Cycle of a Servlet

- **Service method is invoked.**

The web container calls the service method each time when the request for the servlet is received. If the servlet is not initialized, it follows the first three steps as described above and then calls the service method.



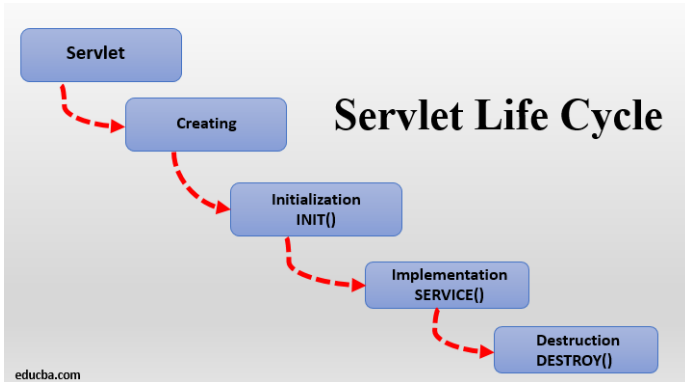
香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Life Cycle of a Servlet

- **Destroy method is invoked.**

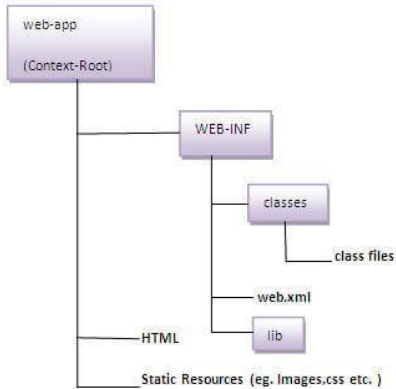
The web container calls the destroy method before removing the servlet instance from the service. It gives the servlet an opportunity to clean up any resource.



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Servlet Example

- **Create a directory structure.** The directory structure defines where to put files so that the web container may get the information and respond to the client.



Servlet Example

- **Create a Servlet.** The HttpServlet class creates the servlet because it provides methods to handle http requests such as doGet(), doPost, doHead() etc.

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

public class DemoServlet extends HttpServlet{
    public void doGet(HttpServletRequest req,HttpServletResponse res)
        throws ServletException,IOException
    {
        res.setContentType("text/html");//setting the content type
        PrintWriter pw=res.getWriter();//get the stream to write the data

        //writing html in the stream
        pw.println("<html><body>");
        pw.println("Welcome to servlet");
        pw.println("</body></html>");

        pw.close();//closing the stream
    }
}
```



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Servlet Example

- **Compile the servlet.** For compiling the Servlet, jar file is required to be loaded. Different Servers provide different jar files.

Jar file	Server
1) servlet-api.jar	Apache Tomcat
2) weblogic.jar	Weblogic
3) javaee.jar	Glassfish
4) javaee.jar	JBoss



Servlet Example

- **Create the deployment descriptor.** The deployment descriptor is an xml file, from which Web Container gets the information about the servlet to be invoked.

```
<web-app>

<servlet>
<servlet-name>sonoojaiswal</servlet-name>
<servlet-class>DemoServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>sonoojaiswal</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>

</web-app>
```



Servlet Example

- **Create the deployment descriptor.** The deployment descriptor is an xml file, from which Web Container gets the information about the servlet to be invoked.
`<web-app>` represents the whole application.
`<servlet>` is sub element of `<web-app>` and represents the servlet.
`<servlet-name>` is sub element of `<servlet>` represents the name of the servlet.
`<servlet-class>` is sub element of `<servlet>` represents the class of the servlet.
`<servlet-mapping>` is sub element of `<web-app>`. It is used to map the servlet.
`<url-pattern>` is sub element of `<servlet-mapping>`. This pattern is used on client side to invoke the servlet.



Servlet Example

- **Start the Server and deploy the project.**

To start Apache Tomcat server, double click on the startup.bat file under apache-tomcat/bin directory.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Java ServletRequest Interface

An object of ServletRequest is used to provide the client request information to a servlet such as content type, content length, parameter names and values, header information, attributes etc.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Java ServletRequest Interface

In this example, we are displaying the name of the user in the servlet by using the `getParameter` method that returns the value for the given request parameter name:

```
<form action="welcome" method="get">  
Enter your name<input type="text" name="name"><br>  
<input type="submit" value="login">  
</form>
```

```
import javax.servlet.http.*;  
import javax.servlet.*;  
import java.io.*;  
public class DemoServ extends HttpServlet{  
    public void doGet(HttpServletRequest req,HttpServletResponse res)  
        throws ServletException,IOException  
    {  
        res.setContentType("text/html");  
        PrintWriter pw=res.getWriter();  
  
        String name=req.getParameter("name");//will return value  
        pw.println("Welcome "+name);  
  
        pw.close();  
    }  
}
```

Java ServletConfig Interface

An object of ServletConfig is created by the web container for each servlet. This object can be used to get configuration information from the **web.xml** file.

- The core **advantage** of ServletConfig is that you don't need to edit the servlet file if the information is modified from the web.xml file.
- Syntax to provide the initialization parameter for a servlet:

```
<init-param>  
  <param-name>parametername</param-name>  
  <param-value>parametervalue</param-value>  
</init-param>
```



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Java ServletConfig Interface

In this example, we print all the initialization parameters from the web.xml file.

```
public class DemoServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
  
        ServletConfig config=getServletConfig();  
        Enumeration<String> e=config.getInitParameterNames();  
  
        String str="";  
        while(e.hasMoreElements()){  
            str=e.nextElement();  
            out.print("<br>Name: "+str);  
            out.print(" value: "+config.getInitParameter(str));  
        }  
  
        out.close();  
    }  
}
```

```
<web-app>  
  
    <servlet>  
        <servlet-name>DemoServlet</servlet-name>  
        <servlet-class>DemoServlet</servlet-class>  
  
        <init-param>  
            <param-name>username</param-name>  
            <param-value>system</param-value>  
        </init-param>  
  
        <init-param>  
            <param-name>password</param-name>  
            <param-value>oracle</param-value>  
        </init-param>  
  
    </servlet>  
  
    <servlet-mapping>  
        <servlet-name>DemoServlet</servlet-name>  
        <url-pattern>/servlet1</url-pattern>  
    </servlet-mapping>
```

- 香港中文大學 (深圳)

the Chinese University of Hong Kong, Shenzhen

Java ServletContext Interface

An object of ServletContext is created by the web container at the time of deploying the project. This object can be used to get configuration information from **web.xml** file. There is only one ServletContext object per web application.

- The core **advantage** of ServletConfig is easy to maintain. if any information is shared with all the servlets, it is better to make it available for all the servlets.
- Syntax to provide the initialization parameter for a servlet:

```
<context-param>  
  <param-name>parametername</param-name>  
  <param-value>parametervalue</param-value>  
</context-param>
```



Creating Servlet Example in IntelliJ IDEA



IntelliJ IDEA

Please follow the video in this [Link](#).



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

Question and Answering (Q&A)



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen