NIIT

Objectives

- In this session, you will learn to:
  - Understand servlets
  - Implement servlets

NIIT

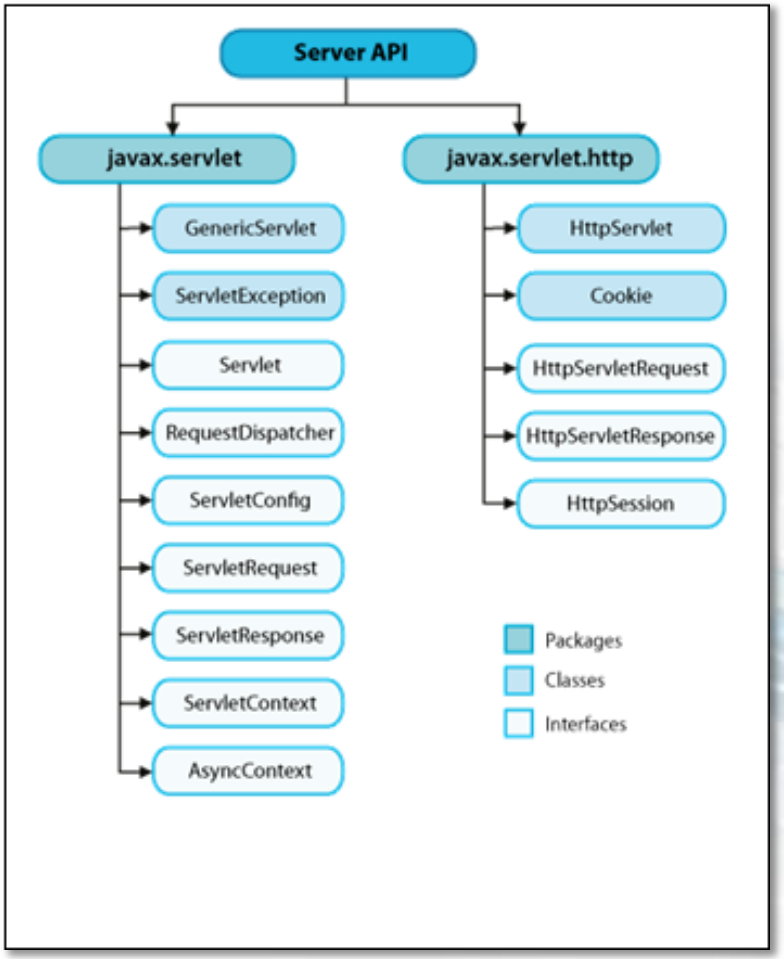
The Servlet API

- Servlet API:
  - is used to create and manage servlets.
  - consists of various interfaces, classes, and methods.
  - classes ad methods are available in the following packages:
    - `javax.servlet`
    - `javax.servlet.http`

NIIT

The Servlet API (Contd.)

- The following figure displays the class hierarchy of the Servlet API.



NIIT

The Servlet API (Contd.)

- The `javax.servlet` package:
  - consists of the following interfaces, as described in the table.

Interface	***Description***
`Servlet`	*It provides the methods that all servlets must implement.*
`ServletConfig`	*It provides information to the servlets during the servlet initialization.*
`ServletContext`	*It provides methods, which is used by the servlets to communicate with the Web container.*
`ServletRequest`	*It provides the client request information to the servlet.*
`ServletResponse`	*It helps the servlet in sending the response to the client.*
`RequestDispatcher`	*It receives a request from a client and sends it to any other resource, such as a servlet, an HTML page, or a JSP page.*
`AsyncContext`	*It provides the methods to handle multiple client requests asynchronously.*

NIIT

The Servlet API (Contd.)

- The `Servlet` interface:
  - provides the methods that must be implemented by all the servlets.
  - methods are implemented by the servlet class that extends from a `GenericServlet` or `HttpServlet` class.
- The various methods of the `Servlet` class are:
  - `void init(ServletConfig config) throws ServletException`
  - `ServletConfig getServletConfig()`
  - `String getServletInfo()`
  - `void service(ServletRequest request, ServletResponse response) throws ServletException, IOException`
  - `void destroy()`

NIIT

The Servlet API (Contd.)

- The `ServletConfig` interface:
  - is implemented by a Web container during the initialization of a servlet to pass the configuration information to a servlet.
  - is passed to the `init()` method of the servlet during initialization.
  - can be used to pass initialization parameters to the servlets.
- The initialization parameters are passed as name-value pairs.
- For example, the connection URL can be passed as an initialization parameter of the servlet.

NIIT

The Servlet API (Contd.)

- Some of the methods of the `ServletConfig` interface are:
  - `String getInitParameter(String param)`: It returns a `String` object containing the value of the initialization parameters.
  - `Enumeration<String> getInitParameterNames()`: It returns the names of all the initialization parameters as an enumeration of `String` objects.
  - `ServletContext getServletContext()`: It returns the `ServletContext` object for the servlet in which the caller is executing.
  - `String getServletName()`: It returns a `String` object containing the name of the servlet instance.

NIIT

The Servlet API (Contd.)

- The `ServletContext` interface provides the environmental information to the servlets in which they are running.
- Each Web application consists of only one `ServletContext` object.
- A `ServletContext` object is also known as a Web context.
- The `ServletContext` object provides methods that can be used to communicate with the Web container.

NIIT

The Servlet API (Contd.)

- Some of the methods of the `ServletContext` interface are:
  - `public void setAttribute(String, Object)`: It binds the object with a name and stores the name-value pair as an attribute of the `ServletContext` object.
  - `public Object getAttribute(String attrname)`: It returns the object stored in the `ServletContext` object with the name passed as a parameter.
  - `public String getInitParameter(String pname)`: It returns the value of the initialization parameter with the name passed as a parameter.

NIIT

The Servlet API (Contd.)

- The `ServletRequest` interface enables a servlet to handle the client requests by providing the client information to the servlet.
- Some of the methods of the `ServletRequest` interface are:
  - `public String getParameter(String paramName)`
  - `public String[] getParameterValues(String paramName)`
  - `public Enumeration getParameterNames()`
  - `public String getRemoteHost()`
  - `public String getRemoteAddr()`

NIIT

The Servlet API (Contd.)

- The `ServletResponse` interface contains methods that enable a servlet to respond to the client requests.
- Some of the methods of the `ServletRequest` interface are:
  - `public ServletOutputStream getOutputStream() throws IOException`
  - `public PrintWriter getWriter() throws IOException`
  - `public void setContentType(String type)`

NIIT

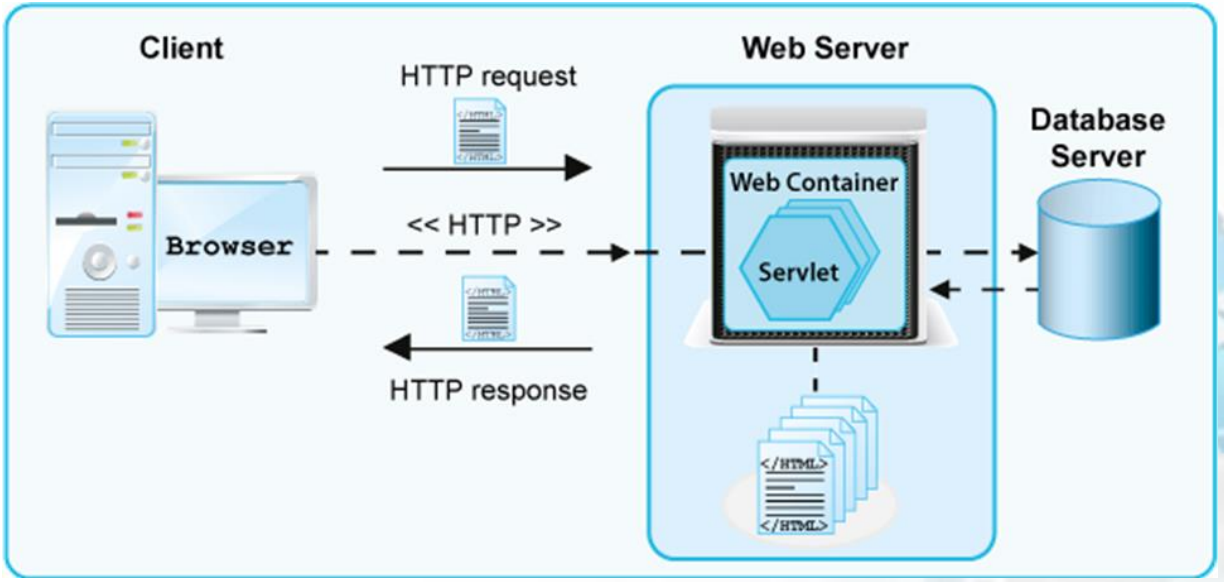
The Servlet API (Contd.)

- The `javax.servlet.http` package:
  - provides classes and interfaces that enable you to create a servlet that communicates using the HTTP protocol.
  - package provides the `HttpServlet` class to create servlets. It is an abstract class.
- Some of the interfaces of the `javax.servlet.http` package are:
  - `HttpServletRequest`
  - `HttpServletResponse`
  - `HttpSession`

NIIT

Understanding the Web Container

- Web container:
  - is a component of the Web server which provides the run-time environment for executing the servlets.
- The following figure depicts the Web container architecture.



NIIT

Understanding the Web Container (Contd.)

- Web container provides the following services to the Web applications:
  - Communication support
  - Lifecycle management
  - Multithreading support
  - Declarative security
  - JSP support

NIIT

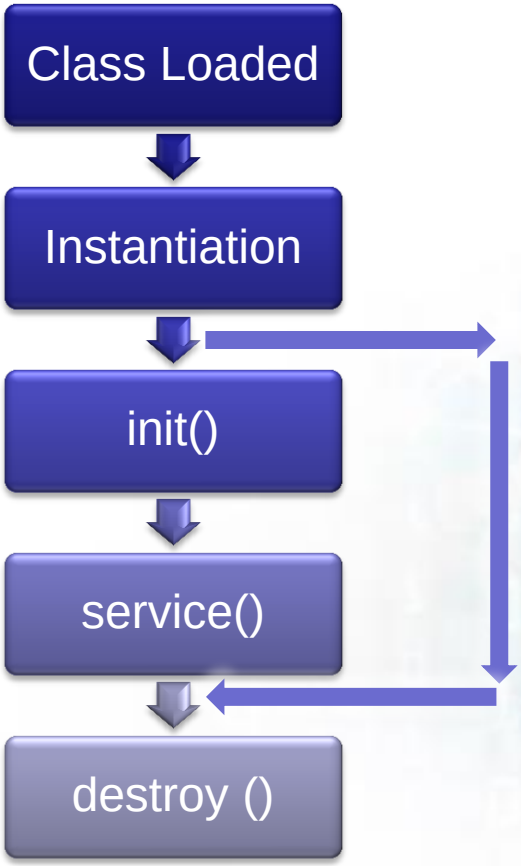
Life Cycle of a Servlet

- The life cycle of a servlet is managed by the Web container, by using the following methods of the `Servlet` interface:
  - `init()`
  - `service()`
  - `destroy()`

NIIT

Life Cycle of a Servlet (Contd.)

- The following animation depicts the execution sequence of the lifecycle method.



NIIT

Processing of a Servlet Request

- The following steps describe processing of a client request by a servlet:
  1. A user sends a request to the server. The server then forwards the request to the Web container.
  2. The Web container creates the `HttpServletRequest` and `HttpServletResponse` objects for request handling and response generation.
  3. The Web container initializes the identified servlet thread by calling the `init()` method and passes the `HttpServletRequest` and `HttpServletResponse` objects to the initialized servlet thread.
  4. The `service()` method of the servlet is called to process the client requests.
  5. Depending upon the type of request, the `service()` method invokes the `doGet()` or `doPost()` method.

NIIT

Processing of a Servlet Request (Contd.)

6. The `doGet()` or `doPost()` method generates a response for the client and attaches it with the response object.
7. The Web container sends the generated response to the client.
8. The Web container deletes the servlet instance along with the request and response objects.

NIIT

Implementing Servlets

- To create a servlet, you need to perform the following tasks:
  1. Create a servlet.
  2. Configure the servlet.
  3. Compile and package the servlet.
  4. Deploy the servlet.
  5. Access the servlet using a browser application.

NIIT

Creating a Servlet

- Servlet can be created by extending the `HttpServlet` or `GenericServlet` class.
- Example:
  - Consider an example, where you have been asked to create a servlet, which displays the current date to the user, each time the servlet is accessed. To implement this, you have to create a servlet that extends from the `HttpServlet` class, as shown in the embedded file:

Microsoft Office
Word Document

NIIT

Configuring a Servlet

- Configuring means to associating a URL with the servlet.
- Servlet can be configured by using:
  - Deployment Descriptor (DD): is an XML file named web.xml, which contains configuration information.
  - annotations: is a metadata that can be added to the code. It is a keyword that is used to add extra explanation to various programming elements, such as classes, fields, and methods in a program.

NIIT

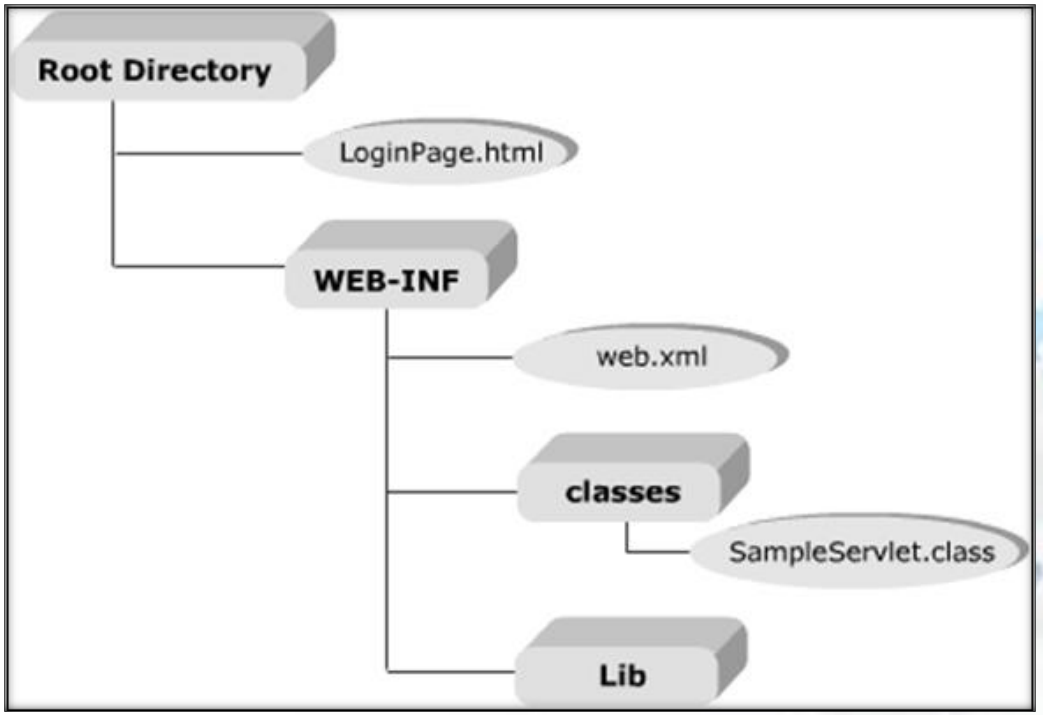
Configuring a Servlet (Contd.)

- Some of the commonly used elements of the web.xml file along with their usage are:
  - `<servlet>`: specifies the configuration details of all the servlets used in a Web application and contains the following sub elements:
    - `<servlet-name>`
    - `<servlet-class>`
  - `<servlet-mapping>`: consists of following sub elements:
    - `<servlet-name>`
    - `<url-pattern>`
  - `<session-config>`
  - `<init-param>`
  - `<mime-mapping>`

NIIT

Compiling and Packaging a Servlet

- Compiling the servlet creates a class file.
- Packaging refers to package the application into standard Java EE structure, as shown in the following figure.



NIIT

Summary

- In this session, you learned that:
  - Servlet is a Java program that runs on the server side. It inherits all the features of the Java programming language.
  - The Servlet API contains various interfaces, classes, and methods that are used to create and manage servlets. These classes and interfaces are available in the following packages:
    - `javax.servlet`
    - `javax.servlet.http`
  - The `Servlet` interface of the `javax.servlet` package defines the methods that the Web container calls to manage the servlet life cycle.
  - The `javax.servlet.ServletConfig` interface is implemented by a Web container during the initialization of a servlet to pass the configuration information to the servlet.