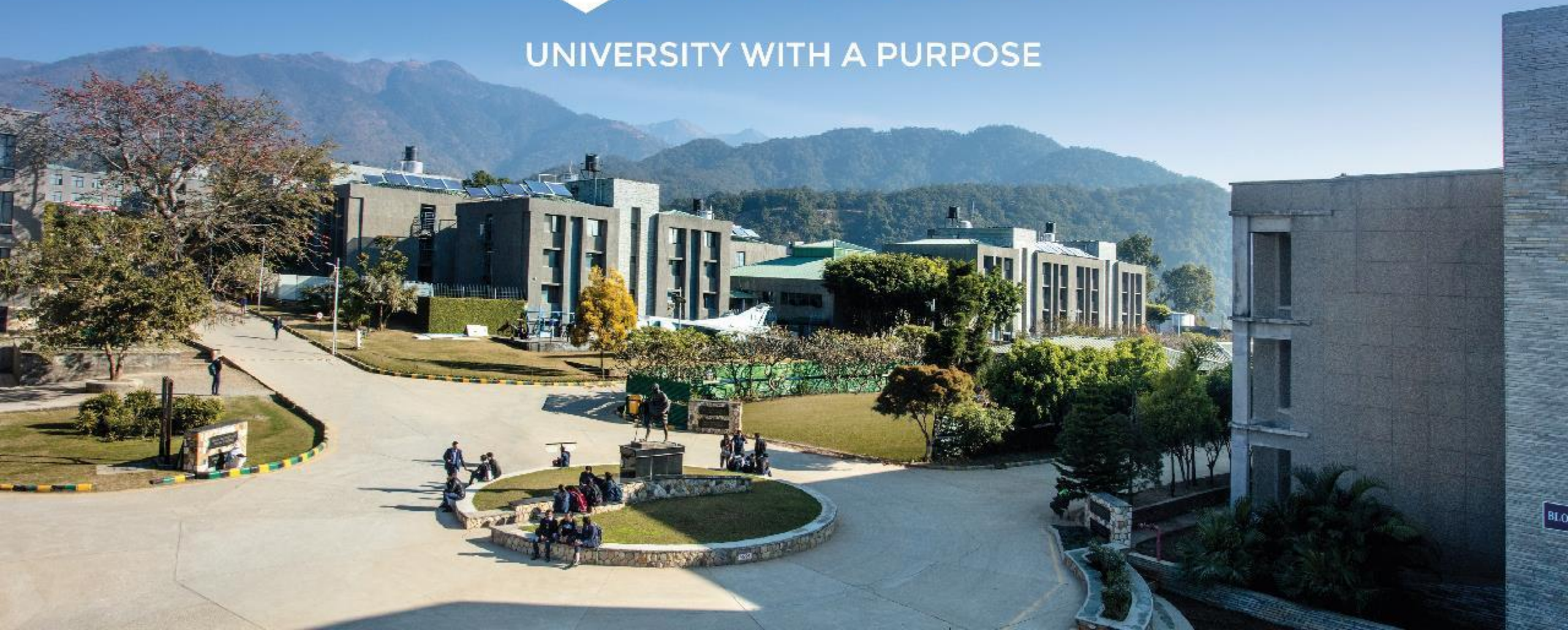


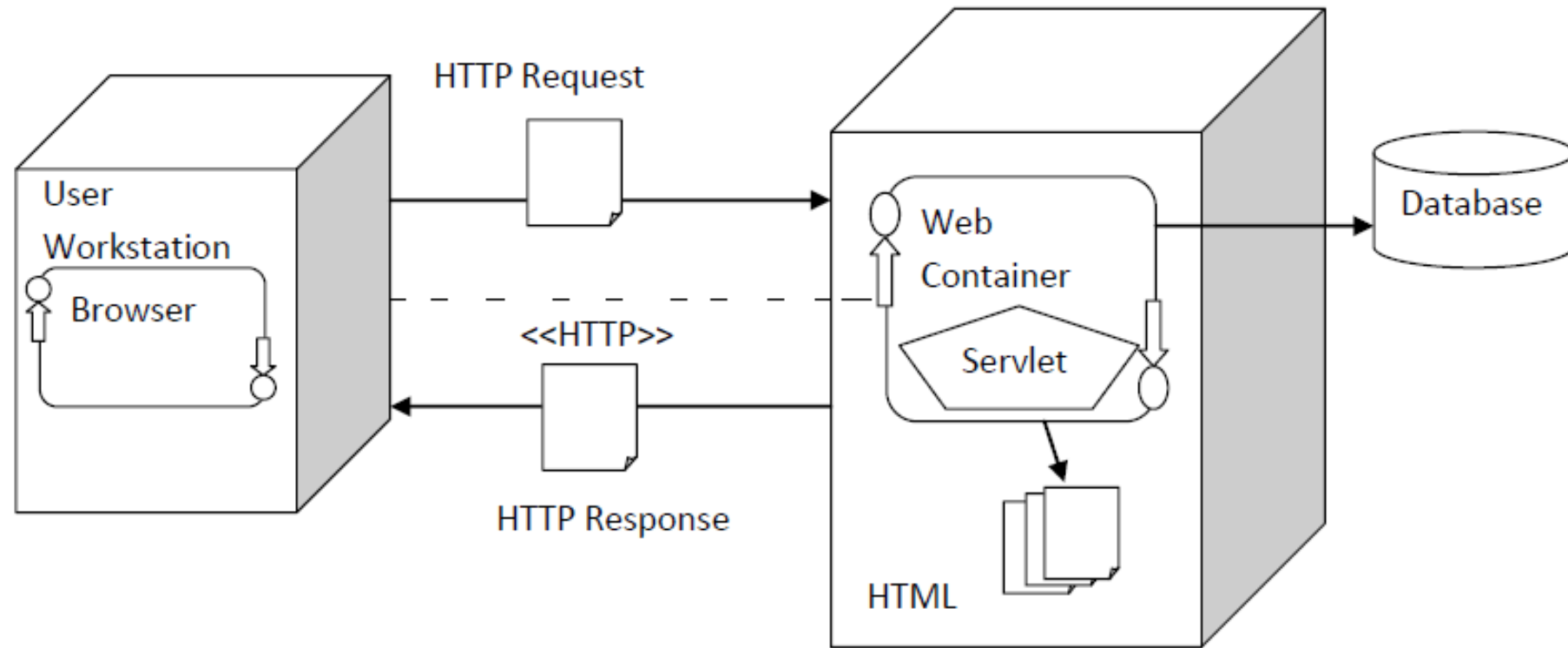


UNIVERSITY WITH A PURPOSE



# Servlet-III

# Servlet



# GenericServlet class

public abstract class **GenericServlet**  
**extends** java.lang.Object  
**implements** Servlet, ServletConfig, java.io.Serializable

- GenericServlet implements the Servlet and ServletConfig interfaces.
- GenericServlet makes writing servlets easier. It provides simple versions of the lifecycle methods init and destroy and of the methods in the ServletConfig interface.
- GenericServlet also implements the log method, declared in the ServletContext interface.
- To write a generic servlet, you need only override the abstract service method.



# GenericServlet methods

| Method           | Summary  |
|------------------|--|
| void             | <b>destroy()</b> : Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.  |
| java.lang.String | <b>getInitParameter</b> (java.lang.String name): Returns a String containing the value of the named initialization parameter, or null if the parameter does not exist. |
| ServletConfig    | <b>getServletConfig()</b> : Returns this servlet's ServletConfig object.   |
| ServletContext   | <b>getServletContext()</b> : Returns a reference to the ServletContext in which this servlet is running.   |
| java.lang.String | <b>getServletInfo()</b> : Returns information about the servlet, such as author, version, and copyright.   |
| java.lang.String | <b>getServletName()</b> : Returns the name of this servlet instance.   |
| void             | <b>init()</b> : A convenience method which can be overridden so that there's no need to call super.init(config).   |
| void             | <b>init</b> (ServletConfig config): Called by the servlet container to indicate to a servlet that the servlet is being placed into service.                            |
| void             | <b>log</b> (java.lang.String msg): Writes the specified message to a servlet log file, prepended by the servlet's name.  |
| abstract void    | <b>service</b> (ServletRequest req, ServletResponse res): Called by the servlet container to allow the servlet to respond to a request.                                |

# HttpServlet class

## All Implemented Interfaces:

Serializable, Servlet, ServletConfig

public abstract class **HttpServlet** extends GenericServlet

- Provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site.
- A subclass of HttpServlet must override at least one method, usually one of these:
  - doGet**, if the servlet supports HTTP GET requests
  - doPost**, for HTTP POST requests
  - doPut**, for HTTP PUT requests
  - doDelete**, for HTTP DELETE requests
  - init and destroy**, to manage resources that are held for the life of the servlet
  - getServletInfo**, which the servlet uses to provide information about itself
- There's almost no reason to override the service method. service handles standard HTTP requests by dispatching them to the handler methods for each HTTP request type (the doXXX methods listed above).

# HttpServlet methods

| Method         | Summary   |
|----------------|---|
| protected void | <b>doDelete(HttpServletRequest req, HttpServletResponse resp)</b><br>Called by the server (via the service method) to allow a servlet to handle a DELETE request. The DELETE operation allows a client to remove a document or Web page from the server.  |
| protected void | <b>doGet(HttpServletRequest req, HttpServletResponse resp)</b><br>Called by the server (via the service method) to allow a servlet to handle a GET request.   |
| protected void | <b>doPost(HttpServletRequest req, HttpServletResponse resp)</b><br>Called by the server (via the service method) to allow a servlet to handle a POST request.   |
| protected void | <b>doPut(HttpServletRequest req, HttpServletResponse resp)</b><br>Called by the server (via the service method) to allow a servlet to handle a PUT request. The PUT operation allows a client to place a file on the server and is similar to sending a file by FTP.  |
| protected void | <b>doHead(HttpServletRequest req, HttpServletResponse resp)</b><br>Receives an HTTP HEAD request from the protected service method and handles the request.   |
| protected void | <b>doOptions(HttpServletRequest req, HttpServletResponse resp)</b><br>Called by the server (via the service method) to allow a servlet to handle a OPTIONS request. The OPTIONS request determines which HTTP methods the server supports and returns an appropriate header. For example, if a servlet overrides doGet, this method returns the following header:<br>Allow: GET, HEAD, TRACE, OPTIONS |
| protected void | <b>service(HttpServletRequest req, HttpServletResponse resp):</b> Receives standard HTTP requests from the public service method and dispatches them to the doXXX methods defined in this class.  |

# ServletConfig and ServletContext

- Both are Interfaces whose implementation is provided by various vendors accordingly.
- Object for ServletContext is only once for whole web application created by web container at the time of deployment of web application.
- Object of ServletConfig is separate for each Servlet created for web container at the time of creating the object of Servlet.

**Note:** One ServletConfig for per servlet and once ServletContext for per web application.

**Why ServletContext Interface:** Usages are as follows:

1. The object of Servlet Context provides an interface between the container and servlet.
2. The ServletContext object can be used to get configuration information from the web.xml file.
3. The ServletContext object can be used to set, get, or remove attribute from the web.xml file.
4. The ServletContext object can be used to provide the inter-application communication.



# ServletConfig and ServletContext

## **How to get the object of ServletContext interface:**

1. `getServletContext()` method of `ServletConfig` interface returns the object of `ServletContext`.
2. `getServletContext()` method of `GenericServlet` class returns the object of `ServletContext`.

## **Advantage of ServletConfig interface:**

The main advantage of `ServletConfig` is that you don't need to edit the Servlet file if information is modified from the `web.xml` file.

## **How to get the object of ServletConfig interface:**

1. `getServletConfig ()` method of `Servlet` interface returns the object of `ServletConfig`

# Interface ServletConfig

## All Known Implementing Classes:

GenericServlet, HttpServlet

### **public interface ServletConfig**

- A servlet configuration object used by a servlet container to pass information to a servlet during initialization.
- One ServletConfig per servlet.

# ServletConfig methods

| Method              | Summary  |
|---------------------|--|
| String              | <b>getInitParameter(String name)</b><br>Gets the value of the initialization parameter with the given name.  |
| Enumeration<String> | <b>getInitParameterNames()</b><br>Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters. |
| ServletContext      | <b>getServletContext()</b><br>Returns a reference to the ServletContext in which the caller is executing.  |
| String              | <b>getServletName()</b><br>Returns the name of this servlet instance.  |

# Interface ServletContext

## **public interface ServletContext**

- Defines a set of methods that a servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.
- There is one context per "web application" per Java Virtual Machine.
- The ServletContext object is contained within the ServletConfig object, which the Web server provides the servlet when the servlet is initialized.

# ServletContext methods

**GoTo:** <https://docs.oracle.com/javaee/7/api/javax/servlet/ServletContext.html>

# References

1. Schildt, H. (2014). *Java: the complete reference*. McGraw-Hill Education Group.
2. <https://docs.oracle.com/javaee/7/api/javax/servlet/ServletContext.html>
3. <https://docs.oracle.com/javaee/7/api/javax/servlet/ServletConfig.html>



# THANK YOU

