

# SERVLETS

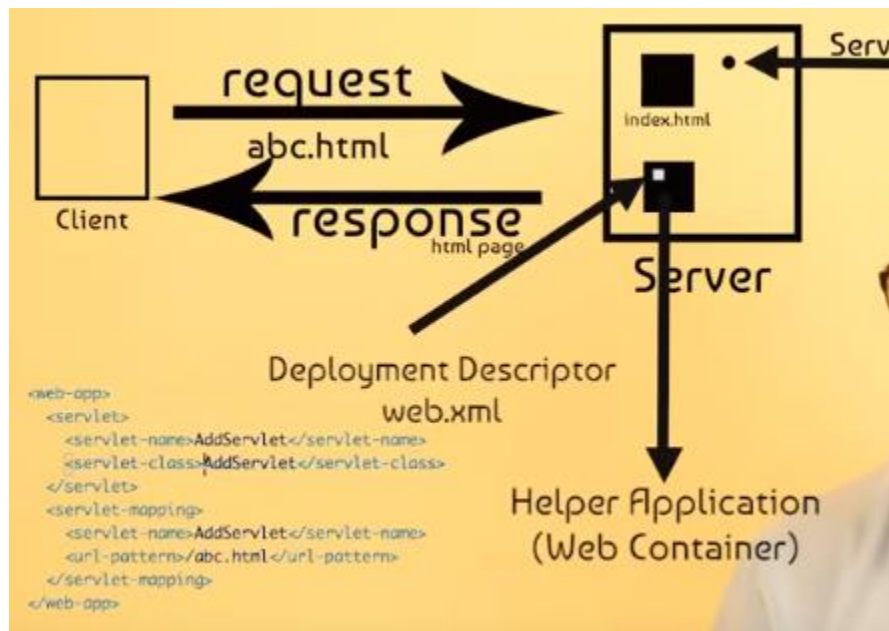
## 1. What is servlet?

It is a class used to extend capabilities of server that host applications by means of request response programming model. Servlets dynamically extend the functionality of the web server.

## 2. What is servlet? Explain in brief.

- Servlet is an interface.
- Servlet is a small java program which runs within web server.
- Servlet receives and responds to requests from the clients across http.
- Which comes under javax package.
- There are two ways to implement this servlet one is by implementing Servlet interface with GenericServlet class and another way is by extending GenericServlet to HttpServlet class.
- This Servlet interface defines methods to initialize servlet, to service requests and to remove a servlet from the server.
- These methods are called life cycle methods of servlet.

## 3. How servlet works?



- Initially the request from the client goes to the server.
- The server sends it to the web container (Helper application).
- The web container uses Deployment Descriptor (web.xml) to detect which servlet should call.
- The servlet processes the information and sends the response to the client.

- The response may be in html, jsp or in any forms.

List out the HTTP protocol methods?

- 1) Get
- 2) Post
- 3) Put
- 4) Options
- 5) Delete

The get() is used to get information from server to client. The post() is used to send data to the server. The post() will not put data into url (query string).

4. What is cookie?

A cookie is information that is present between multiple client requests. A cookie has a name, a single value and attributes such as comment, path, version number, and maximum age.

5. Differentiate between PrintWriter and ServletOutputStream?

PrintWriter	ServletOutputStream
It is a character stream class.	It is byte stream class.
Used to write character based information.	Used to write primitive as well as character based information.

What is servlet mappings?

Servlet mapping is used to map the requests to servlets. Servlet mapping is a process of defining an association between URL pattern and a servlet.

Difference between GenericServlet and HttpServlet?

Both GenericServlet and HttpServlet classes are abstract classes, so we can't create instance of the classes. The HttpServlet basically extends the GenericServlet and also inherits the properties from the GenericServlet.

GenericServlet	HttpServlet
It is protocol independent.	It is a protocol dependent.
Supports only the service() method.	Supports doGet(), doPost(), and other methods.
Belongs to java.servlet package.	Belongs to java.servlet.http package.

What is RequestDispatcher?

The RequestDispatcher is an interface it defines object that receives the request from the client and sends or dispatches it to the other resource such as JSP, HTML, or Servlet. It can call any type of resource with the help of

```
RequestDispatcher dispatcher = Request.getRequestDispatcher("any.jsp");  
dispatcher.forward(request, response);
```

The RequestDispatcher has two methods to receive and respond to the request.

1) forward (ServletRequest request, ServletResponse response) { .... }

This method forwards the requests from one client to the another resource. Usually the requests from the client to server are sending using the request object.

2) Include (ServletRequest request, ServletResponse response) { ... }

This method includes the content of the resources in the response object.

What is servlet filter?

A Filter is an object that is invoked either at the pre-processing or post-processing of a request.

The Servlet Filter is used for:

- Logging the request parameters to log files.
- Authentication and Authorization of the request for the needed resources.
- Formatting the request body/header before sending it to the servlet.
- Compressing the response data sent to client.
- Change the response by adding some cookie and header information.

What is load-on-startup in servlet?

The load-on-startup element in the web.xml or within the @WebServlet annotation is used to load the servlet at the time of deploying the project or the server to start. This saves time for the response of the first request.

Explain MVC pattern?

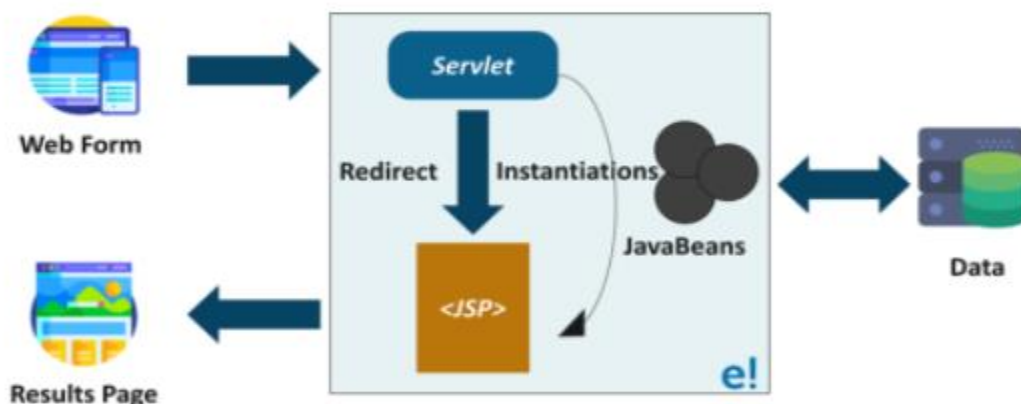
The MVC is a design pattern which divides application into three parts namely Model, View, and Controller.

**Model:** A model deals with behavior of the application, contains database connectivity, business logic. It notifies View and Controller if any changes occur.

**View:** A view displays information to the user. It takes information from the model using which it generates output.

**Controller:** A controller takes inputs from the user and sends the commands to model or view. It controls the flow of the application.

What is workflow of Servlet?



A Servlet is a server side programming language, is used to create java web applications. It is located at the server side and helps to generate dynamic web pages.

It acts as a mediator between incoming HTTP request from the browser and database.

A request is received from a webpage by the servlet. The servlet redirects request to the appropriate JSP pages and JSP page sends response as a result.

### 3. Life-cycle methods of servlet?

- The servlet is constructed then initialized with `init()` method.
- Any calls from clients to the `service()` method are handled.
- If the servlet is taken out of service, then destroyed with the `destroy()` method and then garbage is collected and finalized.

In addition to life-cycle methods servlet also provides `getServletConfig()` by this servlet can be used to get any startup information, `getServletInfo()` gives the basic information about the servlet itself such version, author.

`init(ServletConfig config)`: this method is used by the container to initialize the servlet. This method is invoked only once in the lifecycle of the servlet.

service(ServletRequest req, ServletResponse resp): this method is called once for every request, a container can't invoke service method until init() is executed.

Destroy(): the destroy method is called after the service has been completed.

What is deadlock?

A deadlock can occur when multiple threads competing for shared resources, means each thread is waiting for a resource that the another thread holds.

Deadlocks typically arise when multiple threads are accessing shared resources such as database connections, files.

Deadlock situation can be created in servlet by calling doPost() inside the doGet() or doGet() inside the doPost() will create a deadlock situation.

Explain ServletConfig object?

The ServletConfig object is used to pass configuration information to servlet. Every servlet has its own ServletConfig object and servlet container is responsible for instantiating the object.

Difference between ServletConfig and ServletContext?

ServletConfig	ServletContext
ServletConfig is a unique object per servlet.	ServletContext is a unique object for a complete application.
ServletConfig is used to provide the init parameters to the servlet.	ServletContext is used to provide application level init parameters that all other servlets can use.
Attributes in the ServletConfig can't be set.	We can set attributes in the ServletContext that other servlets can use.

What is interServlet communication?

interServlet communication is a method to invoke another servlet using RequestDispatcherforward() and include() methods and provide additional attribute in the request for other servlet use.

What is MIME type?

The Content Type response header is called MIME (Multipurpose Internet Mail Extension) type. The Server sends the MIME type to client, to let the client to know the what kind of data server is

sending. It helps client to rendering the data for the user. Most commonly used MIME types are text/html, text/xml, and etc.

What is use of HttpServletRequestWrapper and HttpServletResponseWrapper?

Both the classes are used to help developers with a custom implementation of request and response types. Programmers can extend these classes and override only specific methods that they need to implement for customized request and response objects.

Explain session management?

Session management is the process of the maintaining the user sessions in the web applications. Session are used to track user interaction with web applications across the HTTP request. This allows the server to maintain the information about the user.

1) Session creation: when a user accesses a web application for the first time, the servlet container (Apache Tomcat) creates a unique session object for the user. This typically involves generating the session id, which is then sent to client browser.

2) Session tracking: the unique Session ID is typically stored in a cookie on the client side. On subsequent requests, the client sends session ID back to the server for identifying the user session.

3) Session attributes: within each session the session attributes can store, which represents key value pairs representing the data associated the user's session.

4) Session timeout: sessions have timeout period; after which they expire if there is no activity from the user.

5) Session invalidation: sessions can also be invalidated programmatically by the servlet. This typically occurs when user logs out, or session expiring.

6) Session management: Java Servlet API provides the session management through which session can be create, access, and manage the session.

Explain CGI?

The CGI stands for Common Gate Interface. It consists of a set of code segments at server side using which server interacts with the client.

The main drawbacks of CGI are:

- 1) If there are multiple incoming requests, then the response generated by the server will be very slow, which results in low efficiency.
- 2) CGI is platform independent.

Explain web container?

A web container or servlet container is used to interact with servlet, that includes servlet, JSP, XML files inside it. Web container's responsibility is to manage the life cycle of a servlet.

What is servlet chaining?

The servlet chaining or looping is the process where output of one servlet is given as input to other servlet is called servlet chaining.

Why servlets doesn't have main()?

Because servlets are executed on web server that why servlets doesn't have main(). The client requests for servlets to server, then the server sends those requests to the web container where the servlets are deployed.

What is servlet context?

Servlet context is an object that contains information about the application and web container. Using the servlet context, we can LOG EVENTS, get the URL of the specific resource, and store the attribute for the other servlet to use.

The servlet context has following methods:

`getInitParameter()` : returns the values of parameter.

`getInitParameterNames()` : returns the names of parameters.

`setAttribute()` : used to set the values of attribute.

`getAttribute()` : used to get the values of the attributes.

`removeAttribute()` : used to remove the attributes.

### 3. Explain Servlet API?

A Servlet does not have main method instead of that Servlet has methods for purpose of handling requests. It invokes service methods every time Server sends a request to Servlet. To handle requests Servlet must override service method that allow two parameters these are request object

and response object. Request object is used to inform the servlet about the request and response object is used to give the response.

#### 4. Define init and destroy methods of servlet?

Servlet Init method is used to initialize the servlet.

After the web container loads and instantiate the Servlet class and before it delivers requests from clients, the web container initializes the servlet. This process allows the servlet to read configuration data, initialize process, and perform any other one-time activities.

When a servlet container determines that servlet should be removed from the service, the container calls the destroy method to destroy the service.

#### 5. What are the methods of Servlet?

Modifier and Type	Method and Description
void	<code>destroy()</code> Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.
<code>ServletConfig</code>	<code>getServletConfig()</code> Returns a <code>ServletConfig</code> object, which contains initialization and startup parameters for this servlet.
<code>String</code>	<code>getServletInfo()</code> Returns information about the servlet, such as author, version, and copyright.
void	<code>init(ServletConfig config)</code> Called by the servlet container to indicate to a servlet that the servlet is being placed into service.
void	<code>service(ServletRequest req, ServletResponse res)</code> Called by the servlet container to allow the servlet to respond to a request.



## WRAPPER CLASSES

### 1. What is wrapper class?

Wrapper classes are classes that allow us to access primitive datatypes as objects.

Ex: Integer is wrapper class it is used to represent primitive int type. We can use wrapper classes to represent primitive types.

Wrapper classes are Byte, Short, Integer, Float, Double, Long, BigInteger, BigDecimal.

### 2. What are the uses of wrapper classes?

- Wrapper classes allow us to create an object.
- We can add them to collections and pass them as arguments to methods.
- They provide methods to manipulate data and the ability to convert to different types.
- They help to improve performance.
- Ex: when we use autoboxed Integer object, the java compiler can convert the object to primitive int value automatically.
- Because compiler does not have to generate code to convert object to primitive value.

```
Integer myInt = new Integer(10);
```

```
int myInt2 = myInt.intValue(); // Converts the Integer object to a primitive int value
```

### 3. How to convert String type to primitive datatype?

It is possible to convert String datatype to primitive types using static *parse()* methods provided by the wrapper classes.

### 4. What is difference between *parseInt()* and *valueOf()* methods?

The *parseInt()* is used to convert String datatype to primitive type, whereas *valueOf()* is used to convert object into a String.

### 5. Are all wrapper classes immutable?

No, not all wrapper classes are immutable. Only Byte, Boolean, Short, Character are immutable. Integer, Float, Double, Long are mutable.

### 6. What does the *compareTo()* method do?

The `compareTo()` method is used to compare two wrapped objects. It returns 0 if two objects are same, positive 1 if the first object is greater than second object and negative -1 if the first object is less than second object.

## 7. Explain autoboxing?

Autoboxing in java is the automatic conversion of primitive type into their corresponding wrapper classes. With autoboxing the compiler automatically converts primitive types into wrapper classes.

Ex: `int a = 10;`

`Integer b = a; //autoboxing`