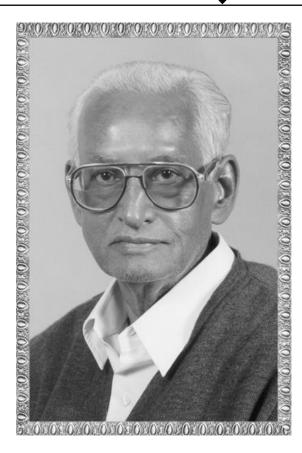
(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645



Lt. Shree Chimanbhai Shukla

B.Sc.I.T. SEM-5. J2EE

Shree H.N.Shukla College Street No. 2, Vaishali Nagar, Nr. Amrapali Under Bridge, Raiya Road, Rajkot. Ph. (0281)2440478, 2472590



Shree H.N.Shukla College Street No. 3, Vaishali Nagar, Nr. Amrapali Under Bridge, Raiya Road, Rajkot. Ph. (0281)2471645

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Chapter-1: Distrubuted Computing using RMI, Database Programming with JDBC

Distributed Computing Using RMI (Remote Method Invocation)

- Introduction to RMI
- RMI Architecture
- Stubs and Skeleton

RMI Definition:

RMI stands for Remote Method Invocation.

It is a mechanism that allows an object residing in one system (JVM) to access or invoke an object running on another JVM.

RMI is used to build distributed applications; it provides remote communication between Java programs. It is provided in the package java.rmi.

"RMI technology allows the client to communicate with Remote object and object to object communication between different JVM."

In Short RMI provides Remote Communication between programs written in java programming language.

Introduction to RMI:-

RMI is the best example of distributed module.

Distributed computing means a program that data process, and enter actual computation and spread over the network.

RMI is the best example of distributed system over TCP/IP. Object to object communication will be possible between different JVM, using RMI.

Overview of RMI Application:

RMI Application is divided into two parts Client and Server.

In an RMI application, we write two programs, a server program (resides on the server) and a client program (resides on the client).

Inside the server program, a remote object is created and reference of that object is made available for the client (using the registry).

The client program requests the remote objects on the server and tries to invoke its methods.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT. (AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

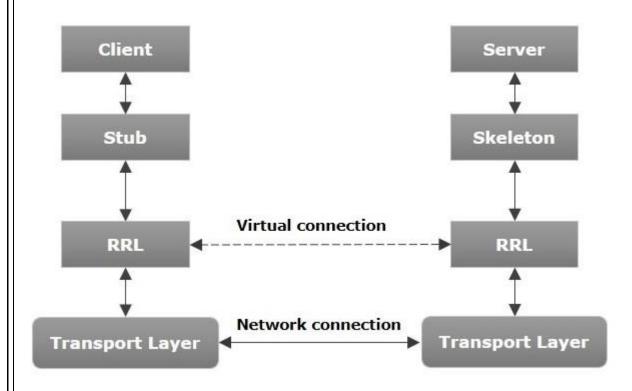
Server program creates some remote object and makes the references to them and waits for the client request to invoke methods on the Remote objects.

A client application get a remote reference to one or more Remote object from the server and invoke the methods on them. RMI provides this mechanism by which server and client communication and pass information back and forth.

Distributed Object Application. Has to include three things:

- 1.Locate Remote Object
- 2. Communicate with remote objects
- 3. Load class definition for object that are pass around

RMI Architecture:-



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

RMI Architecture defines that how the client request to the server for the remote objects and how the server processes for that request. It also defines the implementation of remote class on a remote JVM. It is divided into four layers.

- 1. Application Layer
- 2. Stub / Skeleton Layer/proxy Layer
- 3. Remote Reference Layer
- 4. Transport Layer

Each layer can perform specific task/function like establishing the connection, assemble and disassembled the parameter and transmiting the object.

All layers are independent from each other and they can be used based on client request for service as shown in figure.

Garbage Collection is advantage of Java language and RMI also supports Garbage Collection.

(1) Application Layer

Application Layer consists of client side and server side java programs with remote objects.

Here high level calls are made in order to import and export remote objects.

Client can access remote methods through an interface that extends java.rmi.Remote. on the client side a client simply request a remote object from either registry or remote object whose references has already been obtained.

Once the methods described in remote interface have been implemented, the object must be exported using UnicastRemoteObject class of the java.rmi.server package.

The application will register itself with registry and it is used by client to obtain references of the objects.

Name of the second seco

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

(2) Proxy layer:

Stub/skeleton layer is responsible for listening to the remote methods calls made by client and redirecting them to the server.

This layer consists of stub and a skeleton layer.

The stub and skeleton are created using RMI Compiler(RMIC).

Stub is client side proxy representing the remote objects and communicates the method invocation to the remote object through a skeleton that is implemented on the server. Skeleton is server side proxy that communicates with the stub. it makes call to remote objects. So it is also know as proxy layer.

(3) The Remote Reference layer ::=>

It is an interface between proxy layer and Transport layer.

It handles actual communication protocol.

RRL interpretes the references made by the client to the remote object on the server.

RRL present on clients and server. RRL on client receive requests for methods from stub, then request is transferred to the RRL on server side.

RRL used to:

- Handle replicated object. once the feature is included into RMI system, the replicated object will allow simple dispatch to other programs.
- It is responsible for establishing persistence connection or recovery of lost connections.

(4) The Transport Layer ::=>

The Transport layer takes path of the communication or layer between client & server. It receives a request from client side RRL and establishes a connection with the server through server side RRL.

Transport layer has following responsibilities:

- It is responsible for handling actual machine to machine communication and default communication takes place through TCP/IP.
- It creates stream that is accessed by RRL to send and receive data from other machines.
- It sets up the connection to the remote machines.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

- It manages the connections.
- It moniters the connections. To make sure that they are live.
- It listens for connections from the machines.

STUB AND SKELETON

STUB-Client side proxy.

It presents the same remote interface as the object of the server.

It works with JVM on client machine to serialize any arguments to a remote method call and sends this information to the server machine.

The stub receives any results from remote method and returns it to the client.

SKELETON Server Side proxy.

It receives the remote method call and any associated arguments. It works with JVM and RMI system on server to deserialize any arguments fro remote method call.

It invoke the appropriate method in the server.

It receives any return value from this method call and works with with JVM and RI system on server to serialize the return value and send information back to the client.

RMI Package:=>

The Java.rmi.* is imported in all the program in order to communicate with server .

The Naming Class of RMI package has declared various methods.

1.bind () :- It is used to binds a specified remote name to remote object. Having two parameter (1)name of object and(2) object itself

Ex:- Naming.bind("Hello", greeting);

2. unbind() It removes the binding object with remote object. Having single parameter.

Ex:- Naming.unbind("Hello");

3. rebind() it is used to rebind name with remote object. Having two parameter (1)string message and(2)new object name

Ex:- Naming.rebind("Hello",msg);

4. lookup()- It returns name of the remote object specified by the RMI URL.

Ex:- Naming.Lookup(rmi://hostname/objectname);

SE SHIP A GROUP OF THE SECOND SECOND

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

5. list()- It returns the string array with name of objects bound in the registry .Having one parameter.

Ex:- Naming.list(URL);

Steps for Developing RMI Application

- 1) Create an interface. //(in the practical example rmii.java).
- 2) Create a class that implements the interface. //(in the practical example rmiimpl.java).
- 3) Create a server that creates an instance of this class
- 4) Create a client that connects to the server object using Naming.lookup() //(in the practical example rmiclient.java)
- 5) Compile these classes.
- 6) Run the RMI interface compiler on the .class file of the implementation class //(in the practical example rmiimpl.java").
- 7) Start the RMI registry ("start rmiregistry").
- 8) Start the server class ("start java rmiserver").
- 9) Run the client program ("java rmiclient").



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

<u>Unit - 2 JAVA DATABASE CONNECTIVITY (JDBC)</u>

Introduction and Need for JDBC:-

JDBC Stands for the Java Database Connectivity.

The Role of the JDBC is to most important with respect to java database application.

If you are familiar with SQL and relation Database the structure of JDBC API is similar and easy to understand.

JDBC enables java application & applet common to the database.

The main idea behind this is we have been able to have different application to the different database. The java.sql.package gives us to facility to connect to every database.

JDBC-ODBC:

ODBC stands for the Open Database connectivity, where we want to our regular database like access, oracle, sql, visual FoxPro etc. we have to use the JDBC ODBC driver, ODBC is standard which is provided by Microsoft and we known that Microsoft is widely used in computer technology so sun Microsystems has accepted ODBC as Universal driver and java has developed sun.jdbc.odbc package for this interface.

To connect JDBC by using ODBC we have to develop one ODBC driver by using the Control panel 32-bit ODBC program from your window application.

The Steps for the Creating DSN are as follows.

- → Double click on 32-ODBC Program
- →Click on New Button
- →Give the DSN Name
- → Select appropriate Driver
- →Click on to Complete Preparation

JDBC V/S ODBC:-

- ODBC can not be directly used with Java because it uses C interface. Calls from Java to native C code have number of drawbacks in the security, implementation, robustness and automatic portability of applications.
- ODBC make use of Pointer which have been totally removed from java.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



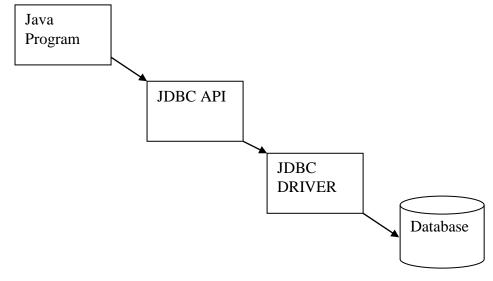
2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

- ODBC mixes simple advanced features together and has complex options for simple queries. But JDBC is designed to keep things simple while allowing advanced capabilities when required.
- **JDBC API** is a natural Java interface and is built on ODBC so JDBC retains some of the basic features of ODBC like SQL call level interface.

JDBC API

JDBC API Is collection of methods, classes and interfaces that enable java applications to communicate with database. For this database should be installed on the computer and RDBMS should provide a driver.

The relations between a java program, JDBC api, the JDBC driver and database is shown in figure.



Using JDBC API a java application can perform insert, delete, update and select rows in a database. It can also retrieve the data from the database and show it in a proper formate to the user according to the SQL query.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

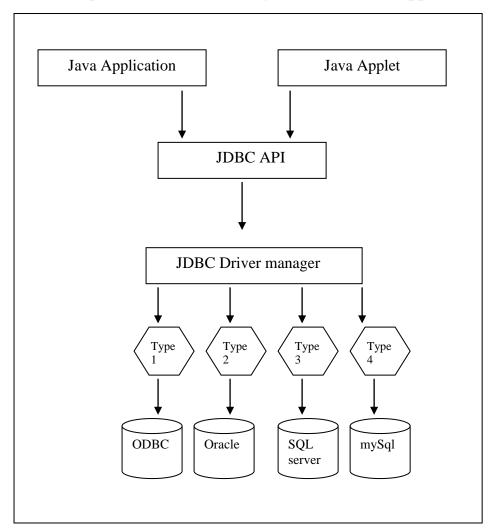
2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

JDBC Architecture:

JDBC Architecture describes the interaction of JDBC API with java application and java applet.

JDBC API consists of several call level interfaces for interaction with JDBC Driver Manager and JDBC driver for defining database.

JDBC API is responsible for transferring data between an application and a database.



SHIP ON WHAT

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

The JDBC architecture is divided into two parts: JDBC Driver types and JDBC API

JDBC Data types:-

The JDBC driver converts the Java data type to the appropriate JDBC type before sending it to the database. It uses a default mapping for most data types. For example, a Java int is converted to an SQL INTEGER. Default mappings were created to provide consistency between drivers.

The following table summarizes the default JDBC data type that the Java data type is converted to when you call the setXXX() method of the PreparedStatement or CallableStatement object or the ResultSet.updateXXX() method.

SQL	JDBC/Java	setXXX	updateXXX
VARCHAR	java.lang.String	setString	updateString
CHAR	java.lang.String	setString	updateString
LONGVARCHAR	java.lang.String	setString	updateString
BIT	boolean	setBoolean	updateBoolean
NUMERIC	java.math.BigDecimal	setBigDecimal	updateBigDecimal
TINYINT	byte	setByte	updateByte
SMALLINT	short	setShort	updateShort
INTEGER	int	setInt	updateInt
BIGINT	long	setLong	updateLong
REAL	float	setFloat	updateFloat
FLOAT	float	setFloat	updateFloat
DOUBLE	double	setDouble	updateDouble
VARBINARY	byte[]	setBytes	updateBytes
BINARY	byte[]	setBytes	updateBytes
DATE	java.sql.Date	setDate	updateDate
TIME	java.sql.Time	setTime	updateTime
TIMESTAMP	java.sql.Timestamp	setTimestamp	updateTimestamp

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

CLOB	java.sql.Clob	setClob	updateClob
BLOB	java.sql.Blob	setBlob	updateBlob
ARRAY	java.sql.Array	setARRAY	updateARRAY
REF	java.sql.Ref	SetRef	updateRef
STRUCT	java.sql.Struct	SetStruct	updateStruct

JDBC 3.0 has enhanced support for BLOB, CLOB, ARRAY, and REF data types.

The setXXX() and updateXXX() methods enable you to convert specific Java types to specific JDBC data types. The methods, setObject() and updateObject(), enable you to map almost any Java type to a JDBC data type.

ResultSet object provides corresponding getXXX() method for each data type to retrieve column value. Each method can be used with column name or by its ordinal position.

Date & Time Data Types:

The java.sql.Date class maps to the SQL DATE type, and the java.sql.Time and java.sql.Timestamp classes map to the SQL TIME and SQL TIMESTAMP data types, respectively.

Handling NULL Values:

SQL's use of NULL values and Java's use of null are different concepts. So how do you handle SQL NULL values in Java? There are three strategy you can use:

Use wrapper classes for primitive data types, and use the ResultSet object's wasNull() method to test whether the wrapper class variable that received the value returned by the getXXX() method should be set to null.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

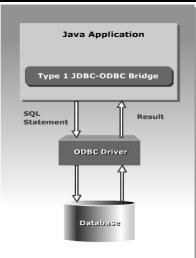


2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

JDBC Database Drivers:

JDBC Drivers specification divide JDBC drivers into four types of groups.

Type -1 : JDBC to ODBC Driver



The JDBC-ODBC bridge driver converts all JDBC calls into ODBC calls and sends them to ODBC Driver.

The ODBC driver then converts the call to the database server. For example: MS ACCESS do not provide JDBC Driver driver. Instead they provide ODBC Driver. One can access such database by using ODBC-JDBC Bridge.

Advantages:

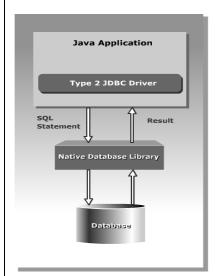
It allows access to any database.

Disadvantages:

- This driver not fully written in java so not portable.

- Performance is slow compare to all drivers because JDBC calls are converted to ODBC calls and ODBC calls are converted to database specific calls.same step in reverse order when query is obtained.
- Client system requires ODBC installation to use the drivers.
- -Not good for web application or for large amount of database based application.

(2) Type-2: NATIVE API Driver



The Java/Native Code driver Communicates directly with database server, therefore the vendor database library needs to be loaded on each client computer. It converts JDBC calls into database specific calls for databases. For example:Oracle will have native API

Advantages:

They offer better performance than JDBC-ODBC Bridge. Because as the layers of communication are less than type-1 driver and also uses native api which is database specific.

Disadvantages:

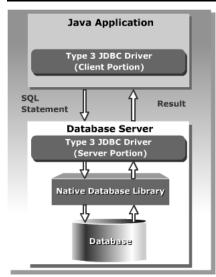
(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

- Native API must be installed in client system and hence type-2 driver cannot be used for internet.
- It is not written in java which forms a portability issues.
- If we change the database we have to change Native Api as it is specific to database.
- Mostly outdated because it is not thread safe.
- It is not suitable for distributed application.

(3)Type-3 JDBC Drivers/NET Protocol Driver:



This drivers follows a 3 tired approach. In 3 tired approach JDBC requests are passed to a middle tier server, the middle Tier server then translates request to the database specific native connectivity interface and forward the request to the database server.

It is used to connect client application/applet to database server over TCP/IP connection. The presence of any vendor database library on client computer is not required because type-3 driver is servere based.

Advantages:

- It is sever based so no need for any vendor database library to be present on client machine.
- This is fully written in java and hence portable, so it is suitable for web application.
- It provides portability, performance, security and scalability.
- Net protocol can be designed to make client JDBC driver very small and fast to load.
- It provides support for feature such as caching,load balancing,advanced system administration.
- Very flexible and allows access to multiple database using one driver
- They are efficient among all drivers.

Disadvantages:

Type-3 driver requires database specific coding to be done on middle tier and it requires additional server for that. Additionally Traversing the recordset may take longer because the data comes-through the backend server or backend database.

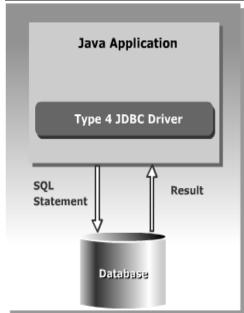


(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

(4)Type-4 JDBC Driver/Native Protocol



Type-4 driver are completely written in java to achieve platform independency. It converts JDBC calls into vendor specific DBMS protocol. Therefore client application can communicate directly with the database server when type-4 driver is used.

Type-4 driver is better than other driver because type-4 driver do not have to translate database request to ODBC calls or a native connectivity interface or pass on to another server.

Advantages:

- They are written in java to achieve platform independence. It reduces deployment administration issues. It is most suitable for web application.
- Number of layers are very less Type-4 do not translate database request to ODBC or native connectivity interface or to pass to the server so performance is good.
- You do not need to install special software in the client machine. These driver can be downloaded dynamically.

Disadvantages:

- Different driver is needed for different database.

Processing SQL Statements with JDBC

In general, to process any SQL statement with JDBC, you follow these steps:

- Establishing a connection.
- Create a statement.
- Execute the query.
- Process the ResultSet object.
- Close the connection.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Establishing Connections

First, establish a connection with the data source you want to use. A data source can be a DBMS, a legacy file system, or some other source of data with a corresponding JDBC driver. This connection is represented by a Connection object.

Creating Statements

A Statement is an interface that represents a SQL statement. You execute Statement objects, and they generate ResultSet objects, which is a table of data representing a database result set. You need a Connection object to create a Statement object.

There are three different kinds of statements:

- **Statement**: Used to implement simple SQL statements with no parameters.
- **PreparedStatement:** (Extends Statement.) Used for precompiling SQL statements that might contain input parameters.
- CallableStatement: (Extends PreparedStatement.) Used to execute stored procedures that may contain both input and output parameters.

Executing Queries

To execute a query, call an execute method from Statement such as the following:

- **execute**: Returns true if the first object that the query returns is a ResultSet object.
- **executeQuery**: Returns one ResultSet object.
- **executeUpdate**: Returns an integer representing the number of rows affected by the SQL statement. Use this method if you are using INSERT, DELETE, or UPDATE SQL statements.

Processing ResultSet Objects

You access the data in a ResultSet object through a cursor. Note that this cursor is not a database cursor. This cursor is a pointer that points to one row of data in theResultSet object.

Closing Connections

When you are finished using a Statement, call the method Statement.close to immediately release the resources it is using.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

JDBC Exception Handling:-

Exception handling allows you to handle exceptional conditions such as program-defined errors in a controlled fashion.

When an exception condition occurs, an exception is thrown. The term thrown means that current program execution stops, and control is redirected to the nearest applicable catch block. If no applicable catch block exists, then the program's execution ends.

JDBC Exception handling is very similar to Java Exception handling but for JDBC, the most common exception you'll deal with is java.sql.SQLException.

SQLException Methods:

SQLException can occur both in the driver and the database.

When such an exception occurs, an object of type SQLException will be passed to the catch clause.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 † 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

The passed SQLException object has the following methods available for retrieving additional information about the exception:

Method	Description	
getErrorCode()	Gets the error number associated with the exception.	
getMessage()	Gets the JDBC driver's error message for an error handled by the driver or gets the Oracle error number and message for a database error.	
getSQLState()	Gets the XOPEN SQLstate string. For a JDBC driver error, no useful information is returned from this method. For a database error, the five-digit XOPEN SQLstate code is returned. This method can return null.	
getNextException()	Gets the next Exception object in the exception chain.	
<pre>printStackTrace()</pre>	Prints the current exception, or throwable, and its backtrace to a standard error stream.	
<pre>printStackTrace(PrintStream s)</pre>	Prints this throwable and its backtrace to the print stream you specify.	
<pre>printStackTrace(PrintWriter w)</pre>	Prints this throwable and its backtrace to the print writer you specify.	

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

By utilizing the information available from the Exception object, you can catch an exception and continue your program appropriately. Here is the general form of a try block:

```
try {
    // Your risky code goes between these curly braces!!!
}
catch(Exception ex) {
    // Your exception handling code goes between these
    // curly braces, similar to the exception clause
    // in a PL/SQL block.
}
finally {
    // Your must-always-be-executed code goes between these
    // curly braces. Like closing database connection.
}
```

JDBC API FOR DATABASE CONNECTIVITY

1) Connection Interface:

This defines connection to different database. This provides following methods.

(1) createStatement()

It is used to create a Statement object for sending SQL Statements to the database.

(2) prepareStatement()

It is used to create a Prepared Statement object for sending SQL request to database.

(3) prepareCall()

It is used to create a Callable Statement object for calling stored procedure of database.

(4) close()

It is used to release connection.they do not wait for it to release it automatically.

(5) commit()

It is used to save all the state of the database. Which are changed by transaction.

(6) rollback()

It is used to cancel the effect of current running transaction.

(7) setAutoCommit ()

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

It is used to do auto commit all the transaction. if we set auto commit with connection all SQL statements executed and committed together as individual transaction.

(8) getAutoCommit ()

It is used to get current committed state.

(9) isClosed()

It is used to check whether the connection has been closed or not.

(10) getMetaData()

It is used to get metadata(data about data)related to current connection with database. it provides information like table driver name, driver type, its version and so on.

2. DriverManager Class

DriverManager class belongs to java.sql package. It consists of static method to manage JDBC Drivers. Each and every driver must register with DriverManager class. There are many JDBC Drivers used for different JDBC servers.

DriverManager class has one method getConnection() to establish connection with different database server.

Statement:-

Statement object is used to execute SQL statements and obtain the result produced after excuting the SQL statements. it is having following methods.

(1) executeQuery ():

It is used to fetch records from database using select query. It passes SQL statement as a parameter.

(2) executeUpdate ():

It is used to perform insert,update,delete operation on record of database.It returns integer value indicating no. of records updated in database.It passes SQL as a parameter.

(3)execute()

The execute() of the Statement is used when they returns Multiple result.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar
Nr. Amrapali Under Bridge
Nr. Amrapali Under Bridge

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Prepared Statement:-

Prepared Statement interface is derived from Statement interface. It uses a template to create SQL request and uses a Prepared Statement object to send precompiled SQL statements with one or more parameters. It is more convenient to use a Prepared Statement object for sending SQL statements to database.

If you want to execute SQL statements many times Prepared Statement object reduces execution time in comparision to other Statement object.

Main feature is it is used to execute precompiled statements.

Execute methods:

(1) executeQuery ():

It is used to fetch records from database using select query. It passes SQL statement as a parameter. It returns single Resultset Objects that contains rows, columns and metadata that represent data requested by query. Example:

String sql="select * from tablename where fieldname=?"; setString(1,"C01");
PreparedStatement pst =con.prepareStatement ();;
ResultSet rs=pst.executeQuery();

(2) executeUpdate ():

It is used to perform insert,update,delete operation on record of database. It returns integer value indicating no. of records updated in database. It passes SQL as a parameter. Example:

Statement st = con.createStatement (); Int n=st.executeUpdate("insert into tablename values (val1,val2,.....,valn");

(3) execute()

It is used to execute SQL statements which returns multiple results. It returns the Boolean indicating that value if it returns true then it defines that next result is ResultSet object and if



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

it returns false then it defines that number of records updated or no more results is there. It passes any sql statements as parameter.

Callable Statement

The callable statement is useful to called stored procedure from within a J2EE object. Stored procedure is a block of code and is identified by a unique name. The type and style of code depend on dbms vendor and can be return in SQL. The stored procedure is invoked by giving name of stored procedure.

The callable statement uses three types of parameters when calling a stored procedure. These parameters are IN, OUT & INOUT.

The IN parameter contains any data that needs to be passed to the stored procedure and whose value is assign using the setXxx().

The OUT parameter contains the value return by the stored procedure if any OUT parameter must be registered using the register OUT parameter() and then his later retrieve by J2EE component by using getXxx().

The INOUT parameter is a single parameter that is use for both that is to pass information to stored procedure and to retrieve information from stored procedure.

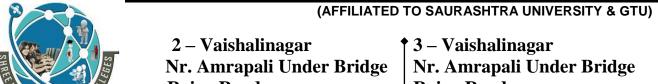
The stored procedure requires one parameter that is represented by "?".

Next the preparedCall() of connection object is called and it pass the query.

The execute() of callable statement will be useful to execute the query. After the stored procedure is executed, getString() is called to return the value of specified parameter of stored procedure.

ResultSet:-

It is used to access the data of the table from database. Resultset object stores the data of the table by executing the query. It also maintains the cursor position for navigation of the data. Cursor can move on first(), last(), previous(), next() from the current row position.



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

It also fetch data from the table using getXXX() method depends on columnType(getString(),getInt()etc.). It can fetch data either using getXXX(). Index number always starts with one.

There are six methods of ResultSet object that are used to position the virtual cursor. They are first(), last(), previous(), next(), absolute(), relative(), and getRow().

Fields of resultset interface:

- (1) **TYPE_SCROLL_INSENSITIVE**: It defines that the cursor can scroll but can not be modified or updated.
- (2) TYPE_SCROLL_SENSITIVE: It defines that the cursor can scroll and also be modified or updated.
- (3) **TYPE_FORWARD_ONLY**: It defines that the cursor from the current row move forward only.
- (4) CONCURE_READ_ONLY: It defines that cursor can scroll and also can be modified or updated.

Methods of ResultSet Interface:

- (1) **getString**(): It is used to fetch the string type of records from table. It returns the String object.
- (2) **getInt()**: It is used to fetch the integer type of records from table. It returns the String object.
- (3) **getBoolean**(): It is used to fetch the boolean type of records from table. It returns either true/false.
- (4) **getDouble**(): It is used to fetch the decimal type of records from table. It returns the double value.
- (5) getFloat(): It is used to fetch the float type of records from table. It returns the String object.
- (6) **getDate**() : It is used to fetch the date type of records from table.
- (7) **getLong**() : It is used to fetch the long type of records from table.
- (8) **getShort**(): It is used to fetch the short type of records from table.
- (9) **getByte**() : It is used to fetch the string type of records from table.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

(10) getBlob(): It is used to fetch binary large object from table that can be image file, audio file, video file.

Example: ResultSet rs=st.executeQuery("select * from emp");

Int id=re.getInt(1);

String name=rs.getString(2);

OR

Example: ResultSet rs=st.executeQuery("select * from emp");

Int id=re.getInt("eno");

String name=rs.getString("ename");

Navigation Method:

• **first():** It is used to move the cursor position on the first record of the table.

• last():It is used to move the cursor position on the last record of the table.

• next():It is used to move the cursor position on the next record of the table.

• **previous():**It is used to move the cursor position on the previous record of the table.

• afterLast():It is used to move the cursor on after the last row.

• **beforeFirst():**It is used to move the cursor on before the first row.

• **relative(int row):** It is used to move the cursor on relative no. of roes.

• **absolute(int rows):** It is used to move the cursor on absolute row.

OTHER JDBC APIs:-

Meta Data :-

Metadata is data about data. There are two types of metadata interface are available (1)ResultsetMetadata and (2)Database Metadata.

• ResultSetMataData:

This interface is used to obtain information about type and properties of the column in a ResultSetMataData is used to store following information of Resultset object:

- what is the number of columns in ResultSet?
- what is the name of columns?
- what is the type of columns in?
- what is the maximum size of columns?
- can you put a null in this column?

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Method	Description
getColumnCount()	Returns the number of columns contained in the
	ResultSet
getColumnName(int n)	Returns the name of the column specified by the
	column number
getColumnType(int n)	Returns the data type of the column specified by the
	column number.
isNullable(int column);	Indicates the null ability of values in the designated
	columns.

• DatabaseMetaData Interface:

This interface describes the database as whole. Many methods of DatabaseMetaData interface returns list f information in the forof Resultset objects.It provides many methods that represents comprehensive information of the database.

- What tables are available?
- What is your username known to the database?
- Is the database in read only moe?
- What is the name of product?
- What is the name of driver?
- What is the version of driver?

Method	Description	
getDatabaseProductname()	Returns the product name of the database	
getUserName()	Returns the username	
getURL()	Returns the URL of the database	
getSchemas()	Returns all the schema names available in this	
	database.	
getPrimaryKeys()	Returns primary keys	
getProcedures()	Returns stored procedure names	
getTables()	Returns names of tables in the database.	
getDriverName()	Retrives the name of JDBC Driver	
getDriverVersion()	Retrives the version of JDBC Driver	
isReadOnly()	Retrives whether this database is read only.	



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Connecting with Databases:-

After you've installed the appropriate driver, it's time to establish a database connection using JDBC.

The programming involved to establish a JDBC connection is fairly simple. Here are these simple steps:

- Import JDBC Packages
- Register JDBC Driver
- Database URL Formulation
- Create Connection Object
- Closing JDBC connectionS

(1) Import JDBC Packages:

The **Import** statements tell the Java compiler where to find the classes you reference in your code and are placed at the very beginning of your source code.

(2) Register JDBC Driver:

You must register the your driver in your program before you use it.

Approach (I) - Class.forName():

The most common approach to register a driver is to use Java's **Class.forName()** method to dynamically load the driver's class file into memory, which automatically registers it.

Approach (II) - DriverManager.registerDriver():

The second approach you can use to register a driver is to use the static **DriverManager.registerDriver()** method.

(3) Database URL Formulation:

After you've loaded the driver, you can establish a connection using the **Driver Manager.get Connection**() method.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Following table lists down popular JDBC driver names and database URL.

RDBMS	JDBC driver name	URL format
MySQL	com.mysql.jdbc.Driver	jdbc:mysql://hostname/ databaseName
ORACLE	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin: @hostname:port Number:databaseName
DB2	COM.ibm.db2.jdbc.net.DB2Driver	jdbc:db2:hostname:port Number/databaseName
Sybase	com.sybase.jdbc.SybDriver	jdbc:sybase:Tds:hostname: port Number/databaseName

(4) Create Connection Object:

The most commonly used form of getConnection() requires you to pass a database URL, a username, and a password.

(5) Closing JDBC connections:

At the end of your JDBC program, it is required explicitly close all the connections to the database to end each database session. However, if you forget, Java's garbage collector will close the connection.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Chapter - 2 Servlet Programming

Introduction of Servlet:-

A *servlet* is a small program that runs on a server.

Java *Servlets* are server-side Java program modules that process and answer client requests and implement the *servlet* interface.

There are two Java *Servlet* types: Basic and HTTP.

primary reason for development of servlet is limitation of html and CGI script.

Because with technology and usage of www growth, people's demand extends from static pages to dynamic pages.

A servlet is similar to the script except that the servlet runs under the JVM on the server. So it is safe and portable and it does not require the support for the java on a browser.

Servlet can be used for any number of web-related application for example developing an e-commerce web site becomes one of the most common usage for the java servlet.

Benefits of Servlets:

1. Performance:

Its performance is better than CGI because it is not necessary to create the separate process to handle each HTTP request, it executes within the address space of web server.

2. Platform Independence:

A number of web-server from different vendor supports Servlet API. It is written in Java so program developed for this API can be run on any environment without re-compilation.

3. Security:

Servlet run under the JVM and it is secured. It is also using java security manager. it is server side component, so it inherits the security provided by the web server.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

4. Efficient Servlet Initialization Code:

It is only executed for the first time when the web server loads it. Once the server has been loaded, it is only a matter of calling the service to handle the new request.

5. Persistent:

Servlets can maintain the state between request. Once the servlet has been loaded, it takes its state in the memory until the last request will arrive that means servlet will improve the performance of applications.

6. Portable:

Servlets are developed using java therefore they are portable.

7. Robust:

Servlets are developed with the jdbc. So they provide maximum robust solution with the help of java's well defined exception hierarchy, garbage collection, network support, file support etc.

8. Extensible:

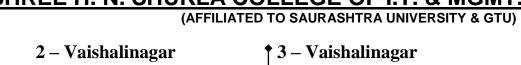
Servlets are developed in pure OOP like java. So that can be extended & polymorphed into new object which sets your needs.

9. Flexible:

Servlets are quite flexible because it can be added to a static page using the servlet tag <servlet>.

10. Secure:

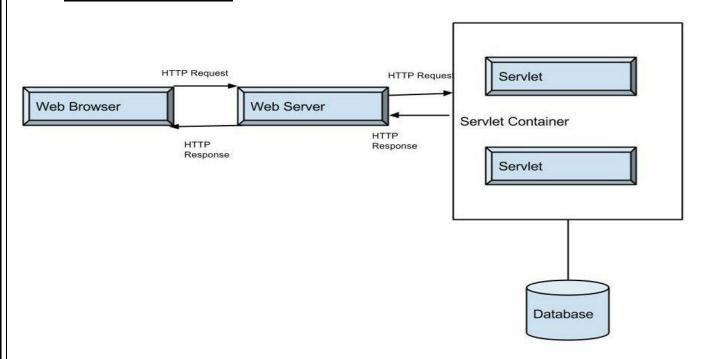
Servlets are run on server side & gives us the facility & security internally provided by the web servlet.



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Servlet Architecture:



Servlet architecture is used for build up an application and divided it to three tier mode.

First tier:- it include java enabled browser.

Second tier:- it include servlets which runs the application using protocol.

Third tier:- it include data related interface, using connection driver you can access the data.

There are two packages that make up the servlet architecture. They are as follows:

(1) javax.servlet package (2) javax.servlet.http package

The javax.servlet package contains the generic interface and classes that are implemented & extended by all servlets.

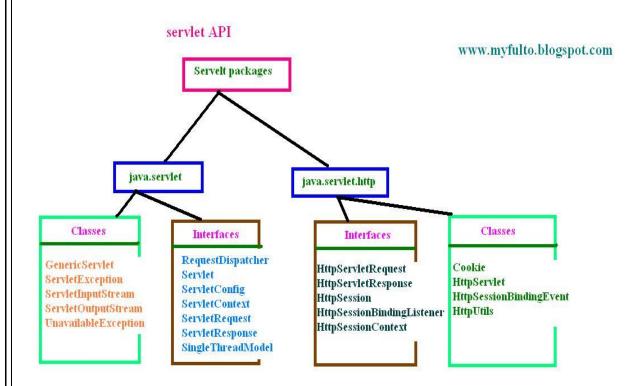
The javax.servlet.http package contains the classes that are extended while creating the http specific servlet.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

73 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645



javax.servlet package having following classes and interfaces.

- 1) Servlet Interface: It is having methods which define life cycle of servlet.
- 2) ServletConfig Interface: It provides the basic or initialization parameters of the servlet.
- 3) ServletContext Interface: It provides the runtime environment to the servlet.It also logs the event using log().
- 4) GenericServlet Class: It implements the Servlet, ServletConfig, Serialization interface.
- 5) ServletRequest Interface: It is used to read the client request.
- 6) ServletResponse Interface: It is used to write the response data.
- 7) ServletException Class: It defines the servlet occurred error.
- 8) UnavailableException Class: It defines that servlet is temporarily or permanently not available.
- 9) ServletInputStream Class: It provides input stream o read the data from the client request.
- 10) ServletInputStream Class:It provides output stream to write the data as request to client.
- 11) SingleThreadModel interface: It provides the mechanism to make servlet thread safe.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

javax.servlet.http package having following classes and interfaces.

- 1) HttpServletRequest Interface: It allows the servlet to read data from HTTP request.
- 2) HttpServletResponse Interface: It allows the servlet to write data to an HTTP response.
- 3) HttpServlet Class: It provides the methods for handling HTTP request and HTTP response.
- 4) Cookie Class: It used to store the state information at client side or on the client machine.
- 5) HttpSession Interface: It used to create session and provides the way of reading and writing the session and accessing the information related to session.
- 6) HttpSessionBindingListener interface: It informs the object that it is bound or unbound to the session
- 7) HttpSessionEvent class: It wraps up the session chage events.
- 8) HttpSessionBindingEvent Class: It defines that when session should be bound or unbound with the object value and its attribute.

To implement servlet in the program there are four ways. we can use anyone from following classes or interface to create servlet program.

- Servlet Interface
- GenericServlet Class
- HttpServlet Class
- SingleThreadModel Interface

Servlet Interface

All the user defined servlet should inherit the servlet interface.

Its having following methods.

(1) init()

It is called by servlet container to initialize the servlet itself. It is called only once when servlet engine loads the servlet. It finishes its work before service() starts for request and response.

public abstract void init(ServletConfig config) throws ServletException

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

(2) service()

It used to read client request and write the response data to the client. It passes two parameters first the object of ServletRequest interface and second is the object of ServletResponse interface.

Public abstract void service (ServletRequest req, ServletResponse res) throw ServletException, IOException

(3) destroy()

This method is called by servlet Container automatically when objects or resources like memory, thread etc need to be destroyed or servlet stops it execution.

Public abstract void destroy ()

(4) getServletConfig()

It returns the ServletConfig object which contains initialization parameter and startup configuration of servlet.It passed as a parameter in the init().

Public abstract ServletConfig getServletConfig()

(5) getServletInfo()

It returns the basic information related to the servlet such as author name, version, etc.

Public abstract void getServletInfo()

Generic Servlet Class

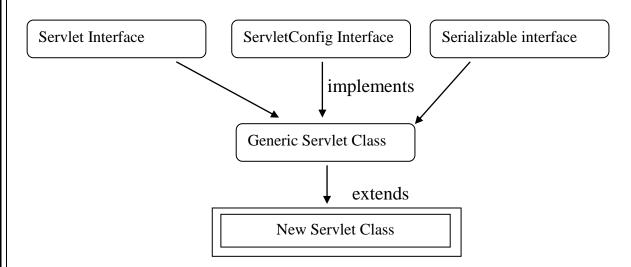
It belongs to the javax.servlet package. It is extended to provide the protocol based things like FTP,SMTP protocol but it is not having a HTTP protocol facility .So most probably with web application we need to extend the HttpServlet Class.

Generic servlet class provides the implementation of servlet interface because in most of the classes they are required to use service() to handle request and response.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645



Methods:-

(1)init()

It called by servlet container in addition if we override init()with servlet we need to write super.init(config) to provide reference of config object.

public abstract void init(ServletConfig config) throws ServletException

(2) service()

It provides the servlet request response facility to each and every servlet should override this method directly writing service () with the program or indirectly using doXXX () of HttpServlet class.

Public abstract void service (ServletRequest req, ServletResponse res) throw ServletException, IOException

(3) destroy ()

It is called by servlet container at the end of the execution. we can override this method in our subclass to do some clean up task, when servlet is taken out of service.

Public abstract void destroy ()



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

(4) log ()

log() is used to write the server log file. log() method writes the message to web container log file.

Public void log (String message)

(5) getInitParameter()

It is used to get the initialization parameter. If the parameter does not exists it returns null.

Public string getInitParameter (String name)

(6) getInitParameterNames()

It returns the name of initialization parameter names in the form of enumeration.

Public Enumeration getInitParameter Names ()

(7)getServletContext()

It returns the object of ServletContext.

Public ServletContext getServletContext ()

(8) getServletName()

It returns the name of the servlet.

Public String getServletName ()

SingleThreadModel

SingleThreadModel is implemented by SingleThreadModel interface. It ensures that servlet handle only one request at a time. This interface has no methods. If a servlet implements this interface you are guarantee by synchronizing access to single instance of



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

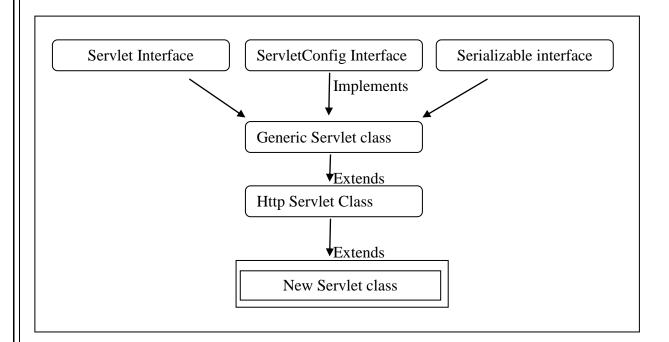
2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

servlet, or by maintaining a pool of servlet instances and dispatching each new request to a free servlet.

HTTP SERVLET CLASS

HttpServlet class extends GenericServlet Class. It is commonly used by programmers when developing servlet that receive and process HTTP requests, because it is having HTTP protocol functionality. Hierarchrchy of HttpServlet class is shown below.



Methods of HttpServlet Class are as follows:

(1) doGet()

doGet() is called by servlet via service() to handle an Http request. A GET request allows client to sent form data to server. With the get request the form data is attached to the end of the URL sent by the web browser to the server as a query string. The amount of form data that can be sent is limited to the maximum length of the URL. Basic format of doGet() is as follows.

public void doGet(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

(2) doPost()

doPost() is called by servlet via service() to handle an Http POST request. A POST request allows client to sent form data to server. With the post request, the form data is sent to the server separately instead of being appended to to the URL . This allows a large amount of data to be sent. Basic format of doPost() is as follows.

public void doPost(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException

(3) doDelete ()

doDelete() is called by servlet via service() to handle an Http DELETE request. A DELETE request allows client to remove a document or web page from the server. Basic format of doDelete() is as follows.

public void doDelete(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException

(4) doOption()

doOption() is called by servlet via service() to handle an Http OPTION request. An OPTION determines which Http method the server supports and sends the information back to the client by way of header.Basic format of doOption() is as follows.

public void doOption(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException

(5) doPut()

doPut() is called by servlet via service() to handle an Http PUT request. A PUT request allows client to sent place a file on the server and is conceptually similar to sending the file to the server via FTP. Basic format of doPut() is as follows.

public void doPut(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException

(6) doTrace()

doTrace() is called by servlet via service() to handle an Http TRACE request. It is useful

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

for debugging purpose. This method is rarely overridden. Basic format of doTrace() is as follows.

public void doTrace(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException

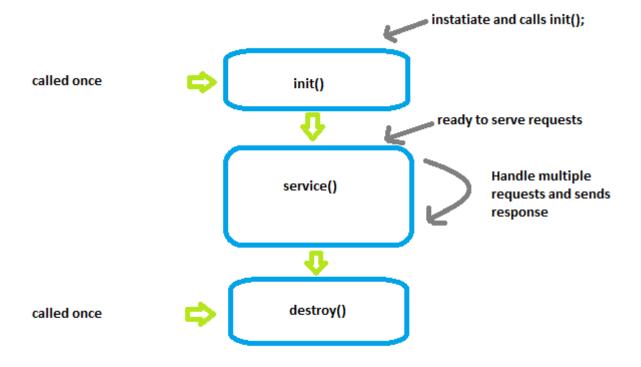
Servlets are the Java programs which run inside a Servlet Container.

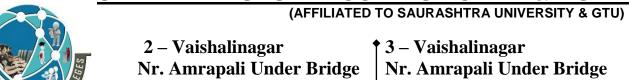
A Servlet Container is much like a Web Server which handles user requests and generates responses. It can also contain Java Servlets and JSP pages to generate dynamic response.

Servlet Container is responsible for loading and maintaining the lifecycle of the a Java Servlet. Servlet Container can be used standalone or more often used in conjunction with a Web server. Example of a Servlet Container is Tomcat and that of Web Server is Apache.

There are three methods used to maintain the lifecycle of servlet.

- 1) **init()**
- 2) service()
- 3) destroy()





2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

(1) init()

when the servlet is first created, its init() is invoked, so init is where you put one time setup code. It is called when the servlet Is first created, and not called again for each user request, so it is used for one time initialization, just as with the init() of applet.

This method is called once when the servlet is loaded into the servlet engine, before the servlet is asked to process its first request.

The init method has a ServletConfig parameter. The servlet can read its initialization arguments through the ServletConfig object. It may be usually defined in a configuration file.

(2) service()

Each time the server receives a request for a servlet, the server needs a new thread and calls service().

The service() checks the HTTP request type(GET,POST,PUT,DELETE etc) and calls doGet,doPut,doDelete etc.as appropriate.

(3) destroy()

The service may decide to remove a previously loaded servlet instance, perhaps because it is explicitly asked to do so by the server administrator or perhaps because the servlet is idle for a long time.

ServletException:-

The javax.servlet defines two exceptions,

- (1) First is ServletException which indicates that a servlet problem has been occured.
- (2) Second is UnavailableException which extends ServletException and indicates that servlet is unavailable.

(1)Servlet Exception

Servlet Exception indicates that a servlet problem has occurred. The general form of Servlet Exception class is:

public class Servlet Exception extends Exception



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

ServletException has following constructor.

- Public ServletException():Basic exception constructor
- Public ServletException(String msg):It creates a new servlet exception with a message.

(2) Unavailable Exception

Unavailable Exception extends ServletException. It defines exception that a servlet throws to indicate that it is permanently or temporarily unavailable.

When a servlet is permanently unavailable, something is wrong with it, and it can not handle request until some action is taken.

A servlet is temporarily unavailable if it can not handle request momentarily due to some system problem.

public class UnavailableException extends ServletException

UnavailableException has following constructor.

- Public UnavailableException():Basic exception constructor
- Public UnavailableException (String msg):It creates a new exception with a message that a servlet is permanently Unavailable.
- Public UnavailableException (String msg,int sec): It creates a new exception with a message indicating that a servlet is temporarily Unavailable and giving an estimate of how long it will be Unavailable.

It provides following methods:

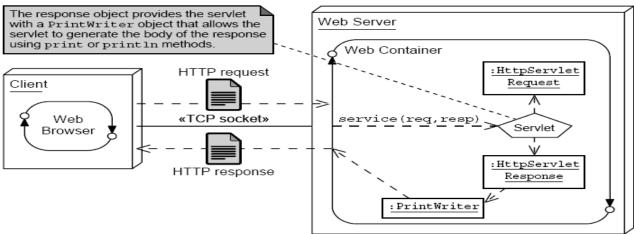
- I. getUnavailableSeconds (): It returns the number of seconds the servlet expects to be temporarily available.
- II. getServlet () : It returns the servlet that threw this exception or null if the servlet instance was not provided to the constructor.
- III. IsPermanent (): It indicates that the servlet is permanently unavailable or not.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

ServletRequest and ServletResponse ::=>



The ServletRequest interface is implemented by the server which enables the servlet to obtain the information about the client request.

ServletRequest Class:-

Methods	Description	
Void setAttribute(String name, Object	It stores an attribute in the request.	
obj)	Mostly used with Request Dispatcher.	
Object getAttribute(String name)	Returns the value of the attribute named	
	name.	
Enumeration getAttributNames()	It returns the array of the name of the attributes.	
G		
String getParameter(String prm)	Returns the value of the parameter	
	named prm.	
Enumeration getParameterNames()	Returns an enumeration of the parameter	
	names for this request.	
String[] getParameterValues(String	Returns an array containing values	
nm)	associated with the parameter specified	
	by nm.	
String getRemoteAddr()	Returns the String equivalent of the	
	client IP address.	
String getRemoteHost()	Returns the String equivalent of the	
	client host name.	



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 † 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

ServletResponse Class

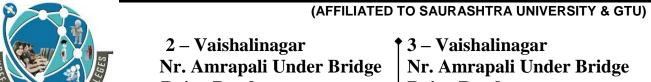
Methods			Description	
PrintWriter	getWriter()	throws	Returns a PrintWriter that can be used to	
IOException			write character data to the response. An	
			IllegalStateException is thrown if	
			getOutputStream () has already been	
			invoked for this request.	
<pre>void setContentType(String type)</pre>		oe)	Sets the content type for the response to	
			type.	

HttpServletRequest Class

Methods	Description
Enumeration getHeaders()	Returns all the values of the specified request
	headers as an Enumeration.
Cookies[] getCookies()	Used to obtain the array of cookie that are
	present in the request.
String getQueryString()	Returns the query string that is contained in
	the request URL.
HttpSession getSession()	Returns the current session associated with
	the request.

HttpServletResponse Class

Methods	Description
Void addCookie(Cookie cookie)	Adds cookie to specified HTTP response.
String encodeURL(String url)	It encodes the URL by including session id
	in it, if encoding is not needed returns URL
	unchanged.
void sendRedirect(String url)	Redirects the client to url.
throws IOException	
void setHeader(String name,	Used to set response header with given name
String value)	and value.
void setStatus(int sc)	Sets the status code for HTTP response
	SC_MOVED_TEMPORALILY,SC_OK



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 † 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Session Tracking Approaches

When a user make a page requests to the server, the server creates a temporary session to identify that user so when user goes to another page on the same site the server can recognize that user easily so a Session is temporary small unique connection between a server and the client enabling it to identify that user across multiple page request or visit to that site.

Http is a stateless protocol. Each request is independent of the previous one.

There are four ways to track the session:-

- URL rewriting
- Hidden Form Field
- Cookies
- Http Session

•

URL rewriting ::=>

To maintain the session state the URL rewriting is the best practice. URL rewriting is based on the idea of inserting a unique id(generated by server) in each url of the response from the server.

While generating the response to the first request, the server inserts this id in each url.when client submit such a request to one such url, the browser send this id back to the server. The server can identify the id with all requests.

<u>Advantage</u>: User remains unknown and they are universally supported.

<u>Disadvantage</u>: It is tedious to rewrite all the url and it only works with dynamically created documents.

Hidden form fields :-

The component can include in the html form a field that is not displayed on the form. This field is called hidden field. A hidden field is similar to the other fields of the form. The main thing is the value of hidden field will not be entered by user but it is assigned by the Component.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

The data available in the hidden field will be treated same as the other regular fields. The value of the hidden filed can be abstracted from the form and can be set by the browser as a parameter Example <input type = "hidden" name = "grno" value = "1">

Advantages:

It is universally supported and allow anonymous users.

Disadvantages:

- It only works for a sequence of dynamically generated forms.
- It breaks down with static documents, emailed documents, and bookmarked documents.
- There are no browser shut downs.

Cookies:-

Cookies are small bits of textual information that a Web server sends to a browser and that the browser later returns unchanged when visiting the same Web site or domain.

All modern browser can receive cookies from web servers and then send them back along with requests. All browser allow users to disable this functionality, which leads to browser not recognizing cookies. This is because cookies have a bad impression that they have sometimes been used togather information about consumer without their knowledge.

Identifying a user during an e-commerce session. Many on-line stores use a "shopping cart" in which the user selects an item, adds it to shopping cart, then continues shopping. Since the HTTP connection is closed after each page is sent, when the user selects a new item for his cart, how does the store know that he is the same user that put the previous item in his cart? Cookies are a good way of accomplishing this.

Avoiding username and password. Many large sites require you to register in order to use their services, but it is inconvenient to remember the username and password. Cookies are a good alternative for low-security sites. When a user registers, a cookie is sent with a unique user ID. When the client reconnects at a later date, the user ID is returned, the server looks it up, determines it belongs to a registered user, and doesn't require an explicit username and password.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Customizing a site. Many portal sites let you customize the look of the main page. They use cookies to remember what you wanted, so that you get that result initially next time. Focusing advertising. Cookies let the site remember which topics, interest certain users and show advertisements relevant to those interests.

The Servlet Cookie API

To send cookies to the client, a servlet would create one or more cookies with the appropriate names and values via new Cookie(name, value) and add the cookies to the response headers via response.addCookie(cookie).

To read incoming cookies, call request.getCookies(), which returns an array of Cookie objects.

Creating Cookies

A Cookie is created by calling the Cookie constructor, which takes two strings: the cookie name and the cookie value. Neither the name nor the value should contain whitespace or any of:

```
[\ ]\ (\ ) = , "/?@:;
```

Placing Cookies in the Response Headers

The cookie is added to the Set-Cookie response header by means of the addCookie method of HttpServletResponse. Here's an example:

```
Cookie userCookie = new Cookie("user", "hns");
response.addCookie(userCookie);
```

Reading all Cookies stored in Request object

```
Cookie c[] = request.getCookies();
if(c.length==0)
  out.println(" No Cookies in Your PC");

for(int i=0; i<c.length; i++)
{
  out.println("<br> Cookie-" + (i+1));
```

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

```
out.println("<br> Name : " + c[i].getName());
out.println("<br> Value : " + c[i].getValue());
}
```

Session API:

When we write any servlet, it is necessary for the developer to maintain sessions.

There are several methods and classes which are specifically designed to handle the session tracking on behalf of servlet provided by the Servlet API in otherwords we can say servlet has in built facility to manage the session.

HTTPSession interface:-

A public interface HttpSession is a member of javax.servlet.http package. It provides a way to identify a user across more than one page request or visit to a web site and to store information about that user.

The servlet container uses this interface to create a session between an HTTP client and an HTTP Server. The session persist for a specified time period across more—than one connection or page request from the user.

An HttpSession class was introduced in the Servlet API. Instances of this class can hold information for one user session between requests. You start a new session by requesting an HttpSession object from the HttpServletRequest in your doGet or doPost method:

HttpSession mysession = request.getSession(true);

This method takes a boolean argument. true means a new session shall be started if none exist, while false only returns an existing session.

An HttpSession can store any type of object. A typical example is a database connection allowing multiple requests to be part of the same database transaction, or information about purchased products in a shopping cart application so the user can add items to the cart while browsing through the site. To save an object in an HttpSession you use the putValue method:

Connection con = DriverManager.getConnection(databaseURL, user, password); session.putValue("myappl.connection", con);

. . .

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

The HTTPSession interface is implemented by the server. It enables a servlet to read and write the state information that is associated with an HTTPSession which is mostly useful in tracking and managing user sessions.

Method	Description
Void setMaxInactiveInterval(int interval)	It sets the maximum interval between request that this session will be kept by the host server.
Long getCreationTime()	Returns the time in milliseconds since midnight, Jan 1, 1970 for the session has been created.
String getId()	Returns the session ID.
Object getAttribute(String attr)	Returns the value associated with the name passed in attr. Returns null if attr is not found.
Object getValue(String name)	Return the object bound to the given name in the session's application layer data. Null if no such bindings.
void invalidate()	Invalidates this session and removes it from context.
Boolean isNew()	Returns true if the server has created the new session
Void putValue(String name,Object value)	It binds specified object into the application's layer data with the given name.

Servlet Collaboration:-

Sometimes servlets have to co-operate that is useful by sharing some information. This communication can be called as Sort Servlet Collaboration. The servlets which are collaborating can pass the shared information directly from one servlet to another. They can do it through the method invocation but for this approach, each servlet is required to know about the other servlet with which it is collaborating which is an unnecessary burden but java has provides us better technique for it.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Request Dispatcher Interface:-

It defines an object they receives request from the client and send them to any resources such as servlet, html file call JSP file and the server. The servlet container creates the Request Dispatcher. This is used to as Wrapper around server resource located a particular path for given by a particular name. They defines two methods.

- 1. forward
- 2. include

(1)void forward (ServletRequest req,ServletResponse res) throws ServletException,IOException

It will forward a request from the server to another resource. This method allows one servlet to do preliminary processing of a request and another resources to generate the response.

forward() should be called before the response has been committed to the client but if a response has been already commented then this method throws an IlligelStateException.

The request & response particular must be the object of the same class.

(2) void include (ServletRequest req, ServletResponse res) throws ServletException,IOException

It includes the contexts of resource in the response. The ServletResponse object has its path elements & parameter remains unchanged from the caller. The included servlet can not changed the response status code or if any attempt to make change will be ignore.

This method throws ServerException and IOException.

Web.xml

A web application's deployment descriptor describes the classes, resources and configuration of the application and how the web server uses them to serve web requests.

When the web server receives a request for the application, it uses the deployment descriptor to map the URL of the request to the code that use to handle the request.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

The deployment descriptor is a file named web.xml. It resides in the app's WAR under the WEB-INF/ directory. The file is an XML file whose root element is <web-app>.

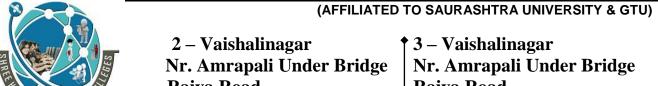
web.xml defines mappings between URL paths and the servlets that handle requests with those paths. The web server uses this configuration to identify the servlet to handle a given request and call the class method that corresponds to the request method (e.g. the doGet() method for HTTP GET requests).

To map a URL to a servlet, you declare the servlet with the <servlet> element, then define a mapping from a URL path to a servlet declaration with the <servlet-mapping> element.

HTTP Status Codes

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

- (1) informational responses,
- (2) successful responses,
- (3) redirects,
- (4) client errors,
- (5) servers errors.



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Information responses

100 Continue

This interim response indicates that everything so far is OK and that the client should continue with the request or ignore it if it is already finished.

101 Switching Protocol

This code is sent in response to an Upgrade request header by the client, and indicates the protocol the server is switching to.

102 Processing (WebDAV)

This code indicates that the server has received and is processing the request, but no response is available yet.

103 Early Hints

This status code is primarily intended to be used with the Link header to allow the user agent to start preloading resources while the server is still preparing a response.

Successful responses

200 OK

The request has succeeded. The meaning of a success varies depending on the HTTP method:

GET: The resource has been fetched and is transmitted in the message body. HEAD: The entity headers are in the message body. PUT or POST: The resource describing the result of the action is transmitted in the message body.

TRACE: The message body contains the request message as received by the server

201 Created

The request has succeeded and a new resource has been created as a result of it. This is typically the response sent after a POST request, or after some PUT requests.

202 Accepted

The request has been received but not yet acted upon. It is non-committal, meaning that there is no way in HTTP to later send an asynchronous response indicating the outcome of processing the request. It is intended for cases where another process or server handles the request, or for batch processing.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

203 Non-Authoritative Information

This response code means returned meta-information set is not exact set as available from the origin server, but collected from a local or a third party copy. Except this condition, 200 OK response should be preferred instead of this response.

204 No Content

There is no content to send for this request, but the headers may be useful. The useragent may update its cached headers for this resource with the new ones.

205 Reset Content

This response code is sent after accomplishing request to tell user agent reset document view which sent this request.

206 Partial Content

This response code is used because of range header sent by the client to separate download into multiple streams.

207 Multi-Status (WebDAV)

A Multi-Status response conveys information about multiple resources in situations where multiple status codes might be appropriate.

208 Multi-Status (WebDAV)

Used inside a DAV: propstat response element to avoid enumerating the internal members of multiple bindings to the same collection repeatedly.

226 IM Used (HTTP Delta encoding)

The server has fulfilled a GET request for the resource, and the response is a representation of the result of one or more instance-manipulations applied to the current instance.

Redirection messages

300 Multiple Choice

The request has more than one possible response. The user-agent or user should choose one of them. There is no standardized way of choosing one of the responses.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

301 Moved Permanently

This response code means that the URI of the requested resource has been changed permanently. Probably, the new URI would be given in the response.

302 Found

This response code means that the URI of requested resource has been changed *temporarily*. New changes in the URI might be made in the future. Therefore, this same URI should be used by the client in future requests.

303 See Other

The server sent this response to direct the client to get the requested resource at another URI with a GET request.

304 Not Modified

This is used for caching purposes. It tells the client that the response has not been modified, so the client can continue to use the same cached version of the response.

305 Use Proxy

Was defined in a previous version of the HTTP specification to indicate that a requested response must be accessed by a proxy. It has been deprecated due to security concerns regarding in-band configuration of a proxy.

306 unused

This response code is no longer used, it is just reserved currently. It was used in a previous version of the HTTP 1.1 specification.

307 Temporary Redirect

The server sends this response to direct the client to get the requested resource at another URI with same method that was used in the prior request. This has the same semantics as the 302 Found HTTP response code, with the exception that the user agent *must not* change the HTTP method used: If a POST was used in the first request, a POST must be used in the second request.

308 Permanent Redirect

This means that the resource is now permanently located at another URI, specified by the Location: HTTP Response header. This has the same semantics as the 301 Moved Permanently HTTP response code, with the exception that the user agent *must not* change the HTTP method used: If a POST was used in the first request, a POST must be used in the second request.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Client error responses

400 Bad Request

This response means that server could not understand the request due to invalid syntax.

401 Unauthorized

Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.

402 Payment Required

This response code is reserved for future use. Initial aim for creating this code was using it for digital payment systems, however this status code is used very rarely and no standard convention exists.

403 Forbidden

The client does not have access rights to the content, i.e. they are unauthorized, so server is rejecting to give proper response. Unlike 401, the client's identity is known to the server.

404 Not Found

The server can not find requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 to hide the existence of a resource from an unauthorized client. This response code is probably the most famous one due to its frequent occurrence on the web.

405 Method Not Allowed

The request method is known by the server but has been disabled and cannot be used. For example, an API may forbid DELETE-ing a resource. The two mandatory methods, GET and HEAD, must never be disabled and should not return this error code.

406 Not Acceptable

This response is sent when the web server, after performing <u>server-driven content</u> <u>negotiation</u>, doesn't find any content following the criteria given by the user agent.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

407 Proxy Authentication Required

This is similar to 401 but authentication is needed to be done by a proxy.

408 Request Timeout

This response is sent on an idle connection by some servers, even without any previous request by the client. It means that the server would like to shut down this unused connection. This response is used much more since some browsers, like Chrome, Firefox 27+, or IE9, use HTTP pre-connection mechanisms to speed up surfing. Also note that some servers merely shut down the connection without sending this message.

409 Conflict

This response is sent when a request conflicts with the current state of the server.

410 Gone

This response would be sent when the requested content has been permanently deleted from server, with no forwarding address. Clients are expected to remove their caches and links to the resource. The HTTP specification intends this status code to be used for "limited-time, promotional services". APIs should not feel compelled to indicate resources that have been deleted with this status code.

411 Length Required

Server rejected the request because the Content-Length header field is not defined and the server requires it.

412 Precondition Failed

The client has indicated preconditions in its headers which the server does not meet.

413 Payload Too Large

Request entity is larger than limits defined by server; the server might close the connection or return an Retry-After header field.

414 URI Too Long

The URI requested by the client is longer than the server is willing to interpret.

415 Unsupported Media Type

The media format of the requested data is not supported by the server, so the server is rejecting the request.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

416 Requested Range Not Satisfiable

The range specified by the Range header field in the request can't be fulfilled; it's possible that the range is outside the size of the target URI's data.

417 Expectation Failed

This response code means the expectation indicated by the Expect request header field can't be met by the server.

418 I'm a teapot

The server refuses the attempt to brew coffee with a teapot.

421 Misdirected Request

The request was directed at a server that is not able to produce a response. This can be sent by a server that is not configured to produce responses for the combination of scheme and authority that are included in the request URI.

422 Unprocessable Entity (<u>WebDAV</u>)

The request was well-formed but was unable to be followed due to semantic errors.

423 Locked (WebDAV)

The resource that is being accessed is locked.

424 Failed Dependency (WebDAV)

The request failed due to failure of a previous request.

425 Too Early

Indicates that the server is unwilling to risk processing a request that might be replayed.

426 Upgrade Required

The server refuses to perform the request using the current protocol but might be willing to do so after the client upgrades to a different protocol. The server sends an Upgrade header in a 426 response to indicate the required protocol(s).

428 Precondition Required

The origin server requires the request to be conditional. Intended to prevent the 'lost update' problem, where a client GETs a resource's state, modifies it, and PUTs it back to the server, when meanwhile a third party has modified the state on the server, leading to a conflict.

429 Too Many Requests

The user has sent too many requests in a given amount of time ("rate limiting").

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

431 Request Header Fields Too Large

The server is unwilling to process the request because its header fields are too large. The request MAY be resubmitted after reducing the size of the request header fields.

451 Unavailable For Legal Reasons

The user requests an illegal resource, such as a web page censored by a government.

Server error responses

500 Internal Server Error

The server has encountered a situation it doesn't know how to handle.

501 Not Implemented

The request method is not supported by the server and cannot be handled. The only methods that servers are required to support (and therefore that must not return this code) are GET and HEAD.

502 Bad Gateway

This error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.

503 Service Unavailable

The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded. Note that together with this response, a user-friendly page explaining the problem should be sent. This responses should be used for temporary conditions and the Retry-After: HTTP header should, if possible, contain the estimated time before the recovery of the service. The webmaster must also take care about the caching-related headers that are sent along with this response, as these temporary condition responses should usually not be cached.

504 Gateway Timeout

This error response is given when the server is acting as a gateway and cannot get a response in time.

505 HTTP Version Not Supported

The HTTP version used in the request is not supported by the server.

506 Variant Also Negotiates

The server has an internal configuration error: transparent content negotiation for the request results in a circular reference.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

507 Insufficient Storage

The server has an internal configuration error: the chosen variant resource is configured to engage in transparent content negotiation itself, and is therefore not a proper end point in the negotiation process.

511 Network Authentication Required

The 511 status code indicates that the client needs to authenticate to gain network access.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Chapter - 3 Listeners and Filters

(1) Event and Listener in Servlet

Events are basically occurrence of something. Changing the state of an object is known as an event.

We can perform some important tasks at the occurrence of these exceptions, such as counting total and current logged-in users, creating tables of the database at time of deploying the project, creating database connection object etc.

There are many Event classes and Listener interfaces in the javax.servlet and javax.servlet.http packages.

Event classes

The event classes are as follows:

- 1. ServletRequestEvent
- 2. ServletContextEvent
- 3. ServletRequestAttributeEvent
- 4. ServletContextAttributeEvent
- 5. HttpSessionEvent
- 6. HttpSessionBindingEvent

Event Interfaces

Event Interfaces include the following

- 1. ServletRequestListener
- 2. ServletRequestAttributeListener
- ServletContextListener
- 4. ServletContextAttributeListener
- 5. HttpSessionListener
- 6. HttpSessionAttributeListener

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

- 7. HttpSessionBindingListener
- 8. HttpSessionActivationListener

* ServletContextEvent and ServletContextListener

The ServletContextEvent is notified when web application is deployed on the server.

If you want to perform some action at the time of deploying the web application such as creating database connection, creating all the tables of the project etc, you need to implement ServletContextListener interface and provide the implementation of its methods.

Constructor of ServletContextEvent class

There is only one constructor defined in the ServletContextEvent class. The web container creates the instance of ServletContextEvent after the ServletContext instance.

ServletContextEvent(ServletContext e)

Method of ServletContextEvent class

There is only one method defined in the ServletContextEvent class:

1. **public ServletContext getServletContext()**: returns the instance of ServletContext.

Methods of ServletContextListener interface

There are two methods declared in the ServletContextListener interface which must be implemented by the servlet programmer to perform some action such as creating database connection etc.

- 1. **public void contextInitialized(ServletContextEvent e)**: is invoked when application is deployed on the server.
- 2. **public void contextDestroyed(ServletContextEvent e)**: is invoked when application is undeployed from the server.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

* HttpSessionEvent and HttpSessionListener

The HttpSessionEvent is notified when session object is changed. The corresponding Listener interface for this event is HttpSessionListener.

We can perform some operations at this event such as counting total and current logged-in users, maintaing a log of user details such as login time, logout time etc.

Methods of HttpSessionListener interface

There are two methods declared in the HttpSessionListener interface which must be implemented by the servlet programmer to perform some action

- public void sessionCreated(HttpSessionEvent e): is invoked when session object is created.
- public void sessionDestroyed(ServletContextEvent e): is invoked when session is invalidated.

* Interface ServletRequestListener

All Superinterfaces:

java.util.EventListener

public interface **ServletRequestListener** extends java.util.EventListener

Interface for receiving notification events about requests coming into and going out of scope of a web application.

A ServletRequest is defined as coming into scope of a web application when it is about to enter the first servlet or filter of the web application, and as going out of scope as it exits the last servlet or the first filter in the chain.

In order to receive these notification events, the implementation class must be either declared in the deployment descriptor of the web application, annotated with WebListener, or registered via one of the addListener methods defined on ServletContext.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Implementations of this interface are invoked at their

requestInitialized (javax.servlet.ServletRequestEvent) method in the order in which they have been declared, and at their

requestDestroyed (javax.servlet.ServletRequestEvent) method in reverse order.

Method Summary

void | requestDestroyed(ServletRequestEvent sre)

Receives notification that a ServletRequest is about to go out of scope of the web application.

void **requestInitialized**(ServletRequestEvent sre)

Receives notification that a ServletRequest is about to come into scope of the web application.

Method Detail

requestDestroyed

void requestDestroyed(ServletRequestEvent sre)

Receives notification that a ServletRequest is about to go out of scope of the web application.

Parameters:

sre - the ServletRequestEvent containing the ServletRequest and the ServletContext representing the web application

requestInitialized

void requestInitialized(ServletRequestEvent sre)

Receives notification that a ServletRequest is about to come into scope of the web application.

Parameters:

sre - the ServletRequestEvent containing the ServletRequest and the ServletContext representing the web application



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

* Interface ServletContextAttributeListener

All Superinterfaces:

java.util.EventListener

public interface **ServletContextAttributeListener** extends java.util.EventListener

Interface for receiving notification events about ServletContext attribute changes.

In order to receive these notification events, the implementation class must be either declared in the deployment descriptor of the web application, annotated with WebListener, or registered via one of the addListener methods defined on ServletContext.

The order in which implementations of this interface are invoked is unspecified.

Meth	Method Summary			
void	attributeAdded(ServletContextAttributeEvent event) Receives notification that an attribute has been added to the ServletContext.			
void	attributeRemoved(ServletContextAttributeEvent event) Receives notification that an attribute has been removed from the ServletContext.			
void	attributeReplaced(ServletContextAttributeEvent event)			

Method Detail

attributeAdded

void attributeAdded(ServletContextAttributeEvent event)

Receives notification that an attribute has been added to the ServletContext.

Parameters:

event - the ServletContextAttributeEvent containing the ServletContext to which the attribute was added, along with the attribute name and value

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

attributeRemoved

void attributeRemoved(ServletContextAttributeEvent event)

Receives notification that an attribute has been removed from the ServletContext.

Parameters:

event - the ServletContextAttributeEvent containing the ServletContext from which the attribute was removed, along with the attribute name and value

attributeReplaced

void attributeReplaced(ServletContextAttributeEvent event)

Interface ServletRequestAttributeListener

All Superinterfaces:

java.util.EventListener

public interface **ServletRequestAttributeListener** extends java.util.EventListener

Interface for receiving notification events about ServletRequest attribute changes.

Notifications will be generated while the request is within the scope of the web application. A ServletRequest is defined as coming into scope of a web application when it is about to enter the first servlet or filter of the web application, and as going out of scope when it exits the last servlet or the first filter in the chain.

In order to receive these notification events, the implementation class must be either declared in the deployment descriptor of the web application, annotated with WebListener, or registered via one of the addListener methods defined on ServletContext.

The order in which implementations of this interface are invoked is unspecified.

Metho	Method Summary			
void	attributeAdded(ServletRequestAttributeEvent srae) Receives notification that an attribute has been added to the ServletRequest.			
void	attributeRemoved(ServletRequestAttributeEvent srae)			
	Receives notification that an attribute has been removed from the			



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

	ServletRequest.
void	attributeReplaced(ServletRequestAttributeEvent srae)
	Receives notification that an attribute has been replaced on the ServletRequest.

Method Detail

attributeAdded

void attributeAdded(ServletRequestAttributeEvent srae)

Receives notification that an attribute has been added to the ServletRequest.

Parameters:

srae - the ServletRequestAttributeEvent containing the ServletRequest and the name and value of the attribute that was added

attributeRemoved

void attributeRemoved(ServletRequestAttributeEvent srae)

Receives notification that an attribute has been removed from the ServletRequest.

Parameters:

srae - the ServletRequestAttributeEvent containing the ServletRequest and the name and value of the attribute that was removed

attributeReplaced

void attributeReplaced(ServletRequestAttributeEvent srae)

Receives notification that an attribute has been replaced on the ServletRequest.

Parameters:

srae - the ServletRequestAttributeEvent containing the ServletRequest and the name and (old) value of the attribute that was replaced

Interface HttpSessionAttributeListener

All Superinterfaces:

java.util.EventListener

public interface **HttpSessionAttributeListener** extends java.util.EventListener

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Interface for receiving notification events about HttpSession attribute changes.

In order to receive these notification events, the implementation class must be either declared in the deployment descriptor of the web application, annotated with WebListener, or registered via one of the addListener methods defined on ServletContext.

The order in which implementations of this interface are invoked is unspecified.

Metho	Method Summary		
void	attributeAdded(HttpSessionBindingEvent event) Receives notification that an attribute has been added to a session.		
void	attributeRemoved(HttpSessionBindingEvent event) Receives notification that an attribute has been removed from a session.		
void	attributeReplaced(HttpSessionBindingEvent event) Receives notification that an attribute has been replaced in a session.		

Method Detail

attributeAdded

void attributeAdded(HttpSessionBindingEvent event)

Receives notification that an attribute has been added to a session.

Parameters:

event - the HttpSessionBindingEvent containing the session and the name and value of the attribute that was added

attributeRemoved

void attributeRemoved(HttpSessionBindingEvent event)

Receives notification that an attribute has been removed from a session.

Parameters:

event - the HttpSessionBindingEvent containing the session and the name and value of the attribute that was removed

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

attributeReplaced

void attributeReplaced(HttpSessionBindingEvent event)

Receives notification that an attribute has been replaced in a session.

Parameters:

event - the HttpSessionBindingEvent containing the session and the name and (old) value of the attribute that was replaced

Servlet Filters

Servlet filters are used for preprocessing Web application requests and postprocessing responses, as described in the following sections:

- Overview of Servlet Filters
- How the Servlet Container Invokes Filters
- Filtering of Forward or Include Targets
- Filter Examples

Overview of Servlet Filters

When the servlet container calls a method in a servlet on behalf of the client, the HTTP request that the client sent is, by default, passed directly to the servlet. The response that the servlet generates is, by default, passed directly back to the client, with its content unmodified by the container. In this scenario, the servlet must process the request and generate as much of the response as the application requires.

But there are many cases in which some preprocessing of the request for servlets would be useful. In addition, it is sometimes useful to modify the response from a class of servlets. One example is encryption. A servlet, or a group of servlets in an application, may generate response data that is sensitive and should not go out over the network in clear-text form, especially when the connection has been made using a nonsecure protocol such as HTTP. A filter can encrypt the responses. Of course, in this case the client must be able to decrypt the responses.

A common scenario for a filter is one in which you want to apply preprocessing or postprocessing to requests or responses for a group of servlets, not just a single servlet. If you need to modify the request or response for just one servlet, there is no need to create a filter—just do what is required directly in the servlet itself.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

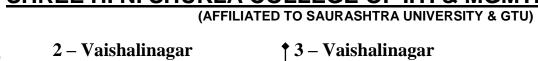
Note that filters are not servlets. They do not implement and override HttpServlet methods such as doGet() or doPost(). Rather, a filter implements the methods of the javax.servlet.Filter interface.

The methods are:

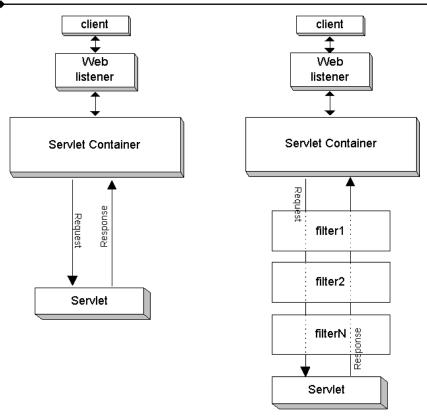
- init()
- destroy()
- doFilter()

How the Servlet Container Invokes Filters

Figure 3-1 shows how the servlet container invokes filters. On the left is a scenario in which no filters are configured for the servlet being called. On the right, several filters (1, 2, ..., N) have been configured in a chain to be invoked by the container before the servlet is called and after it has responded. The web.xml file specifies which servlets cause the container to invoke the filters.



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645



Description of "Figure 3-1 Servlet Invocation with and without Filters"

The order in which filters are invoked depends on the order in which they are configured in the web.xml file. The first filter in web.xml is the first one invoked during the request, and the last filter in web.xml is the first one invoked during the response. Note the reverse order during the response.

* Servlet Filter

A **filter** is an object that is invoked at the preprocessing and postprocessing of a request.

It is mainly used to perform filtering tasks such as conversion, logging, compression, encryption and decryption, input validation etc.

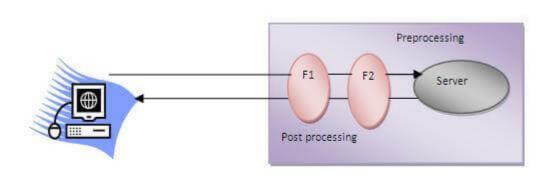
The **servlet filter is pluggable**, i.e. its entry is defined in the web.xml file, if we remove the entry of filter from the web.xml file, filter will be removed automatically and we don't need to change the servlet.

So maintenance cost will be less.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645



Usage of Filter

- recording all incoming requests
- o logs the IP addresses of the computers from which the requests originate
- conversion
- data compression
- encryption and decryption
- o input validation etc.

Advantage of Filter

- 1. Filter is pluggable.
- 2. One filter don't have dependency onto another resource.
- 3. Less Maintenance

Filter API

Like servlet filter have its own API. The javax.servlet package contains the three interfaces of Filter API.

- 1. Filter
- 2. FilterChain
- 3. FilterConfig



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

1) Filter interface

For creating any filter, you must implement the Filter interface. Filter interface provides the life cycle methods for a filter.

Method	Description
public void init(FilterConfig config)	init() method is invoked only once. It is used to initialize the filter.
public void doFilter(HttpServletRequest request,HttpServletResponse response, FilterChain chain)	doFilter() method is invoked every time when user request to any resource, to which the filter is mapped. It is used to perform filtering tasks.
public void destroy()	This is invoked only once when filter is taken out of the service.

2) FilterChain interface

The object of FilterChain is responsible to invoke the next filter or resource in the chain. This object is passed in the doFilter method of Filter interface. The FilterChain interface contains only one method:

1. public void doFilter(HttpServletRequest request, HttpServletResponse response): it passes the control to the next filter or resource.

How to define Filter

We can define filter same as servlet. Let's see the elements of filter and filter-mapping.

- <web-app>
- <filter>
- <filter-name>...</filter-name>
- <filter-class>...</filter-class>



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

<filter-mapping> <filter-name><url-pattern></url-pattern> </filter-name></filter-mapping>	
For mapping filter we can u	•

For mapping filter we can use, either url-pattern or servlet-name. The url-pattern elements has an advantage over servlet-name element i.e. it can be applied on servlet, JSP or HTML.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

CHAPTER 4 JSP PROGRAMMING

Introduction:-

JSP is Java Server Pages (JSP) technology enables you to mix regular, static HTML with dynamically generated content. JSP are run in a server side component known as JSP container which translate them into equivalent Java Servlet. You simply write the regular HTML in the normal manner, using familiar Web-page-building tools. You then enclose the code for the dynamic parts in special tags, most of which start with <% and end with %>.

JSP include:

- Static HTML/XML components
- Special JSP tags.
- Code written in the java language called"scriptless".

Benefits of JSP

No body can borrow the code

The JSP code written and runs and remains on the server. So issue of copy source code does not arise at all. All of JSP's functionality is handled before the page is sent to browser.

•Faster loading of pages.

With JSP, decision can be made about what user want to see at web server prior to pages being dispatched. So only the content that the user is interested will be dispatched to the user. There is no extra code and extra content.

• No browser compatibility Issues.

JSP pages can run same way in browser. The developer ends up sending standard HTML to a user browser. This largely eliminates scripting issues and cross browser compatibility.

JSP support

JSP is supported by number of web server like Apache, Microsoft IIS and PWS ,Netscapes's FastTracks and Enterprise web server and others. Built in support for JSP is available Java Server from Sun Microsystem.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

• Compilation

JSP compiled before the web server processes it. This allows the server to handle JSP pages much faster, because inn the older technologies such as CGI require the server to load an interpreter and the target script each time the page is requested.

• JSP elements in HTML/XML pages

JSP page look like HTML /XML page ,it holds text marked with a collection of tags. while a regular JSP page is not a valid XML page, there is a variant JSP tag syntax that lets the developer use JSP tags within XML documents.

Disadvantages of JSP

1. Attractive Java Code

Putting java code within web page is really bad design, but JSP makes it tempting to do just that. Avoid this as far as possible. It is done using template using.

2. Java Code required

To relatively simple things in JSP ccan actually demand putting java code in a page.

3. Simple task are hard to code

Even including page headers and footers is a bit difficult with JSP.

4. Difficult looping in JSP

In regular JSP pages looping is difficult .In advance JSP we can use some custom tags for looping.

5. Occupies a lot of space.

JSP consumes extra hard drive and memory space.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Servlet v/s JSP

Servlet is a java code.	JSP is a html based 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 MVC approach.	JSP is the view in MVC approach for showing output.
Serviet is faster than JSP.	JSP is slower than Servlet because the first step in JSP lifecycle is the translation of JSP to java code and then compile.
Servlet can accept all protocol requests.	JSP only accept 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.
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 javaBeans.

Step for JSP request shown in figure:

- (1) When user goes to jsp page web browser make a request via Internet.
- (2) JSP request then sent to the server.
- (3) Web server recognize the .jsp file and passes the JSP file to the JSP servlet Engine.
- (4) If JSP file has been called for first time, Then JSP file is parsed, otherwise servlet is instantiated.
- (5) The next step is to generate a servlet from jsp file .the entire HTML required converted to println statements.
- (6) Servlet source code is compiled into class file.
- (7) Servlet is instantiated ,calling the init and service method.
- (8) HTML from the servlet output is sent via the Internet.

HTML results are displayed on the user's web browser.

is



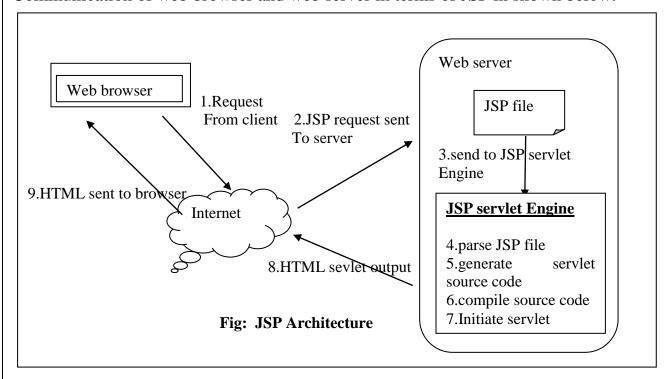
(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

JSP Architecture Development

The purpose of JSP is to provide a declarative, presentation centric method of developing servlet.

Communication of web browser and web server in terms of JSP in shown below:



JSP LIFECYCLE

A Java Server Page life cycle is defined as the process started with its creation which later translated to a servlet and afterward servlet lifecycle comes into play. This is how the process goes on until its destruction.

Following steps are involved in JSP life cycle:

- Translation of JSP page to Servlet
- Compilation of JSP page(Compilation of JSP into test.java)
- Classloading (test.java to test.class)
- Instantiation(Object of the generated Servlet is created)
- Initialization(jspInit() method is invoked by the container)

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar

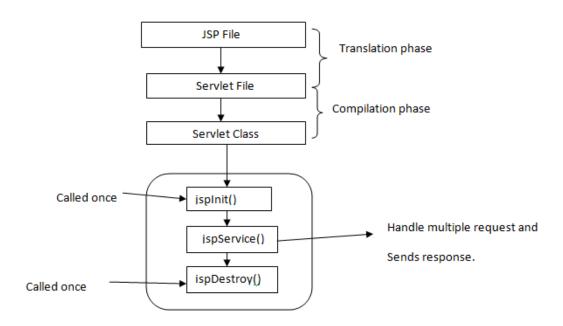
Nr. Amranali Under Bridge

Nr. Amranali Under Bridge

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

- Request processing(_jspService()is invoked by the container)
- JSP Cleanup (jspDestroy() method is invoked by the container)

JSP Life Cycle



Translation of JSP page to Servlet:

This is the first step of JSP life cycle. This translation phase deals with Syntactic correctness of JSP. Here test.jsp file is translated to test.java.

Compilation of JSP page:

Here the generated java servlet file (test.java) is compiled to a class file (test.class).

Classloading:

Servlet class which has been loaded from JSP source is now loaded into container.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Instantiation:

Here instance of the class is generated. The container manages one or more instance by providing response to requests.

Initialization:

jspInit() method is called only once during the life cycle immediately after the generation of Servlet instance from JSP.

Request processing:

_jspService() method is used to serve the raised requests by JSP.It takes request and response object as parameters. This method cannot be overridden.

JSP Cleanup:

In order to remove the JSP from use by the container or to destroy method for servlets jspDestroy()method is used. This method is called once.

• jspInit()

when jsp servlet instane is created ,jspInit() called . It is similar to init() of servlet.

• _jspService()

This is similar to service() of servlet .when jsp servlet instance is called ._jspService() is called where request and response object are sent.

Syntax:

Public void _jspService(HttpServletRequest req,HttpServletResponse res)throws Servlet Exception,IOException



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

• jspDestroy()

This is similar to destroy() of servlet .This method is called when the JSp servlet instance is destroyed from the web container.

Syntax:

Pubic void destroy()

Request-response cycle of JSP

- 1) Clinet sends an HTTP request for a JSP page to the server, either using a GET /POST method.
- 2) the browser identifies the JSP files and compiles and loads the corresponding servlet class if the servlet class is not already present in the JVM. The server calls the service() which often passes the request and response to the doGet() or doPost() .the servlet processes the input form the request and prepares an HTML page as a response.
- 3) The response is sent back to the client browser.

JSP Elements

JSP elements are instruction to JSP container about what code to generate and how it should operate.

JSP elements have a special identity to jsp compiler because it starts and ends with special kind of tags.

template data(HTML) code is not compiled by the jsp compiler and also not recognized by the jsp container.

it is also known as Component of the Jsp pages. There are basically three type of JSP elements are as given follow.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

- 1. Directive Elements
- 2. Scripting Elements
- 3. Action Elements

Directive Elelments

The role of directive is to pass information to the JSP container. Syntax:

<%@ directive {attribute name="value"} %>

There are three type of directive elements are as given below.

- Page Directive
- Include Directive
- Taglib Directive
- * Page Directive:

Page Directive is uses to specify attributes for the whole JSP page.syntax is as follows:

<%@ page [attribute1="value1" attribute2="value2"......attributeN= "valueN"] %>

Where the attribute can be as follows:

Attribute	Use(description)	Default value
Name		
Language	It is used to define the language	Java
	which is used with the scriptlet	
	elements.Most probably its valid	
	value is java only.	
extends	It defines fully qualified name of the	
	super class of the jsp page.	
Import	It defines list of packages which are	Javax.servlet.*,
	imported with the JSP page with its	Javax.servlet.http.*,
	fully qualified name.	Javax.servlet.jsp.*,java.lang.*
Session	It defines the Boolean value	True
	indicating if jsp page require HTTP	
	session then its value Is true else it	
	becomes false.	

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

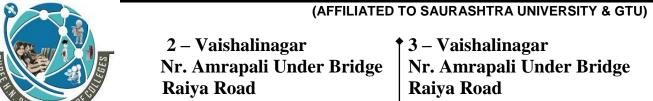


2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 † 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Buffer	It specifies the size of buffer.	8 kb
autoFlush	It defines the Boolean indicating if it	True
	is true then it automatically flushed	
	the buffer ad if false then it throws	
	an exception buffer overflow.	
isThreadSafe	It defines the Boolean indicating if it	True
	is true jsp page handles the all the	
	request simultaneously from	
	multiple threads if false then	
	generates servlet declares that it	
	implements the SingleThreadModel.	
Info	It returns string message	-
	information related to jsp page using	
	getServletInfo()	
isErrorPage	It defines the Boolean indicating if it	False
	is true jsp page considered as error	
	page and if false then it is normal	
	page and implements it with other	
	jsp page to handle exception.	
errorPage	It defines the url of the error page	
	we want to implement the error page	
	with the other jsp pages we can use	
	this directive.	
contentType	It specifies the MIME type and	<% @ page
	character encoding which used with	contentType="text/html"%>
	generated servlet.	
pageEncoding	It defines the character encoding of	ISO 8859 -1 (Latin script)for
	the jsp page	JSP style and UTF-8 for xml
		style tags.

Syntax for all jsp directives:

<%@page [language="java"]
[extends="package.class"]
[import="{package.class|package.*},...."
[session="true/false"]
[buffer="none|8kb|sizekb"]
[autoFlush="true|false"]</pre>



Nr. Amrapali Under Bridge **Rajkot - 360001** Ph. No-(0281) 2440478

† 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

```
[isThreadSafe="true/false"]
[info="text"]
[errorpage="relativeURL"]
[contentType="MIME[;charset=charset]" | "text/html";
Charset=ISO 8859-1"]
[isErrorPage="true|false"]%>
```

Example:

```
<%@page import="java.util.*;" %>
<html>
  <head>
         <title> Page Directive Example </title>
  </head>
  <body>
    <%
    Date d1=new Date();
    out.println("curent date is=");
    out.println(d1);
    %>
  </body>
</html>
```

* Include Directive

Include directive is used to include the static page and dynamic pages with the other jsp pages. For example if we want to Set same header and footer for all the pages we can create header.html and can include it with necessary jsp pages. In short it is used to insert a part of the code that is common to multiple pages.

```
Syntax:<%@include file="relative path"%>
The file attribute is used to specify the name of the file to be included.
```

First.html

```
<html>
 <head><title>Include example</title></head>
 <body>
```





2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

```
Username : <input type="text" name="txtnm" value="">
  Password : <input type="password" name="txtps" value="">
    <input type="submit" name="submit" value="submit">
 </body>
</html>
IncludeDemo.jsp
<%@include file="Incrementdemo.jsp" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;</pre>
     charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>welcome to include directive example</h1>
  </body>
</html>
Example for include one jsp page to another jsp page:
IncrementDemo.jsp
<html>
  <head>
         <title>JSP Page</title>
  </head>
  <body>
    <% for (int sctr=0; ctr<=5; ctr ++)
       { %>
       The value of ctr is <%= ctr% ><br/>
    <% } %>
  </body>
```

(1) Taglib Directive

</html>

It declares that the JSP file uses custom tags, names the tag library that defines them, and specifies their tag prefix. Syntax:



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

<% @ Taglib uri="URI toTagLibrary" prefix="tagPrefix" %>
Example: <%@ taglib uri="/tlds/Fancy/TableGenerator.tld" prefix="fancytable" %> < fancytable: table1>
<pre></pre>
< fancytable: table2>
<pre></pre>

2. Scripting Elements

Scripting Elements are used to write java code with the jsp file. As you know in jsp java code is embedded within HTML code. You need to write some java language statements or use java features within the JSP page.

There are 3 types of scripting elements which are as follows:

- Scriptlets
- Declaration
- Expression

1. Scriptlets:

Scriptlets are block of java code with the jsp page. Scriptlets starts with <% tag and end with %>closing tag. JSP converted into servlet code and then JSP engine adds all processing statements of JSP processes them under _jspService().

Syntax:
<%
Statement 1;
Statement 2;
Statement n;



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

%>

Example:

We can write any valid java statements with the scriptlets declaring variable, writing processing and output part.

2. Declaration:

Declaration tags are used to declare the variables, methods and instance of the classes with the jsp pagescriptlets code become part of the _jspService() whereas declaration code is incorporated into generate d source file outside the _jspService().Declaration starts with <%! tag and end with %>closing tag.

```
<%!
Statement 1;
Statement 2;
Statement n;
%>
Example:
```

3. Expression:

Expression element is used to print value of any variable or any valid expression when the jsp page is requested. All the expressions are printed automatically by converting values into string values. If the result can not be converted into a string ,an error will be raised at translation time .An expression starts with <% = and ends with %>

Syntax:

<%= expressions %>

Example:

Current Time :< %= java.util.Calendar.getInstance().getTime() %>

This expression will be translated into the following statements in the _jspService() of the generated servlet

3.Action Elements:



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Action elements are high level jsp elements which are used to create, modify and use other objects. Syntax of action element's tags just like XML syntax.

- <jsp:param>
- <jsp:include>
- <jsp:forward>
- <jsp:plugin>

I. <jsp:param>

This element Is used to provide the tag/value pair of information ,by including these as sub attribute of the <jsp:include>,<jsp:forward> and the <jsp:plugin> actions.

Syntax:

```
<jsp:param name="pname" value="pvalue"/>
OR
<jsp:param name="pname" value="pvalue">
</jsp: param>
```

II <jsp: include>

This element Is used to include static and dynamic resource of the current JSP page. This object is just used to include resource on the current JSP page. It can not be used to send response.

Syntax:

```
<jsp: include page="jsp page" flush="true/false"/>
<jsp: param name="pname" value="pvalue">
</jsp: include>
Example:

<jsp:include page="login.jsp" flush="true"/>
<jsp: param name="username" value="java">
</jsp:include>
```

Here <jsp: include> has two attribute which are:

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

page: Specifies the resource (JSP/HTML) file which will be included. flush: An optional parameter used to flush buffer. specified true/false.

Note: Once the buffer is flushed, the value can not be recalled as it is flushed from the memory directly.

Iii . <jsp:forward>:

This control is used to transfer control from current JSP page to another source which may be any valid source of application. It is obvious to understand that whenever this action elements is called, execution of current JSP page is stopped and control is transfered to another specified URL into <jsp:forward>:

Note:<jsp:forward> action element is same as forward() of request dispatcher of servlet programming.

Syntax:

<jsp:forward page="destinationpage"/>

Example

<jsp:forward page="abc.jsp"/>

Above example will transfer control from current page to abc.jsp

IV. <jsp:plugin>

This element Is used to embed an applet and java beans with the jsp page.the tag automatically detects the bowser type and inserts the appropriate HTML tag either<embed> or <object> in the output.

Syntax:

```
<jsp:plugin type="plugintype" code="classfilename" codebase="url">
</jsp:plugin>
```

Example:

```
<jsp:plugin type="applet" code="AppletDemo.class">
</jsp:plugin>
```



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

> COMMENTS AND TEMPALTE DATA

We can use two type o fcomment with JSP page one is for HTML and another for JSP.HTML comment:<! --this comment will appear in the client's browser --> JSP comment: <% -- this comment will not appear in the client's browser --%> JSP comment will not appear in the page output to the client.

Template Data:

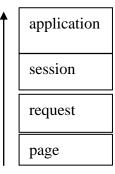
In JSP page everything that is not a directive, declaration, Scriptlets, expression, action elements or JSP comments is termed as Template Data. Usually all the HTML and text in the page.

SCOPE OF THE JSP OBJCTS

Objects that are created as part of the JSP page have a certain lifetime and may or may not be accessible to other components or objects in the web application. The lifetime and accessibility of an object is known as scope.

There are four valid scope.

- ➤ Page Scope
- Request Scope
- Session Scope
- > Application Scope



1. Page scope:

This scope is most restrictive. With page scope, the object is accessible only within the current jsp page in which it is defined.

Jsp implicit objects out, exception, response, config, page Context and page have 'page' context.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

2. Request scope:

Jsp object created using the 'request' scope can be accessed from any pages that serve that request.

This means that the object is available within the page in which it is created, and within pages to which the request is forwarded or included.

Implicit object request has 'request scope'

3. session scope:

Objects with session scope are available to all application component that participate in the client's session.

JSP object that is created using the session scope is bound to the session object. Implicit object session has the 'session' scope.

4.Application Scope:

Objects with application scope are available to the entire application for the life of the application.

JSP object is bound to the application object.Implicit object application has the 'application scope'.

IMPLICIT OBJECT OF JSP

Implicit objects means the object which is already created by JSP itself. Implicit objects are automatically created in JSP pages and can be used without declaring their object.

The reason behind that implicit object is, jsp page is also converted into servlet at last, so some of objects should be provided by JSP also for programming similar to sevlet. implicit object are used within scriptlet and expression elements.

- request
- response
- out
- session
- config
- exception
- application



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

1. request:

This is the most important object which is used in JSP programming. The entire request coming from the client can be obtained from this object.

Similar to servet programming ,request object in JSP also belongs to javax.servlet.httpServletRequest classs .

You can see all the parameters sent by client through the request object.

The request object has request scope. That means that the implicit request object can is in scope until the response to the client is complete.

Example:

```
<%
String u=request.getParameter("txt_username");
String p=request.getParameter("txt_password");
Out.println("welcome,"+u);
%>
```

The above example is used to retrieve username and password from request object and then print it.

2. response:

This is another most important object which is used in JSP programming. If you want to generate any output for client, the you can generate it using response object. As similar to sevlet programming, response object in jsp is belongs to javax.servlet.

3. session:

This object is used for session tracking .this belongs to javax.servlet.http.HttpSession class.session object Is for requesting the client is created under sessionimplicit object. We can use session method like getAttribute(), setAttribute(), getValue(), putValue(), removeAttribute(), removeValue(), isNew(), getId(), getCreationTime().

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

4. application

This object is used for set values and attributes at application level. Application object of JSP is similar to ServletContext object of servlet programming getServletConfig(), getServletContext() can be used to retrieve application object.

5. out:

This object is used to write output to the output stream of the client. The scope of out is current page out obect is created using javax.sevlet.jsp.JspWriter class.methods of out object is print(),println(),clear(),clearBuffer(),close(),flush().

6. config:

This object is similar to ServeltConfig object of servlet class.config object is created using javax.Servlet.

7. page:

As you know JSP page is converted into servlet class at last. Then the servlet instance is created of that perticuar sevlet class. Page is the instance of the jsp servlet class created by web container for the current request.

Implicit objects	Super class	Description	
Request	HttpServletRequest	Provides HTTP request information	
Response	HttpServletResponse	Send back data to the client	
Out	JspWriter	Write data to the response stream	
Session	HttpSession	Track information about a user from	
		one request to another.	
Application	ServletContext	Data shared by all JSP and servlets in	
		the application.	
pageContext	Pagecontext	Contains data associated with the	
		whole page.	
Config	ServletConfig	Provides servlet configuration data.	
Page	Objects	Similar with "this" object in java	
Exception	Throwable	Exception not caught by appliacation	
		code	

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Handling and Errors and Exception with JSP Page

Exception referred to the error occurred at run time. In java exception handled through the exception object. In jsp exception is implicit object with page scope.

It is an instance of java.lang.throwable as you know that Throwable class is the super class of all the exception and error classes in the java. Here with the jsp we can handle exception with Scriptlets, by creating an errorpage and with deployment descriptor.

Exception handling using Scriptlets:

Exception can be handled in a scriptlet in the same way as in java, by using the try and catch block.

Exception handling using the page Directive:

This is the second way to handle exception by creating an error page. To create an error page with JSP we can use is Errorpage=true attribute of page directive. After creating an error page we can handle the exception with any JSP page by invoking this error page with it using errorPage="URL of the error page" attribute of the page directive.

Exception handling using Deployment Descriptor

This is the third way to by which we can handle an exception with JSP page.

Deployment descriptor is a web.xml file which defines the classes, resources and configuration of the application and how the web server uses them to serve web requests. It resides in the application under the WEB-INF/directory.

INCLUDING AND FORWARDING FROM JSP

JSP pages have the ability to include other jsp pages or servlet in the output that is sent to the client, or to forward the request to other jsp page or servlet for servicing. This thing is possible through the standard actions, <jsp:include> and <jsp:forward>.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Include Action

Including a jsp page or servlet through a standard differs from the include directive.

<%@ include file="/WEB-INF/footer.jsp">

<jsp:directive.include file="/WEB-INF/footer.jsp">

When the jsp container translates the page. Using the include directive ,the included file does not need to be a complete and valid jsp page .

Syntax:

<jsp:include page="URL" flush="true/false">

<jsp:param name="paramname" value="paramvalue"/>

</jsp:include>

For the include element, the page attribute is required, and its value is the URL of the page whose output is included in the response. This URL is relative to the jsp page.

If the jsp needs to pass parameters to the included file, it does so with the <jsp:param>element.

Forward action

With the forward action, the current page stops processing the request and forward the request to another web component.

This other components complete the response. Execution never returns

Syntax:

<jsp:forward page="URL">

<jsp:param name="paramname" value="paramvalue"/>

</jsp:forward>

The meaning and use of the attributes and of the <jsp:param>element are the same as those for the include action.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Chapter - 5 MVC, Tag Library

- Integrating Servlets and JSP in a Web Application (MVC Architecture for Web Applications)
- Implementing MVC with Request Dispatcher, understanding Data Sharing Between Servlets and JSP
- JSP Expression Language
- Accessing Scoped Variables, Bean Properties, Collections and Implicit Objects Using EL, Using EL operators
- Tag Library- Basics; Using JSTL c:out, c:forEach, c:forTokens, c:if, c:choose, c:set, c:remove, c:import, c:url, c:param, c:redirect and c:catch Tags

Introduction to MVC:

 $\underline{\mathbf{M}}$ odel $\underline{\mathbf{V}}$ iew $\underline{\mathbf{C}}$ ontroller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

Model - The lowest level of the pattern which is responsible for maintaining data.

View - This is responsible for displaying all or a portion of the data to the user.

Controller - Software Code that controls the interactions between the Model and View.

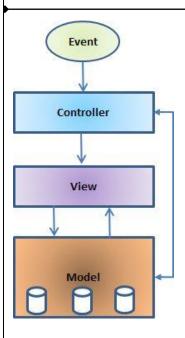
MVC is popular as it separate the application logic from the user interface layers.

Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645



Web Architecture Models

Different programmers may design an application in different ways. Designing an application depends on how the programmers recognize the problem and approaches for solving that problem. A development model facilitates the design process by separating the code according to the functions performed by different parts of the application. Two types of development models are used for web applications.

- Model-1 architecture
- Model-2 architecture

Model-1 Architecture

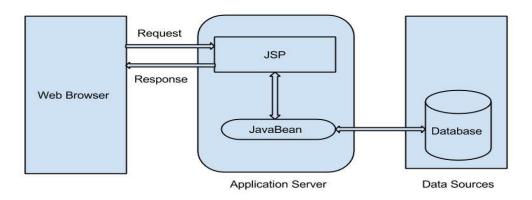
Servlet and JSP are the main technologies to develop the web applications. This model uses JSP to design applications which is responsible for all the activities provided by the application.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

The following figure shows Model-1 architecture.



In Model-1 architecture web applications are developed by combining both business and presentation logic. In this model, JSP pages receive requests which are transferred to data sources by JavaBeans. After the requests are serviced, JSP pages send responses back to client.

Advantages

By using this model it's easy to develop web application.

Disadvantages

- This architecture is not suitable for large applications.
- It increases the complexity of the program so it is difficult to debug the program.
- It needs more time to develop custom tags in JSP.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



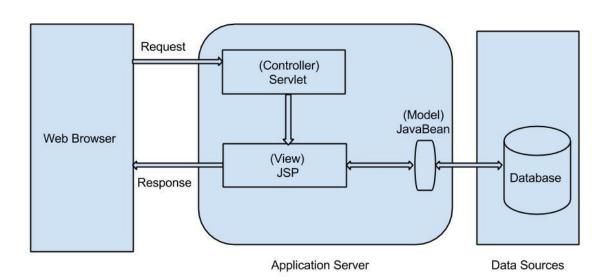
2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

• A single change in one page needs changes in other pages. So it is difficult to maintain.

Model-2 (MVC) Architecture

The Model-2 architecture is based on MVC design model i.e. Model, View and Controller.

Following figure shows Model-2 Architecture.



As shown in above model Servlet acts as Controller, JSP acts as View and JavaBean acts as Model. In this model, Controller receives requests which are transferred to data sources by View component and Model component. After the requests are serviced, JSP pages i.e. View component send responses back to client.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

These components can be described as follows:

- <u>Model:</u> It represents data and business logic of the application that specifies how data is accessed and updated. Model is implemented by using JavaBeans. It has no concerns with the presentation logic.
- <u>View:</u> It represents presentation logic. A View provides the Graphical User Interface (GUI) for model. It specifies how data should be presented to user in an understandable form. A user can interact with application through View and allows user to alter the data.
- <u>Controller</u>: It acts as interface between view and model. It receives HTTP request. It controls all the Views associated with a Model. It receives requests from the client and delegates the responsibility for producing the next phase of user interface to view component.

MVC Advantages

- This model is suitable for large applications.
- It allows use of reusable software components to design the business logic.
- It easy to maintain and test.

MVC Disadvantage

If we change the code in controller, we need to recompile the class and redeploy the application.

MVC architecture:

The **Model-View-Controller** (**MVC**) architecture is a widely used architectural approach for interactive applications that distributes functionality among application objects so as to minimize the degree of coupling between the objects.

To achieve this, it divides applications into three layers: model, view, and controller.

Each layer handles specific tasks and has responsibilities to the other layers:

The model

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

The model represents business data, along with business logic or operations that govern access and modification of this business data.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

The model notifies views when it changes and lets the view query the model about its state. It also lets the controller access application functionality encapsulated by the model.

The view

A presentation of data in a particular format, triggered by a controller's decision to present the data.

They are script based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology. The view renders the contents of a model. It gets data from the model and specifies how that data should be presented.

It updates data presentation when the model changes. A view also forwards user input to a controller.

The controller

The controller is responsible for responding to user input and perform interactions on the data model objects.

The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

The controller defines application behavior. It dispatches user requests and selects views for presentation. It interprets user inputs and maps them into actions to be performed by the model.

In a web application, user inputs are HTTP GET and POST requests. A controller selects the next view to display based on the user interactions and the outcome of the model operations.

RequestDispatcher in Servlet

The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the way of servlet collaboration.

There are two methods defined in the RequestDispatcher interface.

Methods of RequestDispatcher interface

The RequestDispatcher interface provides two methods. They are:

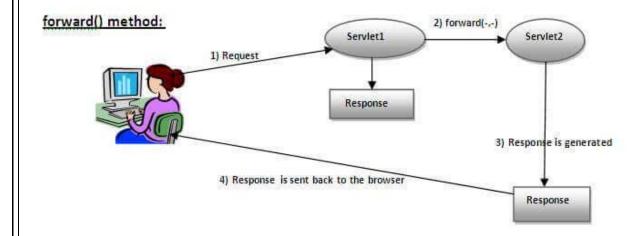
1. public void forward(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

2. public void include(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:Includes the content of a resource (servlet, JSP page, or HTML file) in the response.



As you see in the above figure, response of second servlet is sent to the client. Response of the first servlet is not displayed to the user.



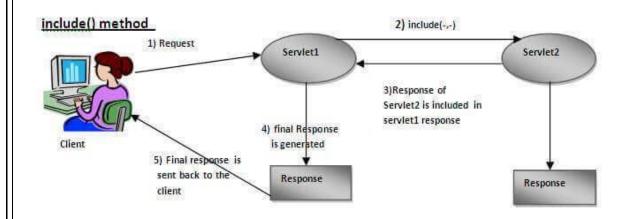
2 – Vaishalinagar † 3 – Vaishalina

Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001

Ph. No-(0281) 2440478

3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



As you can see in the above figure, response of second servlet is included in the response of the first servlet that is being sent to the client.

How to get the object of RequestDispatcher

The getRequestDispatcher() method of ServletRequest interface returns the object of RequestDispatcher. Syntax:

Syntax of getRequestDispatcher method public RequestDispatcher getRequestDispatcher(String resource);

Example of using getRequestDispatcher method

- 1. RequestDispatcher rd=request.getRequestDispatcher("servlet2");
- 2. //servlet2 is the url-pattern of the second servlet
- 4. rd.forward(request, response);//method may be include or forward

Example of RequestDispatcher interface

3.

In this example, we are validating the password entered by the user. If password is servlet, it will forward the request to the WelcomeServlet, otherwise will show an error message: sorry

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar

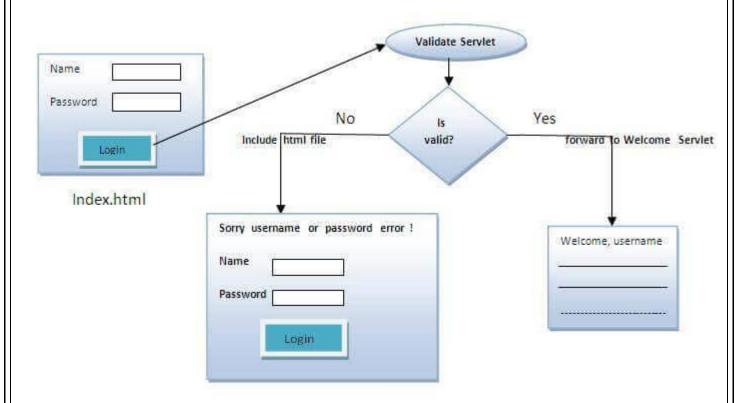
Nr. Amrapali Under Bridge

Nr. Amrapali Under Bridge

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

username or password error!. In this program, we are cheking for hardcoded information. But you can check it to the database also that we will see in the development chapter. In this example, we have created following files:

- o **index.html file:** for getting input from the user.
- Login.java file: a servlet class for processing the response. If password is servet, it will forward the request to the welcome servlet.
- WelcomeServlet.java file: a servlet class for displaying the welcome message.
- **web.xml file:** a deployment descriptor file that contains the information about the servlet.



Sharing Data Between Servlets and JSP Pages

When you use servlets for request processing and JSP pages to render the user interface, you often need a way to let the different components access the same data.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT. (AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

The application scope is just a JSP abstraction of javax.servlet.ServletContext attributes. Similarly, the request and session scopes are JSP abstractions for attributes associated withjavax.servlet.ServletRequest and javax.servlet.http.HttpSession, respectively.

All three classes provide setAttribute(), getAttribute(), and removeAttribute() methods. A servlet uses the setAttribute()method to make a bean available to a JSP page. For instance, a servlet can create a bean, save it as a request attribute, and then forward control to a JSP page like this:

To the JSP page, the bean appears as a request scope variable. It can therefore obtain the bean using the <jsp:useBean> action and then access the properties of the bean.

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

JSP Expression Language (EL)

- 1. Expression Language (EL) in JSP
- 2. Implicit Objects in Expression Language
- 3. Simple example of Expression Language that prints the name of the user
- 4. Example of Expression Language that prints the value set in the session scope
- 5. Precedence of Operators in EL
- 6. Reserve words in EL

The **Expression Language** (EL) simplifies the accessibility of data stored in the Java Bean component, and other objects like request, session, application etc.

There are many implicit objects, operators and reserve words in EL.

It is the newly added feature in JSP technology version 2.0.

Syntax for Expression Language (EL)

\${ expression }

Implicit Objects in Expression Language (EL)

There are many implicit objects in the Expression Language. They are as follows:

Implicit Objects	Usage
pageScope	it maps the given attribute name with the value set in the page scope
requestScope	it maps the given attribute name with the value set in the request scope
sessionScope	it maps the given attribute name with the value set in the session scope
applicationScope	it maps the given attribute name with the value set in the

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

	application scope
param	it maps the request parameter to the single value
paramValues	it maps the request parameter to an array of values
header	it maps the request header name to the single value
headerValues	it maps the request header name to an array of values
cookie	it maps the given cookie name to the cookie value
initParam	it maps the initialization parameter
pageContext	it provides access to many objects request, session etc.

EL param example

In this example, we have created two files index.jsp and process.jsp. The index.jsp file gets input from the user and sends the request to the process.jsp which in turn prints the name of the user using EL.

index.jsp

<form action="process.jsp">

Enter Name:<input type="text" name="name" />

<input type="submit" value="go"/>

</form>

process.jsp

Welcome, \${ param.name }

EL sessionScope example

In this example, we printing the data stored in the session scope using EL. For this purpose, we have used sessionScope object.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

index.jsp

<h3>welcome to index page</h3>

<%

session.setAttribute("user","ABC");

%>

visit

process.jsp

Value is \${ sessionScope.user }

EL cookie example

index.jsp

<h1>First JSP</h1>

<%

Cookie ck=**new** Cookie("name", "ABC");

response.addCookie(ck);

%>

click

process.jsp

Hello, \${cookie.name.value}

Operators in EL

There are many operators that have been provided in the Expression Language. Their precedence are as follows:

[].

()

-(unary) not! empty

* / div % mod

+ - (binary)

SHEET THE STATE OF THE STATE OF

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

<<=>>= lt le gt ge		
== != eq ne		
&& and		
or		
?:		

Accessing Scoped Variables

servlets can use four types of scope for the variables

- 1. Local scope
- 2. Page scope
- 3. Session scope
- 4. Application or server scope.

Local scope:

If a variable gets declared inside any method of the servlet or any block then it becomes local scope.

Page scope:

If a variable can be available to all parts of the servlets and all users share one instance of the variable, then it becomes page scope. An instance or class variable of the servlet can be called as page scope variable. Such types of variables can be used to share common information about all users in all servlets.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Session scope: -

If a variable can be available to all servlets and for each user it becomes individual then it is called as user or session scope. Such types of variable can be used to store individual information for each user and for multiple servlet.

It can be created by using cookie or session.

Application or server scope:-

If a variable can be available to all servlets and all users share a common instance of the variable then it is called as application or server scope. Such type of variables can be used to store common information about all servlet and all users.

Creation and reading process of application scope variables

- 1. Create the object of javax.servlet.servletContext by using getServletContext() method of generic servlet. The object of servlet context represent all the servlets present in the servlet container.
- 2. Use setAttribute() of servlet context to create application scope variable. It accepts two parameters as the name of the variable and value of the variable.
- 3. Use getAttribute() of servlet context to find value of application scope variable. If the specified variable does not exits then it returns null.

Properties or characteristics of Java Beans:

Every JavaBeans class contains simple property, boolean property and indexed properties.

- A simple property is one in which a method takes and returns elementary or single value (set of set and get methods are known as simple properties of a JavaBeans class.)
- A boolean property is one in which a method takes or return boolean value.
- An indexed property is one in which a method takes or return array of values.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

- The setProperty and getProperty action tags are used for developing web application with Java Bean. In web devlopment, bean class is mostly used because it is a reusable software component that represents data.
- The jsp:setProperty action tag sets a property value or values in a bean using the setter method.

Develop a JavaBeans class which will check whether the username and password correct or not?

Answer:

Test.java:

```
package abc;
public class Test {
    String uname;
    String pwd;

public void setUname(String uname) {
        this.uname = uname;
    }

public void setPwd(String pwd) {
        this.pwd = pwd;
    }

public String getUname() {
        return (uname);
    }

public String getPwd() {
        return (pwd);
    }

public boolean validate() {
```

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

```
if (uname.equals("kalpana") && pwd.equals("test")) {
    return (true);
} else {
    return (false);
}
```

Note:It is highly recommended to use a JSP page with scriptless.

The following table summarizes all operators used by the JSP Expression Language (EL):

Category	Operators
Variable	.and []
Arithmetic	+,-(binary), *, / and div, %and mod,-(unary)
Logical	and, &&, or, ,not,!
Relational	==, eq, !=, ne, <, lt, >, gt, <=, le, >=, ge
Conditional	A?B:C
Empty/Null	empty
checking	

Let's see each operator in details with usage examples.

1. Variable Operators

In EL, the dot (.) operator is used to access properties of a JavaBean object or get value of a key in a Map object. Syntax is:

\${objectName.propertyName} \${mapObject.keyName}

For example:

• Access the parameter named "username" in the URL query string (paramis an implicit map object that stores request parameters):

\${param.username}

 Retrieve hostname of client which is stored in HTTP request headers (header is an implicit map object that stores request headers):

\${header.host}

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Read the property called email of a JavaBean object named userBean: \${userBean.email}

The [] operator does same thing as the . operator, however, it can be also used to access elements of an array or a list by specifying an index. For example:

o The above examples can be re-written as follows:

\${param["username"]}

\${header["host"]}

\${userBean["email"]}

 $_{\odot}$ Retrieve value of the 2^{nd} element in an array named jobTitles: ["1"]

• Access email of the 2nd user in a list of users:

\${listUsers[1].email}

It's also possible to access nested property like this:

\${userBean.currentJob.title}

2. Arithmetic Operators

The following table describes all the arithmetic operators as well as examples:

Operator	Meaning	EL Expression Example	Result
+	Addition	\${100 + 200}	300
- (binary)	Subtraction	\${100 - 200}	-100
*	Multiplication	\${9 * 9}	81
/ or div	Division	\${200 / 100}	2.0
		\${200 div 100}	2.0
		\${200 / 0}	Infinity
% or mod	Remainder	\${2013 % 100}	13
		\${2013 mod 100}	13
		\${200 mod 0}	Exception
- (unary)	Negation	\${-20}	-20
		\${20 - (-20)}	40

Note that while the division operator can divide by zero, the remainder operator not. If we try this:

 $\{200 \bmod 0\}$

Then this exception will be thrown:

(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

java.lang.ArithmeticException: / by zero

Also note that in an arithmetic expression, a null is treated as zero. For example in the following expression:

 $\{user.age + 50\}$

If the variable user or age evaluates to null, then result of the expression is 50.

3. Logical Operators

The following table describes the logical operators along with examples:

Operator	Meaning		EL	Expression	Result
			Example		
and, &&	Logical AND		\${true && f	false}	false
			\${false and	false)	false
or,	Logical OR		\${true fals	e}	true
			\${false or fa	ılse}	false
not,!	Unary	boolean			true
	complement		\${!true}		false

Note that in a logical expression, a null is treated as false. Consider the following example: \${user.logged && true}

If the variable user or logged evaluates to null, then result of the expression is false.

4. Relational Operators

The following table describes the relational operators as well as examples:

Operator	Meaning	EL Expression Example	Result
==, eq	Equality test	\$ {100 == 100}	true
		\${101 eq 102}	false
!=, ne	Un-equality test	\${100 != 200}	true
		\${200 ne 200}	false
<, lt	Less than test	\${9 < 10}	true
		\${9 lt 8}	false
>, gt	Greater then test	\${10 > 9}	true
		\${10 gt 20}	false
<=, le	less than or	\${100 <= 200}	true
	equal test	\${100 le 100}	true
>=, ge	greater than or	\${100 >= 200}	false
	equal test	\${100 ge 100}	true

Note that there is no = operator in EL.



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

5. Conditional Operators

The conditional operators ? and : must participate in this ternary form:

A?B:C

If A evaluates to true, then evaluate B, else evaluate C. For example:

 $\{(8+9) > 10? (11+12): (13+14)\}$

Result: 23

6. Empty Operator

The **empty** operator checks a value is null or empty. Syntax:

\${empty *value*}

It returns true if value is null or empty, and returns false otherwise. For example:

\${empty param.username}

The above expression returns true if the parameter username is null or a empty String. Another example:

\${empty listUsers}

The variable listUsers is an ArrayList, the expression returns true if the list has no element (empty).

7. Operators Precedence

Following is the precedence of operators, with priority descends from top to bottom and from left to right in a row:

()
()
(unary) not! empty
*/div% mod
+-(binary)
<> <= >= lt gt le ge
==!= eq ne
&& and
|| or
?:



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478 ↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No–(0281)2471645

Examples:

EL Expression Example	Result
\${2 * 4 + 3 * 4}	20
\${8 + 10 / 5 - 3}	7
\$ {5 > 3 && 4 > 6}	false
$\{1+2>3?4*5:6-7\}$	-1

JSP - Standard Tag Library (JSTL)

The JavaServer Pages Standard Tag Library (JSTL) is a collection of useful JSP tags which encapsulates the core functionality common to many JSP applications.

JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization tags, and SQL tags. It also provides a framework for integrating the existing custom tags with the JSTL tags.

Install JSTL Library

To begin working with JSP tages you need to first install the JSTL library.

To use any of the libraries, you must include a <taglib> directive at the top of each JSP that uses the library.

Classification of The JSTL Tags

The JSTL tags can be classified, according to their functions, into the following JSTL tag library groups that can be used when creating a JSP page

- Core Tags
- Formatting tags
- SQL tags
- XML tags
- JSTL Functions



(AFFILIATED TO SAURASHTRA UNIVERSITY & GTU)

2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

↑3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot - 360001 Ph.No-(0281)2471645

Core Tags

The core group of tags are the most commonly used JSTL tags. Following is the syntax to include the JSTL Core library in your JSP –

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

Following table lists out the core JSTL Tags –

S.No.	Tag & Description
1	<c:out></c:out>
	Like <%= >, but for expressions.
2	<c:set> Sets the result of an expression evaluation in a 'scope'</c:set>
3	<c:remove> Removes a scoped variable (from a particular scope, if specified).</c:remove>
4	">c:catch> Catches any Throwable that occurs in its body and optionally exposes it.
5	<c:if> Simple conditional tag which evaluates its body if the supplied condition is true.</c:if>
6	<pre><c:choose> Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>.</otherwise></when></c:choose></pre>



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

	<c:when></c:when>
7	Subtag of <choose></choose> that includes its body if its condition evalutes to 'true' .
	<c:otherwise></c:otherwise>
8	Subtag of <choose></choose> that follows the <when></when> tags and runs only if all of the prior conditions evaluated to 'false' .
	<c:import></c:import>
9	Retrieves an absolute or relative URL and exposes its contents to either the page, a String in 'var', or a Reader in 'varReader'.
	<c:foreach></c:foreach>
10	The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality.
11	<c:fortokens></c:fortokens>
	Iterates over tokens, separated by the supplied delimeters.
12	<c:param></c:param>
	Adds a parameter to a containing 'import' tag's URL.
13	<c:redirect></c:redirect>
13	Redirects to a new URL.
1 /	<c:url></c:url>
14	Creates a URL with optional query parameters



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

†3 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot - 360001** Ph.No-(0281)2471645

One Mark Questions:-

1 Word Question Answer			
Sr. No	Question	Answer	
1.	JDBC Stands for?	Java Database Connectivity	
2.	ODBC stands for?	Open Database connectivity	
3.	The result is a set of class given in which package?	java.sql package	
4.	Which package gives us to facility to connect to every database?	java.sql	
5.	sun Microsystems has accepted which driver as Universal driver	ODBC	
6.	Which package is developed by java?	sun.jdbc.odbc	
7.	is designed to keep things simple while allowing advanced capabilities when required.	JDBC	
8.	cannot be directly used with Java.	ODBC	
9.	requires manual installation.	ODBC	
10.	API Is collection of methods, classes and interfaces.	JDBC API	

1 Word Question Answer			
Sr.No	Question	Answer	
1.	JDBC Architecture describes the interaction of JDBC API with?	Java application and java applet	
2	is responsible for transferring data between an application and a database.	JDBC API	

1 Word Question Answer			
Sr.No.	Question	Answer	
1.	driver Communicates directly with database server?	Java/Native Code	
3.	are completely written in java to achieve platform indepence	JDBC Driver/Native Protocol	



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

4driver can be downloaded dynamically? JDBC Driver/Native Protoco

1 Word Question Answer			
Sr.No.	Question	Answer	
1.	is used to create a Statement object for sending SQL	createStatement()	
	Statements to the database		
2.	is used to execute this statements multiple times.	prepareStatement()	
3.	is used to create a Callable Statement object for calling stored procedure of database.	prepareCall ()	
4.	used to release connection.	close ()	
5.	is used to save all the state of the database which are changed by transaction	commit ()	
6.	is used to cancel the effect of current running transaction.	rollback ()	

1 word Question Answer		
Sr.No.	Question	Answer
1.	DriverManager class belongs to which package?	java.sql
2.	DriverManager class has one method which is used to establish connection with different database server.	getConnection()
3.	is used to fetch records from database using select query.	ExecuteQuery ()
4.	is used to perform insert, update, and delete operation on record of database	executeUpdate ()
5.	Statement is used when they returns Multiple result.	Execute()
6.	is derived from Statement interface.	Prepared Statement interface
7.	Which statement is useful to called stored procedure from within a J2EE object.	Callable



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

8.	Which parameters callable statement uses?	IN, OUT & INOUT
9.	parameter is a single parameter that is use for both that is to pass information to stored procedure and to retrieve information from stored procedure.	INOUT
10.	is used to access the data of the table from database.	ResultSet
11.	How many methods of ResultSet object that are used to position the virtual cursor?	Six
12.	Which interface describes the database as whole?	DatabaseMetaData

1 Word Question Answer		
Sr.No.	Question	Answer
1.	RMI stands for?	Remote Method Invocation
2.	Which technology which allows the client to Remote object communication and object to object communication between different JVM?	RMI
3.	is the best example of distributed module.	RMI
4.	All communication between remote objects is handled by?	RMI
5.	Remote communication look similar to regular Java?	Method Invocation

1 Word Question Answer		
Sr.No.	Question	Answer
1.	defines the how the client request to the server for	RMI Architecture
	the remote objects and how the server processes for that	
	request.	
2.	RMI architecture divided intolayers?	Four
3.	simply keeps track of addresses of remote objects	Registry
	that are being exported by the application and also name	
	of the remote objects.	
4.	consists of client side and server side java program	Application Layer
	with remote objects.	



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

5.	layer is responsible for listening to the remote	Stub/skeleton
	methods calls made by client and redirecting them to the	
	server.	
6.	layer consists of stub and a skeleton layer.	Proxy layer
7.	The stub and skeleton are created using	RMI Compiler
		(RMIC)
8.	is server Side proxy that communicates with the	Skeleton
	stub.	
9.	is an interface between proxy layer and Transport	The Remote
	layer.	Reference layer
10.	interpretes the references made by the client to the	The Remote
	remote object on the server	Reference layer
11.	Handle replicated object.	RRL
12.	takes path of the communication or layer between	Transport layer
	client & server.	

1 Word Question answer			
Sr. No.	Question	Answer	
1.	Thereceives any results from remote method and returns it to the client.	Stub	
2.	is imported in the entire program in order to communicate with server.	Java.rmi.*	
3.	returns name of the remote object specified by the RMI URL.	lookup()	



2 – Vaishalinagar Nr. Amrapali Ünder Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

Sr. No	Question	Answer
1.	Package contains the generic interface and classes that are implemented & extended by all servlets.	javax.servlet
2.	It provides the runtime environment to the servlet.	ServletContext Interface
3.	provides the methods for handling HTTP request and HTTP response.	HttpServlet Class
5.	is called by servlet container to initialize the servlet itself.	init()
6.	It used to read client request and write the response data to the client.	service()
7.	method is called by servlet Container automatically when objects or resources like memory, thread etc need to be destroyed or servlet stops it execution.	destroy()
9.	In CGI single instance creates multiplein order to process multiple request and response	Threads
10.	SingleThreadModel is implemented by	SingleThreadModel interface
11.	is called by servlet via service () to handle an Http request.	doGet()

Sr.No	Question	Answer
1.	Which exception indicates that a servlet problem	ServletException
1.	has been occurred?	bei viethzeeption
2.	UnavailableException which extends	ServletException
3.	returns the exception that caused this	getRootCause ()
	servlet exception.	
4.	returns the servlet that threw this exception	getServlet()
	or null if the servlet instance was not provided to	
	the constructor.	



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road Rajkot – 360001 Ph. No–(0281) 2440478

Sr.No.	Question	Answer
1.	Returns the value of the attribute named name.	getAttribute()
2.	Returns the value of the parameter.	getParameter()
3.	Returns the String equivalent of the client host name.	getRemoteHost()
4.	Returns a PrintWriter that can be used to write character data to the response.	getWriter()
5.	Redirects the client to url.	sendRedirect()

Sr. No.	Question	Answer
1.	is temporary small unique connection between a server and the client enabling it to identify that user across multiple page requests.	Session
2.	To maintain the session state theis the best practice.	URL rewriting
3.	The component can include in the html form a field that is not displayed on the form. This field is called	hidden field
4.	are small bits of textual information that a Web server sends to a browser and that the browser later returns unchanged when visiting the same Web site or domain.	Cookies
6.	A public interface HttpSession is a member of	Javax.servlet.http
7.	provides a way to identify a user across more than one page using the session in the servlet.	HttpSession



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

Sr. No.	Question	Answer
1.	Sometimes servlets have to communicate with other web resources is called	Servlet Collaboration
2.	defines an object they receives request from the client and send them to any resources such as servlet, html file call JSP file and the server.	Request Dispatcher
3.	should be called before the response has been committed to the client.	Forward()
4.	includes the contexts of resource in the response.	Include()
5.	defines mappings between URL paths and the servlets that handle requests with those paths.	Web.xml

	Question	Answer
1.	JSP stands for?	Java Server Pages
2.	Technology enables you to mix regular, static HTML with dynamically generated content.	JSP
3.	In advance JSP we can use some custom tags for	Looping.
4	used for developing dynamic web content.	JSP and servlet both
5	A servlet is a java class implementing the	javax.sevlet
6	contain mixture of HTML, javascript.	JSP
7	Some of the JSP functionality can be achieved on the client, using	Javascript
8	enables you to write standard HTML pages containing tags that run powerful programs based on Java Technology.	JSP
9	After JSP page converted in to servlet page which code is created?	Bytecode
10	When jsp servlet instane is createdis called.	jspInit()
11	When jsp servlet instance is called is called where request and response object are sent.	_jspService()
12	method is called when the JSp servlet instance is destroyed from the web container.	jspDestroy()



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

13	Directive is uses to specify attributes for the whole JSP page	Page
14	directive is used to include the static page and dynamic pages with the other jsp pages.	Include
15	Directive declares that the JSP file uses custom tags	Taglib
16	Elements are used to write java code with the jsp file.	Scripting Elements
17	are block of java code with the jsp page.	Scriptlets
18	Scriptlets starts withtag and end withclosing tag.	<%,%>
19	Declaration starts with tag and end with closing tag.	<%!,%>
20	is used to print value of any variable or any valid expression when the jsp page is requested.	Expression element
21	An expression starts with and ends with .	<%=,%>
22	Element is used to include static and dynamic resource of the current JSP page.	<jsp: include=""></jsp:>
23	Control is used to transfer control from current JSP page to another source which may be any valid source of application.	<jsp:forward></jsp:forward>
24	element is used to embed an applet and java beans with the jsp page	<jsp:plugin></jsp:plugin>
25	The lifetime and accessibility of an object is known as	Scope
26	Jsp object created using the scope can be accessesd from any pages that serve that request	request
27	Implicit object request has scope	Request
28	Objects with scope is available to all application components that participate in the client's session.	session
29	Objects with application scope are available to the entire for the life of the application	Application
30	objects means the object which is created already created by JSP itself.	Implicit
31	Request object in JSP also belongs to class.	javax.servlet. httpServletRequest
32	If you want to generate any output for client,the you can generate it using object	Response
33	object is used for session tracking	Session
34	object is used for set values and attributes at application level.	Application



2 – Vaishalinagar Nr. Amrapali Under Bridge Raiya Road **Rajkot – 360001** Ph. No-(0281) 2440478

35	ol output stream of the client	bject is used to write output to the	Out
36	re	eferred to the error occurred at run time.	Exception
37	. In jsp exception is of	bject with page scope.	Implicit
38	cl exception and error classes	ass is the super class of all the in the java.	Throwable
39	To create an error page with JSP we can useattribute of page directive.		isErrorpage=true
40	The web application deploy	ment descriptor uses the tag.	<error–page></error–page>