

# J2EE Grooming Class

## JDBC Grooming

### 1. Define Jar File and Explain the need for Jar File.

Jar means Java Archive(Compress). It is a file format based on zip file format which is used to compress many files into one single file.

Jar file is needed to import the properties based on the requirement.

### 2> Define the following

1. Abstraction

2. Interface

3. Loose-Coupling

4. Tight-Coupling

5. NullPointerException

6. Costly Resource.

#### 1. Abstraction:

Hiding the implementations and providing the functionalities to the user with the help of interface is known as abstraction.

#### 2. Interface

Interface is a media to communicate between user and any device.

#### 3. Loose-Coupling:

Change in the implementations which does not affect the user is known as loose-coupling.

#### 4.Tight-Coupling:

Change in the implementations which affects the user is known as Tight Coupling .

#### 5.NullPointerException:

Pointing towards an object which is not present throws an exception is called as NullPointerException.

#### 6.Costly Resources:

Resources which makes use of system properties in the form of stream are known as costly resources.

3>Name the Driver classes provided by the respective database servers along with their port numbers.

DataBase Server	Driver Class	Port Numbers
Oracle	OracleDriver	1521
MySQL	Driver	3306
MS-SQL	SqlServerDriver	1433
Derby	EmbeddedDriver	1527

#### 4.Define API?

API stands for Application Programming Interface. API is used for inter-application communication that is one application can communicate with other application with the help of API to achieve loose-coupling.

Ex:Apache-POI,Jaxcel,JDBC,Servlet,etc

### 5.Why JDBC Driver is an implementation of JDBC API?

Since all the Driver classes must mandatorily implements java.sql.Driver Interface which is part of JDBC API.

Hence we conclude that JDBC Driver is an implementation of JDBC API.

### 6>What are the specifications of JDBC?

\*There are 3 different specifications present namely

1.All the Driver Classes must contain one Static block in it.

2.All the Driver Classes must mandatorily implements java.sql.Driver interface.

3.All the Driver Classes must mandatorily be registered with the DriverManager by using a static method called registerDriver().

### 7>Define JDBC and mention the Steps and advantages of JDBC?

Def:Java DataBase Connectivity which is a specification(Detailed Description) given in the form abstraction API to achieve loose-coupling between Java Application and the DataBase server.

### Steps of JDBC:

There are 6 standard steps of JDBC:

- 1.Load and Register the Driver(Driver Class)
- 2.Establish a Connection with the DataBase Server.
- 3.Create a Statement or Platform
- 4.Execute the SQL queries /SQL Statements
- 5.Process the Resultant Data(Optional).
- 6.Close all the costly Resources.

### Advantages:

- 1.We can Achieve loose-coupling between Java Application and the DataBase Server.
- 2.Platform Independent.

8>What is the outcome any executed DML and DQL Query?

\*The outcome of DML is 0 to n integer value which gives the total number of records affected in the database server.

\*The outcome of DQL is processed or Resultant data which is stored in the cursor or buffer memory which can be fetched with the help of ResultSet interface which is a part of JDBC API.

9>Can we deal with multiple records using Statement interface.?

Yes but it is not a good practice to deal with multiple records by using Statement interface since the compilation takes place each time along with the execution the performance of an application decreases.Hence it is better to make use of PreparedStatement interface.

10>Explain execute() ,executeUpdate() and executeQuery() in brief..

**execute():**

\*execute() is a generic method since it is used to execute any type of SQL queries or SQL Statements.

Hence ,the return type for execute() is boolean where it returns a boolean value called true in case of DQL and false in case of DDL/DML.

**executeUpdate():**

It is a specialized method which is used to execute only DML Queries or DML Statements.Hence,the return type for executeUpdate() is integer.

**executeQuery():**

`executeQuery()` is a specialized method which is used to execute only DQL query. Hence the return type for `executeQuery()` is `ResultSet` interface.

11> Which method is used to fetch the processed or Resultant data from the Cursor/Buffer Memory ?

By using `getXXX()` methods of `ResultSet` Interface.

There are 2 overloaded variants of `getXXX()` are present

1. `getXXX(int column number)`

2. `getXXX(String column name)`

12> Which method is used to set the data for a place holder?

By using `setXXX()` .

\*Method Signature:

+`void setXXX(int place holder number ,xxxdata)`

13> What is the return type for `getXXX` and `setXXX()`?

By default the return type for `getXXX()` are the respective data type

For `setXXX()` is `void`.

14. Explain `next()` and `absolute()` in brief.

**next():**

\*next() is used to check whether the next record is present in the cursor /buffer memory or not and it return a boolean value called true or false but not the record.

\*next() can be used whenever there are minimum number of records present in the cursor/buffer memory.

**absolute(int row number):**

\*It is used to check whether a particular record is present in the cursor /buffer memory or not based on the parameter called int row number and it returns a boolean value called true or false but not the record.

\*absolute() can be used whenever there are n-number of records present in the cursor/buffer memory.

**15>Define place holder and explain the rules to set the data for a place holder.**

Place Holder is a parameter which holds dynamic values at the runtime by the user.

Place Holder is represented as ?

**\*Rules to set the data for a place holder**

1>We need to set the data for a place holder before the execution.

2>The number of data must exactly match the number place holder.

3>We need to set the data for a place holder by using setXXX().

16>Whenever the number of data exceeds the number of place holder what is the output.

Ex:

```
String qry="insert into btm.student values(?,?,?)";  
pstmt.setInt(1,10);  
pstmt.setString(2,"Arun");  
pstmt.setDouble(3,78.40);  
pstmt.setDouble(4,89.56);
```

Output:SQLException

17>Whenever two place holders have the same place holder number what is the output?

Ex:

```
String qry="insert into btm.student  
values(?,?,?)";  
pstmt.setInt(1,10);  
pstmt.setString(2,"Arun");  
pstmt.setDouble(3,78.40);  
pstmt.setDouble(3,89.56);  
output:10/Arun/89.56
```



### 18>In general is Statement interface is faster in performance or PreparedStatement?

In general PreparedStatement is faster in performance compared to Statement interface for two important reasons

1>Since PreparedStatement interface supports the concept of Place Holders to take dynamic values at the runtime by the user.

2>Since PreparedStatement interface supports the concept of compile once and execute many times [Execution Plan].

### 19>What are the different types of Statements present in JDBC?

There are 3 different types of Statements present in JDBC namely

1>Statement

2>PreparedStatement

3>CallableStatement

Syntax:

```
Statement stmt=con.createStatement();
```

```
PreparedStatement pstmt=con.prepareStatement(Qry);
```

```
***CallableStatement cstmt=con.prepareCall(Qry);
```

### 20>Why PreparedStatement and CallableStatement are called as Pre-Compiled Statement?

Since the query is passed at the time of implementation object creation of these interfaces but not in the execute method .Hence the name Pre-compiled Statement.

## 21>What is the role of Place Holder In case of PreparedStatement and CallableStatement interface.

String qry="select username from btm.user where name=? and password=?";

\*In case of PreparedStatement interface place holder is used to take dynamic values at the runtime by the user.

\*In case of CallableStatement interface place holder is used to take the values for IN and OUT parameters .

## Servlet Grooming

### 1>Define Server and Explain the types of Server ?

Server is a software which manages all the resources along with which it process the client request and Serve the client request .

There are different types of Servers Present namely

#### 1.DataBase Server:

It is used to deal with the data.

Ex:Oracle,MySQL,MS-SQL,Derby ,Mongo db,etc...

## 2>Application Server :

Application Server is used to execute a dynamic Application or a realtime application.

Ex:JBoss,OracleWebLogic,IBMWebSphere,etc

## 3>WebServer:

Web Server is used to execute only web-application.

Ex:Apache -Tomcat Server and Oracle GlassFish.

## 2>Define Deployment and Explain the types of deployment ?

Making all the resources available to the Server is known as Deployment .

\*There are 2 different types of Deployment present

### 1.Manual Deployment:

In case of manual deployment all the resources are made available to the Server manually.

### 2>Automated Deployment :

In case of Automated Deployment the resources are made available to the server automatically with the help of automated tools such as ANT,MAVEN,etc

By using this automated tools we create batch files(A file which is accessed based on double click is known as Batch file)

## 3>Define webapps and work folder of Apache-Tomcat Server.?

**webapps-** It is used to deploy all the web applications onto the Apache-Tomcat Server.

**work:**

It is used to store the information/data with respect to translated Servlet (Conversion of JSP into Servlet is known as translated Servlet.)

**4>What are the basic tasks performed by any server in general?**

1>Server is used to manage all the resources.

2>Provide the runtime environment to the JEE Container.

**5>Define JEE Container ?**

JEE container is an engine which is used to manage all the JEE components such as Servlet ,JSP ,EJB,etc.

**6>Explain the Structure of JEE Application in brief?**

1>Each and every application must mandatorily have only one web.xml present without which the JEE container fails to load an application where it throws http 404 error (Resources not available)- Client Error.

2>We can deploy multiple application onto one Single server but in this case each and every application must have unique name or different name.

3>Whenever we start the server all the applications are loaded sequentially one after the other by the JEE container.

4>At the time of application loading the JEE container parses (compiles) the web.xml where if there is any error present in the web.xml then the JEE container throws ParseException.

7>What are the different types of Logics associated with respect to a dynamic application?

There are 3 different types of Logics associated with respect to a dynamic application

#### 1>Presentation Logic:

Presentation Logic is used to present the contents on to an application

Technologies Used:HTML,CSS,JS,JSP,PHP,etc

#### 2>Persistence Logic:

Persist means to store .Persistence Logic is used to persist the data into the persistence system(DataBase).

Technologies Involved:JDBC,SQL,HIBERNATE,etc

#### 3>Business Logic:

It is used to perform the core functionality ie some set of calculation and validation operation on an application.

Technologies Involved:

Servlet and other framework components.

8>What is default port number for Apache-tomcat Server?

The default port number for Apache-Tomcat Server is 8080

But we can change the port number.

9>Explain Welcome file or Landing page?

A File or a page which is automatically displayed whenever a client uses an application is known as welcome file or a landing page.

- \*index is considered to be the default welcome file or landing page.

Since it is automatically loaded by the JEE container whenever client uses an application.

- \*We can explicitly make a file as welcome file or landing page by renaming it as index.

Note:

- \*Default port number for Apache-Tomcat Server :8080

- \*Default welcome file or landing page :index

- \*The default root tag for a xml file:<web-app>

- \*Default root tag for load-on-startup:<servlet>

- \*The default type of Http Request :Get

- \*The current version of xml used:1.0

10>Define web.xml and explain how are the configurations made in web.xml are understood by the JEE Container?

It is a configuration file in xml format which is used to store information with respect to the configurable resources of an application.

- \*All the custom tags or user defined tags are converted into 8-bit Unicode format so that the configurations made in web.xml are understood by the JEE Container.

11>Explain the different ways in which the resources can be configured?

There are 2 different ways in which the resources can be configured

1>By using web.xml

2>Through Annotation.

## 12>Define Servlet and explain the different types of Servlet?

Servlet is a server side java program which performs all the 3 different types of logics such as Presentation Logic ,Persistence Logic and Business Logic along with which it process the Http Client request and get back some http response.

There are 2 different types of Servlet are present

1>GenericServlet(G)

2>HttpServlet(H)

G:It is not specific to any protocol or independent of protocol ,hence the name GenericSevlet.

H:It is specific to a particular type of protocol called http protocol ,hence the name HttpServlet

G:It does not support session.

H:It supports session.

G:It contains 3 methods in it out of which one is an abstract method and other 2 are concrete methods.

H:It contains only concrete methods without any abstract methods in it .

G:It is an abstract class present in javax.servlet package.

H:It an abstract class present in javax.servlet.http package.

G:The abstract method present in GenericServlet is named as service() which has to be overridden for 2 important reasons .

1.Since service() is an abstract method which has to be overridden mandatorily .

2>Since service() is the only method which takes ServletRequest and ServletResponse as a parameter which is responsible for processing each and every client request.

H:In case of HttpServlet we need to override the respective concrete method called doXXX() method for a particular type of Http Request .

G:The other 2 methods of GenericServlet are init() and destroy()  
.overriding these two methods are optional since these are concrete methods.

H:There are 8 different types of Http Requests are present namely

1>POST

2>GET

3>PUT

4>DELETE

5>TRACE

6>OPTION

7>HEAD

8>CONNECT

13>Is there a service() present in HttpServlet or not?

Yes,In case of HttpServlet the service() is a concrete method but overriding this service() is optional.



14>Why a class made as abstract when it contains only concrete methods in it?

Ex:HttpServlet

So that further modifications made does not affect the user.

15>What are the properties to be configured for a servlet in web.xml?

\*There are 3 different properties to be configured for a servlet in web.xml

1>Servlet-name(unique)

2>URL Pattern(Unique)

3>Fully Qualified Class Name.

16>Define UI/Form Data and explain how to fetch UI/Form data?

The data which is entered by the enduser(client) on a form page and is submitted to the server in the form of key and value pair is known as UI/Form data.

The UI/Form data can be fetched by using getParameter method.

Method Signature:

**+String getParameter(String key)**

\*getParameter method takes String key as an argument

\*If the key is present then the method returns associated value.

\*If the key is not present then the method returns null but not any error or exception.

## 17>Define Servlet-LifeCycle and explain the different ways of beginning the Servlet Life-Cycle?

\*Servlet gets a life and starts its life cycle only when a Servlet object is created

Def:

Servlet Life-Cycle depicts or represents events or phases which takes place from Servlet object creation until Servlet object destruction.

\*Entire servlet-life-cycle is managed by JEE container .

\*There are 4 different phases of Servlet Life-Cycle

1>Instantiation or Object Creation Phase :

2>Initialization Phase

3>Service Phase

4>Destruction Phase

There are 2 different ways in which we can begin the Servlet Life-Cycle:

1>Whenever client makes a first request to a Servlet one Servlet object is created by the JEE container by calling the default constructor of Servlet now the Servlet Life-Cycle begins.

2>In case of load-on-startup.:

Load-on-startup is a sub tag of Servlet tag.

In case of load-on-startup the servlet object is created by the JEE container by calling the default constructor of Servlet at the time of **Server startup so** that the delay time made by the first client request can be avoided which helps in increasing the performance of an application.

## 18>Can a Servlet be made SingleThreaded?

By default the Servlet is always multithreaded but it can be made single-threaded in two different ways

1>By writing a Servlet class which implements a marker interface called `SingleThreadedModel` which is a deprecated interface(Outdated).

2>By making `service()` as synchronized.

If we say Servlet is `SingleThreaded` means only one thread can execute Servlet `service()`.

`SingleThreadModel` ensures that Servlet handles only one request at a time.

Since Servlet API 2.4 it is deprecated because it does not solve all the thread safety issues such as static variable and session attribute can be accessed by multiple thread at a time even if we implement the `SingleThreadModel`.

## 19>Explain the life-cycle methods of Servlet?

There are 3 different life-cycle methods of Servlet are present

### 1>`init(ServletConfig)`

`init()` is used to initialize the resources of that particular Servlet object by taking `ServletConfig` as a parameter.

\*`init()` is always called by the JEE container only once.

### 2>`service(ServletRequest req,ServletResponse resp)`

`service()` takes `ServletRequest` and `ServletResponse` as a parameter which is responsible for processing the client request.

\*This service() is always called by the JEE container for n-number of times.

### 3>destroy()

\*destroy() is called by the JEE container to close all the costly resources

\*destroy() is called only once but in two different situation.

1>Whenever we close an application destroy() is called to close all the costly resources related to that application.

2>Whenever we re-deploy an application on to the server the destroy() is called to close the previously used to costly resources related to that application.

### 20>How to configure Servlet in case of load-on-startup?

\*In case of load-on-startup the Servlet must mandatorily be configured with a positive integer value but the JEE container gives the priority based on lowest positive integer value.

\*Whenever we configure load-on-startup with a negative integer value then the Servlet is created based on first client request by neglecting the load-on-startup concept.

\*When two or more Servlets are configured with same positive integer value for load-on-startup then sequential execution takes place.

### 21>What is the difference between Post and Get Request

P: Post Request is used to post some contents or data from client to the server.

Ex: Ex:Posting Resume,Posting Profile in matrimonial site

G: Get Request is used to get some contents or resources from the Server.

P:Post Request deals with unlimited data.

G:Get Request deals with limited data.Ie 1024 Characters.

P:Post Request is non-idempotent

G:Get Request is Idempotent.

P:Post Request can not be bookmarked(Can not be saved).

G:Get Request can be bookmarked.

P: In case of Post Request the data are carried to the server as a part of Http Request Body which is not displayed to the endUcer(Client).Hence the data are secured.

G: \*In case of Get Request the data are carried to the server as a part of Request object in the form of key and value pair which is displayed even to the endUcer.Hence the data are not secured.

P:Whenever we deal with n-number of data then the type of request is Post Request.

G: Whenever we deal with a link then the type of request is get request.

P:Post Request is not safe.

G:Get Request is safe.

G:Whenever the type of request is not mentioned or not configured then by default the type of request is Get Request.