

@author @version @copyright @return @throws @param @WebServlet And @WebFilter

preload means already define data on jsp

SQL COMMIT command

SQL SAVEPOINT command
e unique name in the p

SQL ROLLBACK command
e since the begin trans

The SAVEPOINT Command
A SAVEPOINT is a point
rolling back the entire tra

1. ***Life cycles of jsp & Servlet.***
2. ***Session created, destroy & tracking?***
3. ***Utility classes?***
4. ***Exception Classes?***
5. ***BaseCtl Methods and their return types & Defination.***
6. ***Mail & Email***
7. ***Interface?***
8. ***Maven?***
9. ***Github - Local to Global & Global to Local?***
10. ***JSP Implicit object?***
11. ***Server Difference - web & application?***

Important Questions

1. ***Life cycles of jsp & Servlet?***
2. ***Session created, destroy & tracking?***
3. ***Utility classes?***
4. ***Exception Classes?***
5. ***BaseCtl Methods and their return types & Defination?***
6. ***Mail & Email?***
7. ***Interface?***
8. ***Maven?***
9. ***Github - Local to Global & Global to Local? & Defination?***
10. ***JSP Implicit object?***
11. ***Server?***
12. ***StringBuffer and StringBuiler? And their Differences?***
13. ***Session Creation?***
14. ***Session destroy?***
15. ***Session Tracking?***

(Q1) Which Archytecture is followed by ORS ?

MVC :) MVC is framework methodology . It seperates code implmentation of an application into three component Model , View , Controller .

Model :- Model contain business logic, data acess logic, integration logic to perform business operation and manupulate database .

View :- View component is responsible to render the grafical user inteface .

Controller :- Controller contain navigation logic and are responsible to perform business operation submitted by view with the help of model .

Q. Types of LOGIC?

- 1. Presentation logic:** This is the set of control statement that decide look of user interface .(JSP)
- 2. Control logic:** This is the set of control statement that decide Navigation of user inteface.
- 3. Business logic:** This is set of control statement that perform business operation. Like (Change password , ForgetPassword)
- 4. Data Acess logic:** This is set of control statement that make changes in database with the help of JDBC. (JDBC,SQL)
- 5. Integration logic:** This is set of control statement that integrate application with another application or server .(EMAIL SEND)

Q. MVC Guideline?

MVC provides following guidelines to develop a web application:

1. One screen will have one view. For example login and change password screens will have `Login.jsp` and `ChangePassword.jsp` views.
2. One view will have one controller. For example `Login.jsp` and `ChangePassword.jsp` will have `LoginCtl.java` and `ChangePasswordCtl.java` controller servlets. When screens have large number of fields then screen can be divided into multiple sub-screens in this situation onecontroller will have multiple views.

1 Screen = 1 View = 1 Controller
3. User cannot directly access a view. View is always accessed by a user via controller. Controller is responsible to fetch data from model to display at view. For example `LoginCtl.java` will be called to display `LoginView.jsp` and `ChangePasswordCtl.java` will be called to display `ChangePasswordView.jsp`.
4. View always submits request to its own controller. For example `Login.jsp` always submits request to `LoginCtl.java` servlet and `ChangePassword.jsp` will always submit request to `ChangePasswordCtl.java` servlet.
5. Controller will perform business operations by calling model business methods and keeps model data into request scope and forward request to view. View will get model from request scope and read data to display at screen.
6. When an application needs to navigate to the next view then request is forwarded to controller of next view. For example after authentication if `Welcome.jsp` view has to be displayed then `LoginCtl.java` will forward request to `WelcomeCtl.java` servlet instead of `Welcome.jsp`.

Spring MVC, Struts, JSF and many more open source java frameworks are based on MVC architecture.

Q3. Design pattern ?

Standard problem has standard solution .

1. Singletone design pattern :-

The class that has one instance in their lifetime called singletone design pattern .

2. Builder design pattern :-

Create complex object using simple object using step by step approach .

3. Factory design pattern :-

The class that has ability to create object of another class is called factory class that follow factory design pattern.

4. Front controller design pattern :-

Main controller perform session checking and login operation before calling any application controller.
It Prevent any user to access an application without login .

Q. Which class we make singletone class ? And how to make ?

JDBC datasource class

There are four steps to make single tone class :

1. Make a class Final so that child can not be created for single class.
2. Make a default Constructor Private so that no one other class can instantiate single class .
3. Declare a Static Variable of self type in single class static variable have only one copy in their lifetime .
4. Make a get Instance() Static method in singletone class that will return instance of same single class .

Q5. Acid Properties?

1. Atomicity means all or nothing.
2. Consistency means it make sure that data is in consistant state .
3. Isolation means Transaction are independent .
4. Durability means committed data never lost .

Atomicity:It means all or nothing. All the database changes made by a transaction will be committed or aborted. A transaction can also be rolled back in case of:

1. Deadlocks
2. Database or Application software failures
3. CPU or Disk failures

Consistency:It makes sure that database will be in consistent state. A transaction must always leave the system in a consistent state, no matter how many concurrent transactions are running at any given time.

Isolation: An Application may execute multiple business operations at a time that will run concurrent transactions. Isolation brings benefit of hiding uncommitted changes (also known as dirty data) of a transaction from other concurrent transactions.

Durability: It says a successful transaction must permanently change the state of a system, and before ending it, the state changes are recorded in a persisted transaction log. If our system is suddenly affected by a system crash or a power outage, then all unfinished committed transactions may be replayed.

Q. What is DCP?

Ans: Data Connection Pool is a pool of connections. It is created at the application startup. Ideally, one DCP is created for one Database. If your application has multiple Databases then multiple DCP will be created, one for each Database.

Steps of making DCP

- (1) Add c3p0(0.9.5) dependency in pom.xml.
- (2) JDBC Datasource class ko singleton banaya hai.
 - (2.1) ComboPooledDataSource (cpds) ka object banaya hai.
 - (2.2) cpds = null kiya
 - (2.3) cpds me sare parameter ko set kiya

And (2.4) connection provide kiya (return getInstance().cpds.getConnection();)

- (3) System.properties pr parameter liye hai :-

url = jdbc:mysql://localhost:3306/project4

driver = com.mysql.cj.JDBC.Driver

username = root

password = root

Database = JDBC

Service = javaBean

acquireIncrement = 10

initialPoolSize = 10

maxPoolSize = 100

minPoolSize = 10

timeout = 10

```

1    ## Character '#' is used for one line comment in properties file
2
3    #Organization
4    org=SunilOS
5
6    #Generic error messages
7    error.require={0} is required
8    error.integer={0} must be an integer
9    error.date={0} is invalid date
10   error.email={0} is invalid Email Id
11   error.name={0} must contains alphabet only
12
13  #Database connection Parameters
14  url=jdbc:mysql://localhost:3306/project4
15  driver=com.mysql.cj.jdbc.Driver
16  username=root
17  password=root
18  DATABASE=JDBC
19  service=javaBean
20  acquireIncrement = 10
21  initialPoolSize = 10
22  maxPoolSize = 100
23  minPoolSize = 10
24  timeout = 10
25
26
27  #Email Server Parameters
28  smtp.server=smtp.gmail.com
29  smtp.port=465
30  email.login=dharammtr1998@gmail.com
31  email.pwd=u2xn mgus mcyz utmu
32
33
34
35  #Pagination
36  page.size=10
37

```

```

<dependency>
  <groupId>com.mchange</groupId>
  <artifactId>c3p0</artifactId>
  <version>0.9.5.2</version>
</dependency>

```

```

19  public final class JDBCDataSource {
20
21      /**
22       * JDBC Database connection pool ( DCP )
23       */
24
25
26      private static JDBCDataSource datasource;
27
28      private JDBCDataSource() {
29      }
30
31      private ComboPooledDataSource cpds = null;
32
33      /**
34       * Create instance of Connection Pool
35       *
36       * @return
37       */
38      public static JDBCDataSource getInstance() {
39          if (datasource == null) {
40
41              ResourceBundle rb = ResourceBundle
42                  .getBundle("com.rays.proj4.resourcesB.System");
43
44              datasource = new JDBCDataSource();
45              datasource.cpds = new ComboPooledDataSource();
46
47              try {
48                  datasource.cpds.setDriverClass(rb.getString("driver"));
49              } catch (Exception e) {
50                  e.printStackTrace();
51              }
52
53
54              datasource.cpds.setJdbcUrl(rb.getString("url"));
55              datasource.cpds.setUser(rb.getString("username"));
56              datasource.cpds.setPassword(rb.getString("password"));
57              datasource.cpds.setInitialPoolSize(new Integer((String) rb
58                  .getString("initialPoolSize")));
59              datasource.cpds.setAcquireIncrement(new Integer((String) rb
60                  .getString("acquireIncrement")));
61              datasource.cpds.setMaxPoolSize(new Integer((String) rb
62                  .getString("maxPoolSize")));
63              datasource.cpds.setMaxIdleTime(DataUtility.getInt(rb
64                  .getString("timeout")));
65              datasource.cpds.setMinPoolSize(new Integer((String) rb
66                  .getString("minPoolSize")));
67
68          }
69          return datasource;
70      }
71
72      /**
73       * Gets the connection from ComboPooledDataSource
74       *
75       * @return connection
76       */
77      public static Connection getConnection() throws Exception {
78          return getInstance().cpds.getConnection();
79      }
80
81      /**
82       * Closes a connection
83       *
84       * @param connection
85       * @throws Exception
86       */
87      public static void closeConnection(Connection connection) {
88          if (connection != null) {
89              try {
90                  connection.close();
91              } catch (Exception e) {
92                  e.printStackTrace();
93              }
94          }
95      }
96
97      public static void rollback(Connection connection)
98          throws ApplicationException {
99          if (connection != null) {
100              try {
101                  connection.rollback();
102              } catch (SQLException ex) {
103                  throw new ApplicationException(ex.toString());
104              }
105          }
106
107      }

```

Q. Tools, Dependencies and Technologies used in your Application?

Tool: 1. Eclipse mars

2. MySQL 9.1.0

3. ER win 14.0

4. Agro UML 0.34 (2011)

latest version

Dependencies:

1. Servlet (3.1.0)
2. MySQL (8.0.29)
3. Log4j (1.2.17)
4. c3p0 (0.9.5)
5. Java Mail (1.4.7)

Technologies:

1. JDK (11) - 21
2. JSP(3.2) Servlet (4.2) 6.0
3. JDBC (4.2) - 4.3
4. MySQL (8.0.29) - 8.0.32
5. Java Mail APT (1.4.7)
6. log4j (1.2.17)

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>in.co.rays</groupId>
6     <artifactId>ORS_Project</artifactId>
7     <version>0.1-SNAPSHOT</version>
8     <name>ORS_Project Maven Webapp</name>
9     <url>http://maven.apache.org</url>
10    <dependencies>
11      <dependency>
12        <groupId>junit</groupId>
13        <artifactId>junit</artifactId>
14        <version>3.8.2</version>
15        <scope>test</scope>
16      </dependency>
17
18
19
20
21      <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
22      <dependency>
23        <groupId>mysql</groupId>
24        <artifactId>mysql-connector-java</artifactId>
25        <version>8.0.26</version>
26      </dependency>
27
28
29      <!-- https://mvnrepository.com/artifact/log4j/log4j -->
30      <dependency>
31        <groupId>log4j</groupId>
32        <artifactId>log4j</artifactId>
33        <version>1.2.17</version>
34      </dependency>
35
36
37      <dependency>
38        <groupId>javax.mail</groupId>
39        <artifactId>mail</artifactId>
40        <version>1.4.7</version>
41      </dependency>
42
43
44      <!-- https://mvnrepository.com/artifact/com.mchange/c3p0 -->
45      <dependency>
46        <groupId>com.mchange</groupId>
47        <artifactId>c3p0</artifactId>
48        <version>0.9.5.2</version>
49      </dependency>
49
50
51      <!-- <dependency><groupId>com.sun.mail</groupId><artifactId>javax.mail</artifactId>
52        <version>1.6.8</version> </dependency> -->
53
54
55      <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
56      <dependency>
57        <groupId>javax.servlet</groupId>
58        <artifactId>javax.servlet-api</artifactId>
59        <version>3.1.</version>
60        <scope>provided</scope>
61
62
63      </dependency>
64
65      <build>
66        <finalName>ORS_Project</finalName>
67
68        <plugins>
69          <plugin>
70            <artifactId>maven-compiler-plugin</artifactId>
71            <configuration>
72              <source>11</source>
73              <target>11</target>
74            </configuration>
75          </plugin>
76        </plugins>
77      </build>
78
79    </project>
```

Q. WebServer name?

1. Apache Tomcat
2. Resine
3. Kclone
4. Oracle web tier
5. jigsaw

Q. Application Server name?

1. Jboss
2. Websphere
3. WebLogic
4. Jetty
5. GlassFish

Q. Types of validation in the project?

There are two types of validations in our Projects.

1. ServerSideValidations

1.1. InputValidations

Prograttive Validations.
Declarative Validations.

1.2. Business Validations(Check from Database)

2. ClientSide Validations Use JavaScript



Q. What is Log4j?

A: Log4j stand for logging message for java . It is an open source framework to log message of your program final destination like Console , File , Network ,Database.

Q. How to config Log4j ?

A: Add dependency = Log4j(1.2.17)

Make Log4j properties file

Make Object of Logger in all Ctl & Model

Logger log = Logger.getLogger(UserCtl.class)

1. *Component ofLog4j :-

Logger : Responsible for capturing logging information .

Layout : Responsible for formatting logging information in different style .

Appender : Responsible for publishing logging information to various preferred destination .

2. Environment Of LOG4J :-

Development(Debug level)

Quality Assurance(info level , Error level)

Production Environment(warn level)

3. LOG4J Appender :-

ConsoleAppender

FileAppender

RollingfileAppender

DailyRollingfileAppender

EternalRollingfileAppender

JDBCAppender

SMTPAppender

SocketAppender

TelnetAppender

NullAppender

4. Layout of Log4j :-

DateLayout

HTMLLayout

XMLELayout

SimpleLayout

PatternLayout

(Log4j.appender.stdout.layout = org.apache.log4j.PatternLayout)

5. *Level of Log4j :-

DebugLevel

InfoLevel

WarnLevel

ErrorLevel

FatalLevel

Q. How to enable Log4j ?

Ans : Log4j.rootLogger = Debug , file , Stdout (Console msg)

Debug : Use to print message .

File : (particular file me write krta hai).

Stdout : To print message on console .

Q. How to print message on console ?

Ans : Log4j.appenders.stdout.Target = System.out

Q. How many Appender are use in Log4j ?

Ans : 2 Appender are used === RollingfileAppender , ConsoleAppender

Log4j.appenders.file = org.apache.log4j.RollingfileAppender

Log4j.appenders.stdout = org.apache.log4j.ConsoleAppender

Q Which Layout is followed by Log4j?

Ans : Log4j.appenders.file.Layout = org.apache.log4j.patternLayout

Ans : Log4j.appenders.stdout.Layout = org.apache.log4j.patternLayout

Q. How to off Debug message ?

Ans: Replace Debug by Off

Q. What is Java doc?

A. Java doc is documentation which help other developer to understand the project .

Q. How to apply Javadoc ?

Ans : 1. Alt + Shift + j

2. Project menu Generate Javadoc

3. Config me javadoc.exe path

4. Select Destination

5. Visibility Private (we can access private attribute of class)

Q. How to mapping of JavaDoc ?

Ans: /doc/index.html

Q. Why we did mapping ?

A. It provide LooseCoupling (By which we access any ctrl or view easily)

Q. What is filter?

Ans : Filter perform Pre-Processing and Post-Processing operation on request and response of Client.

Q. What is Session ?

Ans : Session is conversation b/w Browser web client & web server. Session is implicit object .

Q. How to destroy session?

- Ans :
1. session.invalidate();(Minutes)
 2. <session config>(Minutes)
 <session timeout>20</session timeout>
 </session config>
 3. session.setMaxInactiveInterval();(Second)

Q. How to Tracked session ?

- Ans :
1. Cookies (set of key value pair of string)
 2. HiddenForm filled
 <input type="hidden" name="jsessionId" value=<%=session.getId() %>
 3. URL Rewriting (response.encodeURL(url))

Q. What is Servlet ?

A. Servlet share special type of java classes that produce dynamic web contents to develop a web application.

Q. What is JSP ?

Ans : JSP Stand for Java Server Pages . This is the special form of servlet to develope web page .

Q. What is Web application ?

Ans: The application that is access by a web browser (chrome ,FireFox ,IE) using HTTP protocol is called a web application.

Q. What is HTTP protocol ?

Ans: Hyper text Transfer Protocol , which communicate b/w web client & server .
It is a way by which request is submitted .

Q. What is Container ?

Ans : Container is responsible to COmpile , Execute & manage Lifecycle of server resource .

Q. What is Transition Handling?

Ans: Transition is an atomic unit to be committed or rollback in a single attempt .

OR

A transaction is a set of SQL statement or a unit of work that will be committed and rolled back together. Transactions are handled with the help of connection.setAutoCommit(false), connection.commit(), and connection.rollback() method.

1. conn.setAutocommit(false);

// True kyu nhi dete === kyuki vo null value ko save kr leta hai
database me automatically data ko save nhi krna na chahte hai pahle vo jake check kare
OR ===== kyu ki jb bhi koi exception aaye to vo data base me ek bhi record save na kre

***2. conn.commit();**

forcefully data ko save krta hai .

***3. conn.rollback();**

unwanted data ko rollback me aajayega .

DML - it is used to perform CRUD operation.(Add ,Update,delete)

DDL :- it is used to perform create table, alter table, drop table.

DCL :- it is used to rollback and commit.

JAVA = Just Another Virtual Asclator

JSP = Java Server Page

API = Application Programming Interface

DDL = Data Definition Language(Perform create table , alter table ,drop table)

DML = Data Manipulation language(Perform Crud operation)

DCL = Data Control Language(used to rollback and commit)

Q. Why and when PreparedStatement is used?

A. When the same query is fired multiple times with different parameters.

(The PreparedStatement interface extends the Statement interface)

Q. What are the differences between Statement and PreparedStatement?

A. Prepared statement preserves parsed queries and reuses the same query with different parameters whereas Statement queries are parsed in every database call.

Q. Difference between Comparable and Comparator?

Ans: Comparable : It has default nature of sorting order & it has compareTo() .

Comparator : It is customized & it has compare() .

Q. What is Debugging?

Ans: Debugging : It is a process by which developer check the flow programm.

Q. Difference b/w DoGet & DoPost ?

DoGet :-

1. Perform view logic (retrieve data).
2. Data is sent as query string appended to the URL.
3. Parameters are saved in browser history because they are part of the URL.
4. Can send only text and numbers (properly encoded).
5. Can be cached.
6. Can be bookmarked.
7. Suitable for idempotent operations (doesn't modify server state).

DoPost:-

1. Perform submit logic (send data to the server).
2. Data is sent as message body of the HTTP request (not part of URL).
3. Parameters are not saved in browser history.
4. Can send binary data (e.g., files).
5. Cannot be cached.
6. Cannot be bookmarked.
7. Suitable for non-idempotent operations (modifies server state).

Q. Difference between Unchecked Exception and Checked Exception?

Ans:

Unchecked Exception : are optional to handle RuntimeException and Error classes are unchecked exception.

Checked Exception : are Mandatory to handle Keyword throw is mandatory to propagate checked exception.

Q. Implicit objects in JSP?

- A.
1. request(HttpServletRequest)
 2. response(HttpServletResponse)
 3. session(HttpSession)
 4. application(ServletContext)
 5. page(current jsp)
 6. pageContext(PageContext)
 7. out(JspWriter)
 8. exception(Throwable)
 9. config(ServletConfig)

Abstract Factory :- it is pattern work around a super factory which create other factories.

Use Case:- one complete functionality is called UseCase.

Modules :- set of related use case is called Modules.

Q. what is utility classes?

A. Utility classes are used to reusability and reuseable services.

Q. what are the utility classes you have in your project?

Nine utility classes we have used in our project.

1. DataUtility.java :- Data Utility class is used to format data from one format to another.
 2. DataValidator.java :- DataValidator class is used to validate input data.
 3. HtmlUtility.java :- HtmlUtility class is used to produce html content like Drop DownList etc.
 4. ServletUtility.java: This class provides utility operation for Serviet container like forward, redirect, handle generic exception, manage success and error message, manage default Bean and List, manage pagination parameters.
 5. EmailMessage.java: - EmailMessage contains the content of email message.
 6. EmailUtility.java :- Email Utility provides Email Services 1.sendMail();
 7. EmailBuilder.java:- EmailBuilder Class that build Application Email messages.
- Three method :
- 1.getUserRegistrationMessage();
 - 2.getForgotPasswordMessage();
 - 3.getChangePasswordMessage();
8. JDBCDataSource.java:- it is a DATA CONNECTION POOL.

Q. Defference between string buffer and string builder.?

Ans) StringBuffer is synchronized and thread safe and used in multi user system.

StringBuilder is Asychronized and not thread safe and used in single user system.

Q. What is basectl. And all methods.?

A. BaseCtl is parent ctl of all ctl and it contain generic work flow, generic operation,generic constants.

Generic Work flow provides by services method.

Generic Operation provides by basectl method. such as preload, validate,populateBean,getView).

Generic Constants provides by all buttons.

Q. Defference between Static polymorphism and Dynamic polymorphism.?

A. Static polymorphism is done by overloadding and Dynamic polymorphism is done method overriding.

Q. Difference between jboss and tomcat?

- 1.JBoss is a Application server whereas Tomcat is a WebServer.
2. JBoss follows distributed transactions like as EJB where as tomcat follows jsp and servlet.
- 3.JBoss is havely weight as compare to tomcat.
- 4.JBoss works as slow as compare to tomcat.

Q. What is serviet lifecycle.?

A. init();
service();
destroy();

Q. what is the jsp page life cycle?

A. 1. jsplInit (JspPage interface)
2. _service) (HttpJspPage)
3. jspDestroy)(JspPage interface)

Q. Difference between web server and application server?

A. Web server contains only web or servlet container. It can be used for servlet, jsp, struts, jsf etc. It can't be used for EJB.

Application server contains Web and EJB containers. It can be used for servlet, jsp, struts, jsf, ejb etc. It is a component based product.

Q. Difference between checked exception and unchecked exception?

A. Uncheked exceptions are optional to handle. Runtime Exception and Error classes and their subclasses are uncheked exceptions.

Checked exceptions are mandatory to handle. The *throws* keyword is mandatory to propagate checked exceptions and optional for unchecked exceptions.

Q. Differences Between Interface and Abstract Classes in Java In Java, interfaces and abstract classes are used to achieve data abstraction.

Key Differences:

Abstract Classes:

1. Can have both abstract methods (methods without a body) and concrete methods (methods with a body).
2. Can have instance and static variables
3. Can have constructors.
4. A class can extend only one abstract class (single inheritance).

Interfaces:

1. It has abstract methods (prior to Java 8).
2. From Java 8 onwards. it can have default methods (methods with a body)
3. It can have static methods.
4. It has attributes, by default, each attribute is static and final in the interface.
5. It does not have constructors.
6. A class can implement multiple interfaces (multiple inheritance).

Q. How can you achieve data abstraction ?

Ans.

1. Data Abstraction is process of hiding the implementation details and showing only the functionality.
2. Data Abstraction in java is achieved by using interface and abstract class.

Q. What is Abstract Class ?

A. Abstract class is used when parent need to provide default behavior along with abstract method.
It enforces child classes to implement specialized behavior of abstract methods. It can never be instantiated.

Q. What is Maven?

A. Maven is a powerful build automation tools.

Q. What is the build/war?

A. It is whole compile code of entire application.

Q. Types of validation in the project?

There are two types of validations in our Projects.

1. ServerSideValidations

1.1. InputValidations

Progrative Validations.
Declarative Validations.

1.2. Business Validations(Check from Database)

2. ClientSide Validations Use JavaScript

Q. What is input validation?

A. Input validations checks inputs data entered by user.

Q. How you are performing bussiness validation?

A. Firstly we have propagated exception on model and we have catched and Set on ctl and get on view.

Q. What is bussiness validation?

A. Checks the business conditions that may need database communication.