

### Practical No.: 5A

**Aim: Write a Program of RMI for Retrieving Students information from College Database**

*File → New Project → Java Application → Project Name: Database RMI → Finish*

*Right Click on Package Name (database.rmi) → New → Java Class → Class Name: Students*

Students.java

```
package database.rmi;

/**
 *
 * @author TechnoBoy
 */
public class Students implements java.io.Serializable {

    private int id;
    private String name,address;

    public int getId(){
        return id;
    }
    public String getName(){
        return name;
    }

    public String getAddress(){
        return address;
    }

    public void setId(int id){
        this.id=id;
    }
    public void setName(String name){
        this.name=name;
    }

    public void setAddress(String address){
        this.address=address;
    }
}
```

*Right Click on Package Name (database.rmi) → New → Java Interface → Class Name: showData*

showData.java

```
package database.rmi;

import java.rmi.Remote;
import java.rmi.RemoteException;
```

```
import java.util.List;

/**
 *
 * @author TechnoBoy
 */
public interface showData extends Remote{
    public List<Students> getStudents() throws RemoteException;
}
```

*Right Click on Package Name (database.rmi) → New → Java Interface → Class Name:*  
**showDataImpl**

**showDataImpl.java**

```
package database.rmi;

import com.mysql.jdbc.Connection;
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;
import java.sql.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author TechnoBoy
 */
public class showDataImpl extends UnicastRemoteObject implements showData{

    public showDataImpl() throws RemoteException {
    }

    List<Students> list;
    Connection con;
    Statement st;
    @Override
    public List<Students> getStudents() throws RemoteException {

        try {
            list = new ArrayList<>();
            Class.forName("com.mysql.jdbc.Driver");
            java.sql.Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/students", "root","");
            Statement st=con.createStatement();
            String query="select * from details";
            try (ResultSet rs = st.executeQuery(query)) {
                while(rs.next())
                {
                    int id=rs.getInt("Id");
                    String name=rs.getString("Name");
                }
            }
        }
    }
}
```

```

        String address=rs.getString("Address");
        System.out.println("done");
        Students students = new Students();
        students.setId(id);
        students.setName(name);
        students.setAddress(address);
        list.add(students);
    }
}
} catch (ClassNotFoundException | SQLException ex) {
    Logger.getLogger(showDataImpl.class.getName()).log(Level.SEVERE, null, ex);
}
return list;
}
}

```

*Right Click on Package Name (database.rmi) → New → Java Interface → Class Name: **Server***

#### **Server.java**

```

package database.rmi;

import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

/**
 *
 * @author TechnoBoy
 */
public class Server {
    public static void main(String[] args) throws RemoteException {
        Registry reg = LocateRegistry.createRegistry(5000);
        reg.rebind("show",new showDataImpl());
        System.out.println("Server is Ready !!!");
    }
}

```

*Right Click on Package Name (database.rmi) → New → Java Interface → Class Name: **Client***

#### **Client.java**

```

package database.rmi;

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

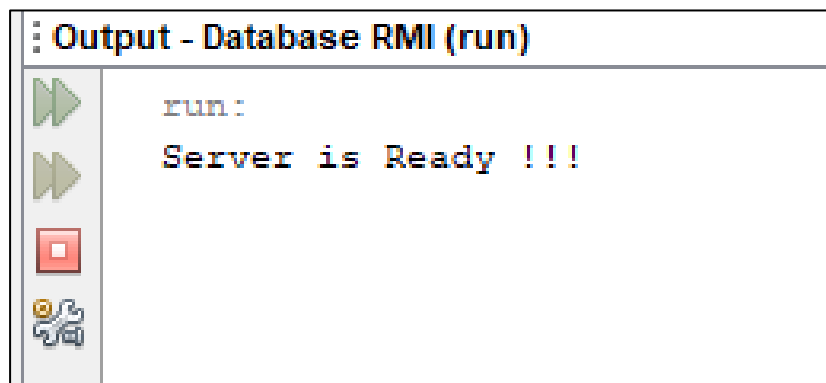
/**
 *
 * @author TechnoBoy
 */
public class Client {
    public static void main(String[] args) {
        try {
            showData sd=(showData) Naming.lookup("rmi://localhost:5000/show");
            System.out.println(sd.getStudents());
            List<Students> list=(List) sd.getStudents();
            list.stream().forEach(s) -> {
                System.out.println(s.getId()+" "+s.getName()+" "+s.getAddress());
            };

        } catch (NotBoundException | MalformedURLException | RemoteException ex) {
            Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
}

```

#### **Output:**



Run Server:

Run Client:

