

Practical No.: 5B

Aim: Write a Program of RMI for Retrieving Books Available in Library

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

BooksRMI.java

```
package books.rmi;

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

    private int book_id;
    private String book_name,book_author;

    public int getBookId(){
        return book_id;
    }
    public String getBookName(){
        return book_name;
    }
    public String getBookAuthor(){
        return book_author;
    }

    public void setBookName(String book_name){
        this.book_name=book_name;
    }
    public void setBookAuthor(String book_author){
        this.book_author=book_author;
    }
}
```

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

showData.java

```
package books.rmi;

import java.rmi.Remote;
import java.rmi.RemoteException;
import java.util.List;

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

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

showDataImpl.java

```
package books.rmi;
```

```
import java.rmi.RemoteException;  
import java.rmi.server.UnicastRemoteObject;  
import java.sql.*;  
import java.util.ArrayList;  
import java.util.List;
```

```
/**  
 *  
 * @author TechnoBoy  
 */
```

```
public class showDataImpl extends UnicastRemoteObject implements showData{
```

```
    List<BooksRMI> list;  
    Connection cn;  
    Statement st;
```

```
    public showDataImpl() throws RemoteException {  
    }
```

```
@Override
```

```
public List<BooksRMI> show() throws RemoteException {  
    try  
    {  
        list=new ArrayList<>();  
        Class.forName("com.mysql.jdbc.Driver");  
        cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/book","root","");  
        st=cn.createStatement();  
        String Query="select * from Details";  
        ResultSet rs=st.executeQuery(Query);  
        while(rs.next())  
        {  
            int id=rs.getInt("B_Id");  
            String name=rs.getString("B_Name");  
            String author=rs.getString("B_Author");  
  
            BooksRMI br=new BooksRMI();  
            br.setBookId(id);  
            br.setBookName(name);  
            br.setBookAuthor(author);  
            list.add(br);  
        }  
    }  
    catch(Exception e)  
    {  
        System.out.println(e);  
    }  
}
```

```
        return list;
    }
}
```

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

Server.java

```
package books.rmi;

import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author TechnoBoy
 */
public class Server {
    public static void main(String[] args) {
        try
        {
            Registry reg = LocateRegistry.createRegistry(5000);
            reg.rebind("books", new showDataImpl());
            System.out.println("Server Ready!!!!");
        }
        catch (RemoteException ex) {
            Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

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

Client.java

```
package books.rmi;

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

/**
 *
 * @author TechnoBoy
```

```

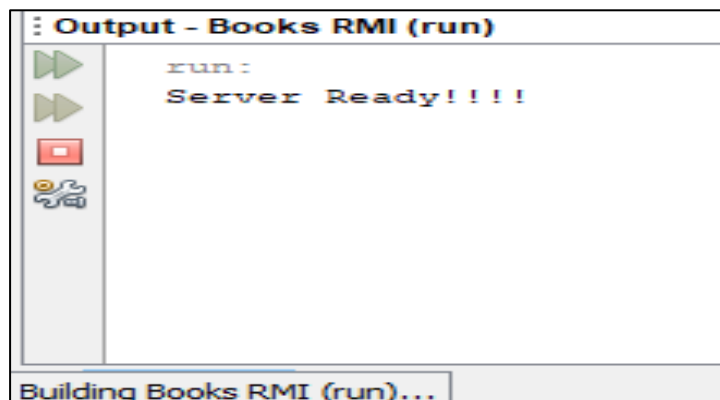
*/
public class Client {
    public static void main(String[] args) {
        try {
            showData br=(showData)Naming.lookup("rmi://localhost:5000/books");
            List<BooksRMI> list=(List) br.show();
            for (BooksRMI b : list) {
                System.out.println(b.getBookId()+" -> "+b.getBookName()+" -> "+b.getBookAuthor());
            }

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

```

Output:

Run Server:



Run Client:

