

### Practical No.: 3B

**Aim: Write a Program using RMI to Retrieve Time and Date function from Server to Client**

*File → New Project → Java Application → Project Name: RMIDate&Time → Finish*

*Right Click on Package Name (rmidate.time) → New → Java Interface → Class Name : Date*

#### Date.java

```
package rmidate.time;

import java.rmi.Remote;
import java.rmi.RemoteException;
/**
 *
 * @author TechnoBoy
 */
public interface Date extends Remote{
    public String getDate() throws RemoteException;
}
```

*Right Click on Package Name (rmidate.time) → New → Java Interface → Class Name : Time*

#### Time.java

##### Source Code:

```
package rmidate.time;
import java.rmi.Remote;
import java.rmi.RemoteException;

/**
 *
 * @author TechnoBoy
 */
public interface Time extends Remote{
    public String getTime() throws RemoteException;
}
```

*Right Click on Package Name (rmidate.time) → New → Java Class → Class Name : TimeImpl*

#### TimeImpl.java

```
package rmidate.time;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

/**
 *
 * @author TechnoBoy
```

```

*/
public class TimeImpl extends UnicastRemoteObject implements Time{

    public TimeImpl() throws RemoteException {
    }

    @Override
    public String getTime() throws RemoteException {
        return "Time :"+java.time.LocalDateTime.now();
    }

}

```

*Right Click on Package Name (rmdate.time) → New → Java Class → Class Name : **DateImpl***

#### **DateImpl.java**

```

package rmdate.time;

import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

/**
 *
 * @author TechnoBoy
 */
public class DateImpl extends UnicastRemoteObject implements Date{

    public DateImpl() throws RemoteException {
    }

    @Override
    public String getDate() throws RemoteException {
        return "Date : "+java.time.LocalDate.now();
    }

}

```

*Right Click on Package Name (rmdate.time) → New → Java Class → Class Name : **Server***

#### **Server.java**

```

package rmdate.time;

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 datetime= LocateRegistry.createRegistry(5000);
            datetime.rebind("date", new DateImpl());
            datetime.rebind("time", new TimeImpl());
            System.out.println("Server Ready");
        } catch (RemoteException ex) {
            Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

Right Click on Package Name (rmdate.time) → New → Java Class → Class Name : **Client**

### **Client.java**

```

package rmdate.time;

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

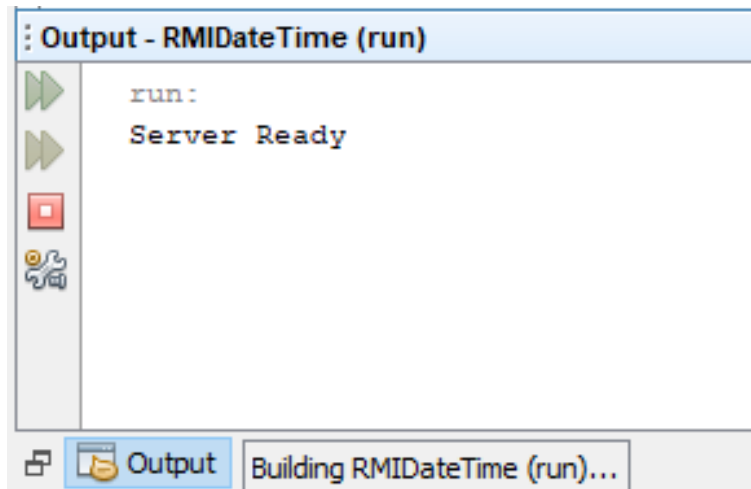
/**
 *
 * @author TechnoBoy
 */
public class Client {
    public static void main(String[] args) {
        try {
            Date d=(Date) Naming.lookup("rmi://localhost:5000/date");
            Time t=(Time) Naming.lookup("rmi://localhost:5000/time");

            System.out.println(d.getDate());
            System.out.println(t.getTime());
        } catch (NotBoundException | MalformedURLException | RemoteException ex) {
            Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

**Output:**

*Run Server.java*



*Run Client.java*

