SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore - 641035.

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++"
Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Computer Applications

Course Code: 23CAT606

Course Name: Java Programming

Unit IV: Advanced Java Programming
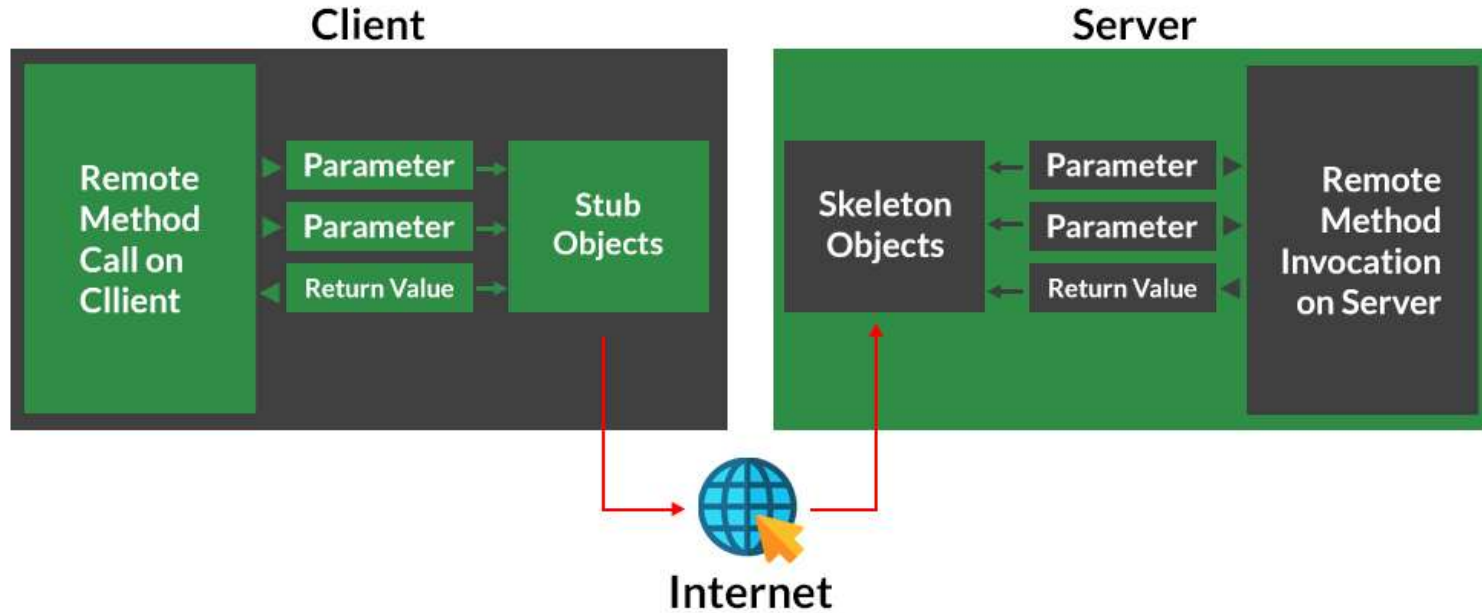
Topic :   RMI

- The RMI (Remote Method Invocation) is an API that provides a mechanism to create **distributed application** in java.

- The RMI allows an object to **invoke methods on an object running in another JVM**.

- The RMI provides remote communication between the applications using **two objects stub and skeleton.**

Understanding stub and skeleton

Stub: Working model

- **The stub is an object, acts as a gateway for the client side. All the outgoing requests are routed through it. It resides at the client side and represents the remote object. When the caller invokes method on the stub object, it does the following tasks:**

1.It initiates a connection with remote Virtual Machine (JVM),

2.It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM),

3.It waits for the result

4.It reads (unmarshals) the return value or exception, and

5.It finally, returns the value to the caller.

Skeleton: Working model

- **The skeleton is an object, acts as a gateway for the server side object. All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:**

1.It reads the parameter for the remote method

2.It invokes the method on the actual remote object, and

3.It writes and transmits (marshals) the result to the caller.

Java RMI Program

1. Create the remote interface

2. Provide the implementation of the remote interface

3. Compile the implementation class and create the stub and skeleton objects using the rmic tool

4. Start the registry service by rmiregistry tool

5. Create and start the remote application

6. Create and start the client application

Create Remote Interface

```
import java.rmi.*;
public interface Adder extends Remote{
public int add(int x,int y)throws RemoteException;
}
```

```
import java.rmi.*;

import java.rmi.server.*;

public class AdderRemote extends UnicastRemoteObject implements Adder{

AdderRemote()throws RemoteException{

super();

}

public int add(int x,int y){return x+y;}

}
```

Create the stub and skeleton objects using the rmic tool.

rmic AdderRemote

The rmic tool invokes the RMI compiler
and creates stub and skeleton objects.

rmiregistry 5000

Now start the registry service by using the rmiregistry tool. If you don't specify the port number, it uses a default port number.

Create and run the server application

```java
import java.rmi.*;
import java.rmi.registry.*;
public class MyServer{
public static void main(String args[]){
try{
Adder stub=new AdderRemote();
Naming.rebind("rmi://localhost:5000/sonoo",stub);
}catch(Exception e)
{System.out.println(e);}
}
}
```

Create and run the client application

```
import java.rmi.*;

public class MyClient{

public static void main(String args[]){

try{

Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/sonoo");

System.out.println(stub.add(34,4));

}catch(Exception e){}

}

}
```

For running **this** rmi example,

1) compile all the java files   javac *.java

2)create stub and skeleton object by rmic tool   rmic AdderRemote

3)start rmi registry in one command prompt   rmiregistry 5000

4)start the server in another command prompt   java MyServer

5)start the client application in another command prompt   java MyClient

Reference

1. Herbert Schildt " The Complete Reference Java 2, 8th edition , Tata McGraw Hill, 2011

2. Ralph Bravaco, Shai Simonson, "Java Programming: From the Ground up Tata McGraw Hill, 2012

3. https://www.javatpoint.com

Thank you