ECAP615

Programming in Java

Harjinder Kaur

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- learn the basic concepts Socket and socket classes
- understand the various methods of socket class
- know the various applications of the socket class

Socket

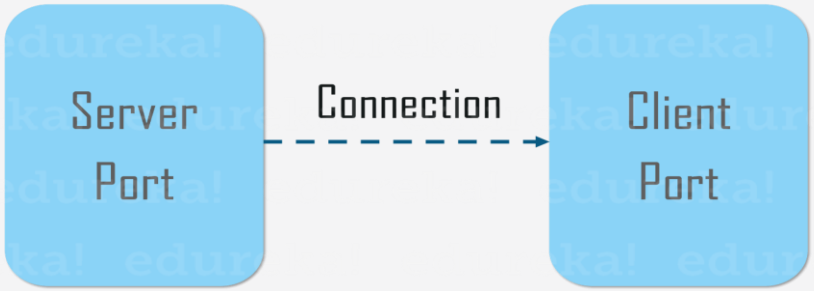
- A socket is one end-point of a two-way communication link between two programs running on the network.
- Socket classes are used to represent the connection between a client program and a server program.

Socket

- The java.net package provides two classes:
  - Socket
  - ServerSocket

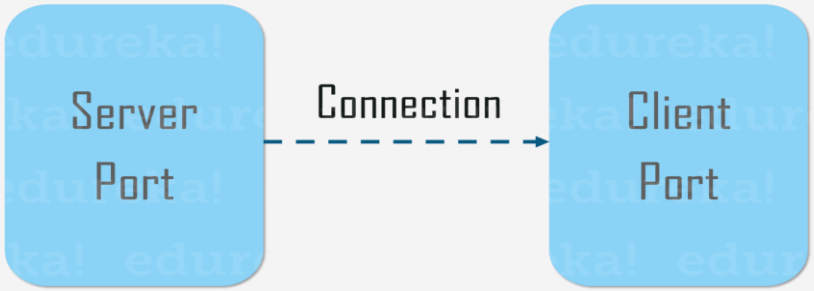
Socket

- A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.
- An endpoint is a combination of an IP address and a port number.



Socket

- The package in the Java platform provides a class, Socket that implements one side of a two-way connection between your Java program and another program on the network.
- The class sits on top of a platform-dependent implementation, hiding the details of any particular system from your Java program.



Socket Class

- The java.net.Socket class allows us to create socket objects that help us in implementing all fundamental socket operations.
- We can perform various networking operations
- Each Socket object that has been created using with java.net.Socket class has been associated exactly with 1 remote host.
- For connecting to another different host, we must create a new socket object.

Socket Class

The syntax for importing Socket class from java.net package :

import java.net.Socket;

The following are the methods used in socket class:

- public InputStream getInputStream()
- public OutputStream getOutputStream()
- public synchronized void close()

Client Side Programming

- The client will first wait for the server to start.
- Once the server is up and running, it will send the requests to the server.
- After that, the client will wait for the response from the server.

Initiate a Client request

In order to initiate a client's request, you need to follow the below-mentioned steps:

- Establish a Connection

You can create a Socket with the help of a below statement:

Socket socket = new Socket("127.0.0.1", 5000)

- Communication
- Closing the connection

Methods used in Socket class

- bind(SocketAddress bindpoint)
- Close()
- connect(SocketAddress endpoint)
- getInetAddress()
- getInputStream()

Methods used in Socket class

- getKeepAlive()
- getLocalAddress()
- getLocalPort()
- getLocalSocketAddress()

Methods used in Socket class

- getOutputStream()
- getPort()
- isBound()
- isClosed()
- isConnected()

Applications of Socket Class

- Socket class is implemented in creating a stream socket and which is connected to a specified port number and port address.

public Socket(InetAddress address, int port)

Applications of Socket Class

- Socket class is used for the creation of socket and connecting to the specified remote address on the specified remote port in javax.net

SocketFactory.createSocket(InetAddress address, int port, InetAddress localAddress, int localPort)

Applications of Socket Class

- In javax.ssl.net, the Socket class is used to return a socket layered over an existing socket connected to the named host at a given port.

SSLSocketFactory.createSocket(Socket s, String host, int port, boolean autoClose)

Applications of Socket Class

- In javax.rmi.ssl, Socket class is used for creating an SSL socket.

SslRMIClientSocketFactory.createSocket(String host, int port)

Applications of Socket Class

- In java.nio.channels, Socket class is used for retrieving a socket associated with its channel

SocketChannel.socket()


That’s all for now…