sockets in java

1 what is a socket?

2 Most important method for socket programming

3 Steps for building a client socket and server socket

Made by :
HARIB Douaa /ZAHEDI Chaymaa
TANTOUNI Mohamed / TROUKI Reda

1 | Sockets →

- **Sockets are endpoints in network communications.**
- **A socket server is usually a multi-threaded server that can accept socket connection requests.**
- **A socket client is a program/process that initiates a socket communication request.**
- **Sockets are useful for both stand-alone and network applications.**
- **Sockets allow you to exchange information between processes on the same machine or across a network, Distribute work to the most efficient machine they easily allow access to centralized data.**

2 | Most used methods for socket programming →

-Socket():Creates an unconnected socket, with the system-default type of SocketImpl.

-socket(InetAddress address, int port):Creates a stream socket and connects it to the specified port number at the specified IP address.

-ServerSocket:This class implements server sockets. A server socket waits for requests to come in over the network. It performs some operation based on that request, and then possibly returns a result to the requester.

-(socketobject).accept:
Listens for a connection to be made to this socket and accepts it.

-(socket object).close: Closes this socket.

-getInetAddress():
Returns the local address of this server socket.

-PrintWriter(OutputStream out): Creates a new PrintWriter, without automatic line flushing, from an existing OutputStream.

-getOutputStream():returns an output stream for writing bytes to the socket.

-InputStream:are used to read bytes from a stream . It grabs the data byte by byte without performing any kind of translation.

-InputStreamReader:is a bridge from byte streams to character streams: It reads bytes and decodes them into characters using a specified charset .

-BufferedReader:Reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.

3 | Steps for building a client socket and server socket →

The following steps are applied for typical communication with the server:

1-the client initiates a connection to a server specified by hostname/ip address and port number:

```
        Socket s=new Socket("localhost",4999);
```

2-Establish a Socket Connection on the server side:

the ServerSocket waits for the client requests and accepts the requests (when a client makes a new Socket())

```
        ServerSocket ss=new ServerSocket(4999);
        Socket s=ss.accept();
```

3-Send data to the server using an outputStream:

```
         PrintWriter pr=new PrintWriter(s.getOutputStream());
         pr.println("hello from client");
         pr.flush();
```

4-read data from the server using intputStream.

```
         InputStreamReader in =new InputStreamReader
(s.getInputStream());
        BufferedReader bf=new BufferedReader(in);
```

5-close the connection.

```
        s.close();
```

steps 4 and 5 can be repeated many times depending on the nature of the communication in both sides.