



---

# DOCUMENTATION

---

Chat Application



Prepared By :

20182619 SOHILA MAMOUN

20183718 SALMA MOHAMMED

2018 2893 RADWA HAMDY

20182164 MARYAM EMAD

Supervised by :

HP

DR. Mohammed Hussien

TA: Shaimaa Tarek

- Project: Chat Application
- Chat Application : The purpose of the **chat application** is to allow users be able to the **chat** with each other, like a normal **chat application**.
- The project is done by all the group
- Our project can be done by both protocol TCP and UDP.
- We use in our project TCP Protocol
- TCP is a **more commonly used protocol than UDP**.
- TCP is connection-oriented and UDP is connectionless. This means **that before sending TCP packets, a connection is established between the server and the client**. This process of setting up a connection is called TCP handshaking. The stream of packets is then sent over this connection.
- **In UDP, there is no such connection**. Each packet is sent individually and directly from the sender to the receiver without a reliable data channel.
- **TCP has error detection and correction methods**.
- **TCP is used in applications where reliability is more important**, such as file transfer, emails, and web browsing.  
**UDP is used in applications where speed is more important** such as video conferencing, live streaming, and online gaming.

Code:Server

```
/*  
 * To change this license header, choose License Headers in Project  
Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package chatapplication;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.PrintStream;  
import java.net.ServerSocket;  
import java.net.Socket;  
import java.util.Scanner;  
  
/**  
 *  
 * @author HP  
 */  
public class ChatApplication {
```

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) throws IOException {
    // this is my server
    //Define the socket
    ServerSocket ss =new ServerSocket(2000);
    //accept connection from client
    Socket sk = ss.accept();
    BufferedReader cin = new BufferedReader (new
InputStreamReader(sk.getInputStream()));
    PrintStream cout = new PrintStream (sk.getOutputStream());
    BufferedReader stin = new BufferedReader(new
InputStreamReader(System.in));
    // which data will be store
    String s;
    Scanner sc = new Scanner(System.in);
    while(true){
        //read from the socket
        s = cin.readLine();
        //print data from the client
        System.out.println("Client "+s+"\n");
    }
}

```

```

        System.out.println("Server : ");
        s=sc.nextLine();
        if(s.equalsIgnoreCase("BYE")){
            System.out.println("Connection is ended by server");
            break; //exit
        }
        cout.println(s);
    }
    ss.close();
    sk.close();
    cin.close();
    cout.close();
    stin.close();
}
}

```

Code:Client

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```
package chatapplication;
```

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
import java.io.PrintStream;
```

```
import java.net.Socket;
```

```
/**
```

```
 *
```

```
 * @author HP
```

```
 */
```

```
public class Client {
```

```
    public static void main(String[] args) throws IOException {
```

```
        //create socket with IP address and port number for transfer  
        the socket
```

```
        Socket sk = new Socket("127.0.0.1",2000);
```

```
        // input stream on socket
```

```
        BufferedReader sin = new BufferedReader (new  
        InputStreamReader(sk.getInputStream()));
```

```
        PrintStream sout = new PrintStream (sk.getOutputStream());
```

```
        //this buffer to get input from the user
```

```
        BufferedReader stin = new BufferedReader(new  
        InputStreamReader(System.in));
```

```

// which data will be store
String s;
while(true){
    System.out.print("Client :");
    s = stin.readLine();
    sout.println(s);
    // if client say bye
    if(s.equalsIgnoreCase("BYE")){
        System.out.println("Connection is ended by client");
        break; //exit
    }

    //reading the data
    s=sin.readLine();
    System.out.println("Server " +s+"\n");

}
sk.close();
sin.close();
sout.close();
stin.close();
}

```



```
}  
: Output  
ChatApplication (run) × ChatApplication (run) #2 ×  
run:  
Client hii  
Server :  
hii  
Client how are you?  
Server :  
fine  
Client bye  
Server :  
bye  
Connection is ended by server  
BUILD SUCCESSFUL (total time: 40 seconds)  
|
```

```
: Output  
ChatApplication (run) × ChatApplication (run) #2 ×  
run:  
Client :hii  
Server hii  
Client :how are you?  
Server fine  
Client :bye  
Connection is ended by client  
BUILD SUCCESSFUL (total time: 32 seconds)  
|
```