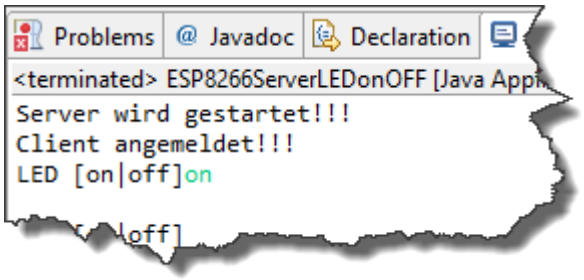
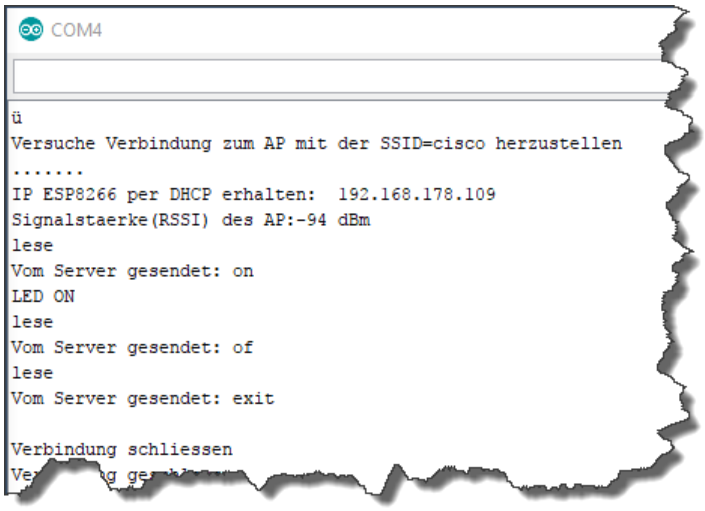
Interne LED des NODEMCU über JAVA-Server (Konsole) ein- bzw. ausschalten.




```java
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class ESP8266ServerLEDonOFF {
	public static void main(String[] args) throws IOException {
		System.out.println("Server wird gestartet!!!");

		ServerSocket einSS = new ServerSocket(50000);
		// Ein ArbeitsSocket erzeugen, wenn sich ein Client am Server meldet
		Socket einAS = einSS.accept();

		BufferedReader vonKonsole=new BufferedReader(
							new InputStreamReader(System.in));

		BufferedWriter zumClient = new BufferedWriter(new OutputStreamWriter(
								einAS.getOutputStream()));
		System.out.println("Client angemeldet!!!");
		String s, client;
		try {
			do{
				System.out.print("LED [on|off]");
				s=vonKonsole.readLine();
				System.out.println();
				zumClient.write(s);
				zumClient.flush();
			}while(!s.equalsIgnoreCase("exit"));
		} catch (Exception e) { }
		System.out.println("Client abgemeldet!!!");
		einAS.close();
		einSS.close();
	}
}
```

```
#include <ESP8266WiFi.h>
//Globale Konstanten
const char* ssid = "deineSSID";
const char* password = "deinPW";
const char* host = "192.168.178.106"; //IP des Java-Servers
const int serverPort = 50000; //Port des Java-Servers (ServerSocket)
const int builtinLED = D4;
//Globale Variablen
WiFiClient client;
String befehl;

void setup() {
  Serial.begin(115200); //Kontrollausgabe aktivieren
  delay(800);
  Serial.print("Versuche Verbindung zum AP mit der SSID=");
  Serial.print(ssid);
  Serial.println(" herzustellen");

  WiFi.begin(ssid, password);
```

```
  /*Solange keine Verbindung zu einem AccessPoint (AP) aufgebaut wurde*/
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println();
  Serial.print("IP ESP8266 per DHCP erhalten:  ");
  Serial.println(WiFi.localIP());
  long rssi = WiFi.RSSI();/*Signalstärke des AP*/
  Serial.print("Signalstaerke(RSSI) des AP:");
  Serial.print(rssi);
  Serial.println(" dBm");

  pinMode(builtinLED, OUTPUT);
  digitalWrite(builtinLED, HIGH);//Interne LED aus

  client.setTimeout(500);
  if (!client.connect(host, serverPort)) {
    Serial.print("X");
    return;
  }
}

void loop() {
  if (client.available()) {
    Serial.println("lese");
    befehl = client.readStringUntil('\n');
    client.flush();//wichtig!!!
    delay(100);
    if (befehl.equals("")) {
      return;
    }
    //Kontrollausgabe
    Serial.print("Vom Server gesendet: ");
    Serial.println(befehl);
    // Gelesenen Befehl vom JavaServer auswerten
    //Inverse Logik bei interner LED beachten
    //LOW an D4 bedeutet interne LE leuchtet
    if (befehl.equalsIgnoreCase("on")) {
      digitalWrite(builtinLED, LOW );
      Serial.println("LED ON");
    } else if (befehl.equalsIgnoreCase("off")) {
      digitalWrite(builtinLED, HIGH);
      Serial.println("LED OFF");
    } else if (befehl.equalsIgnoreCase("exit")) {
      Serial.println();
      Serial.println("Verbindung schliessen");
      client.flush();
      client.stop();
      Serial.println("Verbindung geschlossen");
    }

  }
  delay(100);
}
```