```
/* web server with servo
*/
#include <WiFiNINA.h>   //IoT
char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int LED = LED_BUILTIN; // IoT
int status = WL_IDLE_STATUS;
WiFiServer server(80);

#include <Servo.h>
Servo skeletonHead;

int pos = 0;    // variable to store the servo position
int wait = 100;
int range = 20;

void setup() {
  skeletonHead.attach(4);  // use DIO4
  skeletonHead.write(90);
  Serial.begin(9600);
  pinMode(LED, OUTPUT);

  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to Network named: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    // wait 10 seconds for connection:
    delay(10000);
  }
  server.begin();
  printWifiStatus();
}

void loop() {
  CheckPage();
  if (digitalRead(LED)){
    skeletonHead.write(90+range);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(wait);
    skeletonHead.write(90-range);
    digitalWrite(LED_BUILTIN, LOW);
    delay(wait);
  }
  else skeletonHead.write(90);
}

 void CheckPage() {
  WiFiClient client = server.available();   // listen for incoming clients
  if (client) {
    Serial.println("new client");
    String currentLine = "";
```

```
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n') {
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Content-Length: 121");
            client.println();
            client.println("<HTML>");
            client.println("<BODY>");
            client.print("Click <a href=\"/H\">here</a> turn the LED on<br>");
            client.print("Click <a href=\"/L\">here</a> turn the LED off<br>");
            client.println();
            client.println("</BODY>");
            client.println("</HTML>");
            break;
          }
          else currentLine = "";
        }
        else if (c != '\r') currentLine += c;      // add it to the end of the
         currentLine
        if (currentLine.endsWith("GET /H")) digitalWrite(LED, HIGH);   // GET
         /H turns the LED on
        if (currentLine.endsWith("GET /L")) digitalWrite(LED, LOW);    // GET
         /L turns the LED off
      }
    }
    // close the connection:
    client.stop();
    Serial.println("client disconnected");
  }
}

void printWifiStatus() {
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");
    Serial.print("Go to http://");
    Serial.println(ip);
}
```