IoThook Nesnelerin ■nterneti

version 1.3

@mesebilisim, @iothook, @electrocoder, @sahinmersin

September 24, 2017

Contents

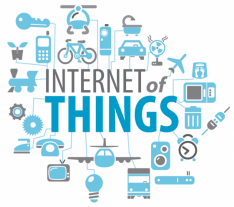
IOT HOOK

Quick reference

Iot Nedir

Iot Nedir?

Nesnelerin interneti “internet of things” 1999 y■l■nda Kevin Ashton taraf■ndan kullan■lan bir kavramd■r ve teknolojideki geli■meler ile birlikte bugünkü haline gelmi■tir. RFID teknolojisi için üretilen bu kavram günümüzde tüm elektronik cihazlara uygulanmaktad■r.



IoT - Nesnelerin ■nterneti

Iothook nedir?

Iothook internete ba■l■ nesneler (iot) aras■nda veri transferi yapan web servis a■■ projesidir. Iothook ile Arduino, Raspberry Pi, Android, iOS, Windows Phone, Web Site, Banana Pi, Orange Pi, Beaglebone, ARM, Pic, Windows, Mac OS X, ve Linux tabanl■ sistemleri birbirine ba■lar.

Niçin Iothook?

- Iothook h■zl■d■r,
- S■n■rs■z kanal olu■turabilirsin,
- S■n■rs■z element ekleyebilirsin,
- Tüm cihazlar■n ile kolayca veri gönderebilirsin (post),
- Tüm iot cihazlar■ndan kolayca veri alabilirsin (get),
- Datalar■n■ gerçek zamanl■ takip edebilirsin,
- Datalar■n için gerçek zamanl■ grafik olu■turabilirsin,

Iothook’ un sundu■u avantajlar:

- Kanal olu■turma,
- Kanal elementi ekleme,
- Web api,
- Web Sorgu,
- Form api,
- Twit atma,
- SMS atma,
- E-posta,
- Grafik,
- 7/24 destek,

Iothook tüm cihazlar■n■z aras■nda kesintisiz veri aktar■m■ yapan, internete ba■l■ nesnelerin kolayca ula■abilece■i ileti■im protokollerini destekler.

Google developer chart apileri ile entegre olarak verileri gerçek zamanl■ izleme olana■■ sa■lar.

Kanal Aç

Iothook kanal; internete ba■l■ nesneler aras■nda veri iletimini sa■lamak için olu■turulmu■ kanca sistemidir. Kanal ile iot sistemleri veri payla■■m■ yap■labilir, veri gönderim i■lemleri tan■mlan■r.

Iothook web servislerini kullanabilmek için üye olunmal■d■r. Üyelik seçenekleri ‘Free’, ‘Student’, ‘Pro’ ve ‘Ultra’ olmak üzere 4 kullan■m plan■ vard■r. Üye olmak için adrese gidiniz.

Üyelik ad■m■ndan sonra yönetim paneli arac■l■■■ ile ‘Kanal Ekle’ ekran■na girilir.

Kanal Ekle

Kanal ekleme ad■mlar■ ■u ■ekildedir:

- **Form Metod: Http (Hyper Text Transfer Protocol) de veriler TCP/IP metodu ile iletilmektedir. Http protokolü üzerinden veri iletimi request ve response istekleri ile gerçekle■ir. 'Request' gerçekle■mesi istenen talep, 'Response' ise yan■t olarak kullan■l■r. HTTP protokolüne göre POST, GET veya POST/GET metodu seçilir. Iothook ileti■iminde post ve get metodlar■ kullan■lmaktad■r.**
    - Post: Verilerin iot cihazda mesaj gövdesine yerle■tirilerek gönderilme i■lemidir.
    - Get: Verilerin iot cihaz ile sorgulanma ve cevap al■nma talebidir.
    - Post/Get: Veri aktar■m■n■n iot nesnesi ile server aras■nda çift tarafl■ olaca■■n■ gösterir.
- Form enctype: "application/x-www-form-urlencoded" ile iot cihaz■ndan gönderilen karakterlerin gönderilmeden önce kodlanaca■■n■ belirtir. "mutlipart/form-data" ise verilerin içerisinde ASCII olmayan verilerin oldu■unu dosya veya image format■nda veri oldu■unu belirtir.
- Ayg■t türü: Iot cihaz■n türünü belirler. Arduino, Raspberry Pi… gibi
- Kanal ad■: Verilerin toplanaca■■ kanal■n ad■.
- Web site: Veriler bir web sitesinde kullan■lacak ise web site adresi girilmelidir.
- Email ile haber ver: Veri al■nd■■■nda kay■tl■ olan mail adresine mesaj gonderir. Aktif edilirse 15dk. da bir veri gönderilmesi gerekir.
- Verileri kaydet: Iot nesnesinden gelen verilerin iothook veritaban■nda saklanmas■ için gereklidir.
- Resim: Kanal tan■t■m resmi olarak kullan■l■r.
- Aç■klama: Kanal bilgileri girilmelidir.
- Is public POST: Bu kanal genel kullan■ma aç■k ve veri eklenmesine aç■kt■r.
- Is public GET: Bu kanal genel kullan■ma aç■k ve verilerin okunmas■na izin verir.
- Yay■ndam■: Kanal■ aktif et.

Element Ekle

Iot cihaz■n■z için kanal olu■turduktan sonra kanalda bulunmas■n■ istedi■iniz veri alanlar■n■ olu■turmal■s■n■z. Bu alanlar veri almaya ba■lamak için eklenir. Element verilerine POST veta GET metodu ile ula■abilirsiniz.

Element ayarlar■:

- Kanal ad■: Elementin hangi kanala veri aktaraca■■ seçilir.
- Grafik türü: Toplanan verilerin çizilece■i grafik türünü belirler.
- Element tipi: Verilerin depolanaca■■ alan tipini belirler. Grafik çizimi sadece "number" veri tipinde yap■lmal■d■r.
- Kanal ad■: Verilerin toplanaca■■ kanal■n ad■.
- Element ad■: Verilerin tutulaca■■ element ad■.
- Yay■ndam■: Elementi aktif et.

Veri Gönder

Veri göndermek için öncelikle kanal ve element eklemeniz gerekir. Kanal olu■turuldu■unda size özel "api_key" üretilerek belirlenen eri■im metoduna göre (POST, GET, POST/GET) veri i■lemi gerçekle■tirilir.

Örne■in; Kanal■m■z ■s■, ■■■k, hareket, bar ve nem de■erlerini alan bir yap■da olsun. Kanal içerisinde bulunacak iot cihazlar■m■z bizlere bu datalar■ 15 sn. yede bir 100 kere göndersin.

Olu■turulan "API_KEY" Key Yöneticisi sayfas■ndan görülebilir.

Python ile JSON Veri Gönderme

Python ile Json Post Örne■i:

Bu örne■i http://bit.ly/2jI1FNQ sayfas■ndan indirebilirsiniz.

```
# -*- coding: utf-8 -*-

"""
  Python ile IoThook REST Api Testi

  Kod çal■■t■r■ld■■■nda APIKEY ile do■rulama gerçekle■tirilir.
  Kanal api_key ile ilgili kanal ve element de■erleri IoThook a post edilir.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  20 Eylul 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

"""

import requests
import json
import random
import pprint
import time


headers = {'Content-type': 'application/json'}

url = 'https://iothook.com/api/latest/datas/'

for i in range(100):
    data={
        'api_key': '519ac5d4-95a5-116e185a343eac447b', # demo hesap api_key
        'value_1': i*1,
        }

    data_json = json.dumps(data)

    response = requests.post(url, data=data_json, headers=headers)
```

```
    pprint.pprint(response.json())
    time.sleep(15)
```

Python GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Python ile Get metodu kullanarak veri gönderme Örne■i:

```
# -*- coding: utf-8 -*-

"""
  Python ile IoThook REST Api Testi

  Kod çal■■t■r■ld■■■nda APIKEY ile gonderim gerçekle■tirilir.
  Kanal api_key ile ilgili kanal ve element de■erleri IoThook a GET metodu ile gonderilir.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  11 Eylul 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

"""

import requests
import json
import random
import pprint
import time


headers = {'Content-type': 'application/json'}

API_KEY = '85c4ba5f-96ae-11841415634e983487e'

for i in range(10):
    url = 'https://iothook.com/api/latest/datas/update?api_key=' + API_KEY + '&value_1=10&va

    response = requests.get(url)
    data = response.json()
```

```
    print data
    time.sleep(15)
```

Arduino, ESP8266 POST Metodu ile Veri Gönderme

Bu örnekde Arduino Uno ya RX ve TX ile ba■lanm■■ olan ESP8266 ile iothook a veri gonderme örne■i verilmi■tir. Örnekde 0-100 aras■nda rastgele say■ üretilerek iothook da https://iothook.com/tr/channel/api/public/240 id numaral■ cihaz için gönderim gerçekle■mi■tir. Cihaz datalar■n■ https://iothook.com/tr/channel/api/public/240 linkinden gercek zamanl■ olarak takip edebilirsiniz.

Bu örne■e https://github.com/electrocoder/IoThook/tree/master/examples/IoThook/v1_3/arduino sayfas■ndan ula■abilirsiniz.

```
/*
  Arduino ile ESP8266 Wifi Modul Testi

  Kod Arduino ya yuklendiginde Arduino IDE nin Serial Monitor u
  ile ESP8266 arasinda haberlesme gozlenebilir.

  Arduino ile ESP8266 arasindaki iletisim Baud ayari
  115200 olmalidir.

  Arduino 0 ile 100 arasinda uretmis oldugu Random say■y■ iothook a gonderir.

  Bu cihaza ait datalar
  https://iothook.com/tr/channel/api/public/240
  adresinden gercek zamanli olarak izlenebilir.

  Bu ornek IoThook servisine veri gondermek icin baslangic ayarlarinin
  yapilmasini amaclamaktadir.

  24 Eylul 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/IoThook

  sitelerine gidiniz.
  Sorular ve destek talepleri icin
  https://github.com/electrocoder/IoThook/issues
  sayfasina gidiniz.

  Yayin ve sahiplik http://mesebilisim.com
*/

#include "SoftwareSerial.h"

String ssid = "WIFI_ID";
String password = "WIFI_PASSWORD";

SoftwareSerial esp(10, 11);// RX, TX

String data;

String server = "iothook.com";

String uri = "/api/latest/datas/";
```

```
void setup() {

  esp.begin(115200);

  Serial.begin(115200);

  Serial.println("Arduino ile ESP8266 Wifi Modul Testi");
  Serial.println("          www.IoThook.com           ");
  Serial.println("");

  reset();

  connectWifi();

}


void reset() {

  esp.println("AT+RST");

  delay(2000);

  if (esp.find("OK") ) Serial.println("Modul Reset yapildi");
  else Serial.println("Module Reset yap■lamadi");

}


void connectWifi() {

  String cmd = "AT+CWJAP=\"" + ssid + "\",\"" + password + "\"";

  esp.println(cmd);

  delay(4000);

  if (esp.find("OK")) {

    Serial.println("ESP8266 Wifi ye baglandi");

  }

  else {

    connectWifi();

    Serial.println("ESP8266 Wifi ye baglanamad■!");
  }

}


void loop () {

  data = "{\"api_key\":\"b4301a9f-9854-11790bdf8d320140da\",\"value_1\":" + String(random(0,

  httppost();

  delay(5000);
```

```
}


void httppost () {

  esp.println("AT+CIPSTART=\"TCP\",\"" + server + "\",80");

  if ( esp.find("OK")) {

    Serial.println("TCP baglanti hazir");

  }
  else
    Serial.println("TCP baglanti hatali");

  delay(3000);

  String postRequest =

    "POST " + uri + " HTTP/1.0\r\n" +

    "Host: " + server + "\r\n" +

    "Accept: *" + "/" + "*\r\n" +

    "Content-Length: " + data.length() + "\r\n" +

    "Content-Type: application/x-www-form-urlencoded\r\n" +

    "\r\n" + data;

  String sendCmd = "AT+CIPSEND=";

  esp.print(sendCmd);

  esp.println(postRequest.length() );

  delay(1500);

  if (esp.find(">")) {
    Serial.println("Gonderiliyor...");
    esp.print(postRequest);

    if ( esp.find("SEND OK")) {
      Serial.println("Gonderildi :)");

      while (esp.available()) {

        String tmpResp = esp.readString();

        Serial.println(tmpResp);

      }

      esp.println("AT+CIPCLOSE");

    }
    else
      Serial.println("Gonderilemedi :(");
```

```
  }
  else
    Serial.println("Gonderim hatasi! ESP hazir degil!");
}
```

Arduino, ESP8266 POST Metodu ile 2 Veri Gönderme

Bu örnekde Arduino Uno ya RX ve TX ile ba■lanm■■ olan ESP8266 ile iothook a veri gonderme örne■i verilmi■tir. Örnekde 0-100 aras■nda rastgele 2 say■ üretilerek iothook da https://iothook.com/tr/channel/api/public/192 id numaral■ cihaz için gönderim gerçekle■mi■tir. Cihaz datalar■n■ https://iothook.com/tr/channel/api/public/192 linkinden gercek zamanl■ olarak takip edebilirsiniz.

Bu örne■e https://github.com/electrocoder/IoThook/tree/master/examples/IoThook/v1_3/arduino sayfas■ndan ula■abilirsiniz.

```
/*
  Arduino ile ESP8266 Wifi Modul Testi

  Kod Arduino ya yuklendiginde Arduino IDE nin Serial Monitor u
  ile ESP8266 arasinda haberlesme gozlenebilir.

  Arduino ile ESP8266 arasindaki iletisim Baud ayari
  115200 olmalidir.

  Arduino 0 ile 100 arasinda uretmis oldugu 2 adet Random say■y■ iothook a gonderir.
  Bu say■lar 'data' de■i■keni içerisinde value_1 ve value_2 de■erleridir. Bu de■erler
  sensör olarak kullan■lmaktad■r. S■cakl■k  ve Nem gibi sensörlerinizi bu alanlara
  gönderebilirsiniz.

  Bu cihaza ait datalar
  https://iothook.com/tr/channel/api/public/192
  adresinden gercek zamanli olarak izlenebilir.

  Bu ornek IoThook servisine veri gondermek icin baslangic ayarlarinin
  yapilmasini amaclamaktadir.

  24 Eylul 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/IoThook

  sitelerine gidiniz.
  Sorular ve destek talepleri icin
  https://github.com/electrocoder/IoThook/issues
  sayfasina gidiniz.

  Yayin ve sahiplik http://mesebilisim.com
*/

#include "SoftwareSerial.h"

String ssid = "WIFI_SSID";
String password = "WIFI_PASSWORD";

SoftwareSerial esp(10, 11);// RX, TX
```

```
String data;

String server = "iothook.com";

String uri = "/api/latest/datas/";

void setup() {

  esp.begin(115200);

  Serial.begin(115200);

  Serial.println("Arduino ile ESP8266 Wifi Modul Testi");
  Serial.println("          www.IoThook.com           ");
  Serial.println("");

  reset();

  connectWifi();

}


void reset() {

  esp.println("AT+RST");

  delay(2000);

  if (esp.find("OK") ) Serial.println("Modul Reset yapildi");
  else Serial.println("Module Reset yap■lamadi");

}


void connectWifi() {

  String cmd = "AT+CWJAP=\"" + ssid + "\",\"" + password + "\"";

  esp.println(cmd);

  delay(4000);

  if (esp.find("OK")) {

    Serial.println("ESP8266 Wifi ye baglandi");

  }

  else {

    connectWifi();

    Serial.println("ESP8266 Wifi ye baglanamad■!");
  }

}


void loop () {
```

```
  data = "{\"api_key\":\"5180e8bd-95a5-11cc4ce6cfe4ee481c\",\"value_1\":" + String(random(0,

  httppost();

  delay(8000);

}


void httppost () {

  esp.println("AT+CIPSTART=\"TCP\",\"" + server + "\",80");

  if ( esp.find("OK")) {

    Serial.println("TCP baglanti hazir");

  }
  else
    Serial.println("TCP baglanti hatali");

  delay(3000);

  String postRequest =

    "POST " + uri + " HTTP/1.0\r\n" +

    "Host: " + server + "\r\n" +

    "Accept: *" + "/" + "*\r\n" +

    "Content-Length: " + data.length() + "\r\n" +

    "Content-Type: application/x-www-form-urlencoded\r\n" +

    "\r\n" + data;

  String sendCmd = "AT+CIPSEND=";

  esp.print(sendCmd);

  esp.println(postRequest.length() );

  delay(1500);

  if (esp.find(">")) {
    Serial.println("Gonderiliyor...");
    esp.print(postRequest);

    if ( esp.find("SEND OK")) {
      Serial.println("Gonderildi :)");

      while (esp.available()) {

        String tmpResp = esp.readString();

        Serial.println(tmpResp);

      }
```

```
      esp.println("AT+CIPCLOSE");

    }
    else
      Serial.println("Gonderilemedi :(");

  }
  else
    Serial.println("Gonderim hatasi! ESP hazir degil!");
}
```

Arduino, ESP8266, Nodemcu GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde Arduino, ESP8266 ve NodeMCU ile ile Get metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 18.09.2017
// nodemcu ile sicaklik ve nem takibi
// electrocoder@gmail.com
// sahin mersin
// v1

#include <ESP8266WiFi.h>          //https://github.com/esp8266/Arduino

//needed for library
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>          //https://github.com/tzapu/WiFiManager

//for LED status
#include <Ticker.h>

#include <ESP8266HTTPClient.h>

#include "DHT.h"

#define DHTPIN 4     // what digital pin we're connected to   // D2 - GPIO4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

Ticker ticker;

void tick()
{
  //toggle state
  int state = digitalRead(BUILTIN_LED);  // get the current state of GPIO1 pin
  digitalWrite(BUILTIN_LED, !state);     // set pin to the opposite state
}

//gets called when WiFiManager enters configuration mode
void configModeCallback (WiFiManager *myWiFiManager) {
  Serial.println("Entered config mode");
  Serial.println(WiFi.softAPIP());
  //if you used auto generated SSID, print it
  Serial.println(myWiFiManager->getConfigPortalSSID());
  //entered config mode, make led toggle faster
  ticker.attach(0.2, tick);
}
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  //set led pin as output
  pinMode(BUILTIN_LED, OUTPUT);

  // start ticker with 0.5 because we start in AP mode and try to connect
  ticker.attach(0.6, tick);

  //WiFiManager
  //Local intialization. Once its business is done, there is no need to keep it around
  WiFiManager wifiManager;
  //reset settings - for testing
  //wifiManager.resetSettings();

  //set callback that gets called when connecting to previous WiFi fails, and enters Access
  wifiManager.setAPCallback(configModeCallback);

  //fetches ssid and pass and tries to connect
  //if it does not connect it starts an access point with the specified name
  //here  "AutoConnectAP"
  //and goes into a blocking loop awaiting configuration
  if (!wifiManager.autoConnect("MeseIoT", "MeseIoT**")) {
    Serial.println("failed to connect and hit timeout");
    //reset and try again, or maybe put it to deep sleep
    ESP.reset();
    delay(1000);
  }

  //if you get here you have connected to the WiFi
  Serial.println("connected...yeey :)");
  ticker.detach();
  //keep LED on
  digitalWrite(BUILTIN_LED, LOW);

  dht.begin();
}


void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
```

```
// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t");
Serial.print("Heat index: ");
Serial.print(hic);
Serial.print(" *C ");
Serial.print(hif);
Serial.println(" *F");

///
HTTPClient http;

// configure server and url
http.begin("http://iothook.com/api/latest/datas/update/?api_key=095c75f-9c40-11e14084d3e&v
//http.begin("192.168.1.12", 80, "/test.html");

Serial.print("[HTTP] GET...\n");
// start connection and send HTTP header
int httpCode = http.GET();
if (httpCode > 0) {
  // HTTP header has been send and Server response header has been handled
  Serial.printf("[HTTP] GET... code: %d\n", httpCode);

  // file found at server
  if (httpCode == HTTP_CODE_OK) {

    // get lenght of document (is -1 when Server sends no Content-Length header)
    int len = http.getSize();

    // create buffer for read
    uint8_t buff[128] = { 0 };

    // get tcp stream
    WiFiClient * stream = http.getStreamPtr();

    // read all data from server
    while (http.connected() && (len > 0 || len == -1)) {
      // get available data size
      size_t size = stream->available();

      if (size) {
        // read up to 128 byte
        int c = stream->readBytes(buff, ((size > sizeof(buff)) ? sizeof(buff) : size));

        // write it to Serial
        Serial.write(buff, c);

        if (len > 0) {
          len -= c;
        }
```

```
        }
        delay(1);
      }

      Serial.println();
      Serial.print("[HTTP] connection closed or file end.\n");

    }
  } else {
    Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str())
  }

  http.end();
  ////
  delay(13000);
}
```

GO GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde GO dili ile ile Get metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 04 Eylul 2017
// Sahin MERSIN
// iothook.com
// postman kullanilarak olusturulmustur


package main

import (
    "fmt"
    "net/http"
    "io/ioutil"
)

func main() {

    url := "http://iothook.com/api/latest/datas/update?api_key=22dbb35d-9dd5-113c0200ec44bb9

    req, _ := http.NewRequest("GET", url, nil)

    req.Header.Add("cache-control", "no-cache")

    res, _ := http.DefaultClient.Do(req)

    defer res.Body.Close()
    body, _ := ioutil.ReadAll(res.Body)

    fmt.Println(res)
    fmt.Println(string(body))

}
```

PHP GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde PHP dili ile ile Get metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 04 Eylul 2017
// Sahin MERSIN
// iothook.com
// postman kullanilarak olusturulmustur

<?php

$request = new HttpRequest();
$request->setUrl('http://iothook.com/api/latest/datas/update');
$request->setMethod(HTTP_METH_GET);

$request->setQueryData(array(
  'api_key' => '22dbb35d-9dd5-12300ec44bb9',
  'value_1' => '10',
  'value_2' => '2',
  'value_3' => '3'
));

$request->setHeaders(array(
  'postman-token' => '791ba738-7cb8-a920-0e5c-883cfb3e4498',
  'cache-control' => 'no-cache'
));

try {
  $response = $request->send();

  echo $response->getBody();
} catch (HttpException $ex) {
  echo $ex;
}
```

NodeJS GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde NodeJS Native metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 04 Eylul 2017
// Sahin MERSIN
// iothook.com
// postman kullanilarak olusturulmustur

var http = require("http");

var options = {
  "method": "GET",
  "hostname": "iothook.com",
  "port": null,
  "path": "/api/latest/datas/update?api_key=22dbb35d-9dd5-113200ec44bb9&value_1=10&value_2=2
  "headers": {
    "cache-control": "no-cache",
    "postman-token": "033da3c8-6196-cd49-f72d-1850a7d18500"
  }
};

var req = http.request(options, function (res) {
  var chunks = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });
```

```
  res.on("end", function () {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });
});

req.end();
```

Javascript Jquery Ajax GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde NodeJS Native metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 04 Eylul 2017
// Sahin MERSIN
// iothook.com
// postman kullanilarak olusturulmustur

var settings = {
  "async": true,
  "crossDomain": true,
  "url": "http://iothook.com/api/latest/datas/update?api_key=22dbb35d-9dd5-113c0342c44bb9&va
  "method": "GET",
  "headers": {
    "cache-control": "no-cache",
  }
}

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

Java Unirest GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde NodeJS Native metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 04 Eylul 2017
// Sahin MERSIN
// iothook.com
// postman kullanilarak olusturulmustur

HttpResponse<String> response = Unirest.get("http://iothook.com/api/latest/datas/update?api_
  .header("cache-control", "no-cache")
  .asString();
```

Java Unirest GET Metodu ile Veri Gönderme

IoThook Api v1.3 güncellemesi ile GET metodu ile veri göndermeye izin vermektedir.

Bu örnekde NodeJS Native metodu kullanarak veri gönderme örne■i verilmi■tir:

```
// 04 Eylul 2017
// Sahin MERSIN
// iothook.com
// postman kullanilarak olusturulmustur

OkHttpClient client = new OkHttpClient();
```

```
Request request = new Request.Builder()
  .url("http://iothook.com/api/latest/datas/update?api_key=22dbb35d-9dd5-113c03420ec44bb9&va
  .get()
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Veri Al

Iot cihaz■ndan gönderilen ■s■, nem, voltaj, ■■■k gibi de■erleri iothook data merkezinden çekebilmek için öncelikle kanal üye kullan■c■ ad■ ve giri■ ■ifresineihtiyaç vard■r. Iot cihaz■ndan gelen veriler, Android, iOS gibi mobil cihaz■n■zdan veya web sitenizden izlenebilir. Kanal■n■z■n kullan■m■ genel kullan■ma aç■k ise di■er kullan■c■lar ile de bu verileri payla■abilirsiniz.

Örne■in; Kanal ad■m■z “Temperature sensor” olarak belirlenmi■ ve kanal içerisinde bulunacak iot cihaz■m■zdan “temperature” ve “humidity” element verileri gönderiliyor olsun.

Python 2, Python 3 Json ile Veri Alma

Python Json ile Get Örne■i:

Bu örne■i http://bit.ly/2jI1FNQ sayfas■ndan inceleyebilirsiniz.

```
# -*- coding: utf-8 -*-

"""
  Python 2, Python 3 ile IoThook REST Api Testi

  Kod çal■■t■r■ld■■■nda 'data' de■i■kenine verilen 'all' de■i■keni ile
  auth sahipli■indeki tüm veriler al■n■r.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  10 May■s 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

"""

import requests
```

```
API_KEY = '511b0173-95a5-11c814c2297e434c06'

url = 'https://iothook.com/api/latest/datas/?api_key=' + API_KEY

response = requests.get(url)
data = response.json()
print data
```

Python 2, Python 3 Json ile ■lk Veriyi Alma

Python ■lk Veriyi Alma, Json ile Get Örne■i:

Bu örne■i http://bit.ly/2jI1FNQ sayfas■ndan inceleyebilirsiniz.

```
# -*- coding: utf-8 -*-

"""
  Python 2 ile IoThook REST Api Testi

  Kod çal■■t■r■ld■■■nda 'data' de■i■kenine verilen 'first' de■i■keni ile
  auth sahipli■indeki ilk veri al■n■r. 'channel' de■i■keni Iothook dashboard
  Kanal olu■turma s■ras■nda otomatik verilen id numaras■d■r.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  10 May■s 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

"""

import requests

API_KEY = '511b0173-95a5-11c814c2297e434c06'

url = 'https://iothook.com/api/latest/datas/?data=first&api_key=' + API_KEY

response = requests.get(url)
data = response.json()
print data
```

Python 2, Python 3 Json ile Son Veriyi Alma

Python Son Veriyi Alma, Json ile Get Örne■i:

Bu örne■i http://bit.ly/2jI1FNQ sayfas■ndan inceleyebilirsiniz.

```
# -*- coding: utf-8 -*-

"""
  Python 2 ile IoThook REST Api Testi

  Kod çal■■t■r■ld■■■nda 'data' de■i■kenine verilen 'last' de■i■keni ile
  auth sahipli■indeki en son veri al■n■r. 'channel' de■i■keni Iothook dashboard
  Kanal olu■turma s■ras■nda otomatik verilen id numaras■d■r.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  10 May■s 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

"""

import requests


API_KEY = '516b0073-95a5-11c814c2597e434c06'

url = 'https://iothook.com/api/latest/datas/?data=last&api_key=' + API_KEY

response = requests.get(url)
data = response.json()
print data
```

Python 2, Python 3 Json ile Veriye Ait Detay Alma

Python veriye ait detay alma örne■i:

Bu örne■i http://bit.ly/2jI1FNQ sayfas■ndan inceleyebilirsiniz.

```
# -*- coding: utf-8 -*-
```

```
"""
  Python 2, 3 ile IoThook REST Api Testi

  Kod çal■■t■r■ld■■■nda datas url yap■s■na parametre olarak verilen
  de■er Kanal ve Element içerisinde tan■ml■ datan■n ayr■nt■lar■n■ getirir.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  10 May■s 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

"""

import requests

API_KEY = '516b0073-95a5-11c814c2297e434c06'

url = 'https://iothook.com/api/latest/datas/4545/?api_key=' + API_KEY

response = requests.get(url)
data = response.json()
print data
```

C# ile Json Format■nda Veri Alma

IoThook kullan■c■s■n■n tüm kanallar■na ait veriyi alabilmesi için GET metodu ile '?data=last' de■i■keninin gönderilmesi gereklidir. Ayn■ örnekdeki data de■i■keninin alabilece■i de■erler:

- ?data=all : Kullan■c■n■n tüm datalar■n■ getir
- ?data=first : Kullan■c■n■n ilk datas■n■ getir
- ?data=last : Kullan■c■n■n son datas■n■ getir

CSharp örne■inde *HttpWebRequest* metodu kullan■lm■■t■r.

Bu örne■i http://bit.ly/2jI1FNQ Github sayfas■ndan inceleyebilirsiniz.

```
/*
  C# ile IoThook REST Api Testi
```

```
  Bu örnek ile CSharp ve Request metodu ile kullan■c■n■n datalar■n■n get metodu ile al■nma
  'autorization' ile kullan■c■ ad■ ve parola de■eri verilmelidir.

  Bu ornek IotHook servisine veri almak/gondermek icin baslangic seviyesinde
  testlerin yapilmasini amaclamaktadir.

  29 Temmuz 2017
  Sahin MERSIN

  Daha fazlasi icin

  http://www.iothook.com
  ve
  https://github.com/electrocoder/iotHook

  sitelerine gidiniz.

  Sorular ve destek talepleri icin
  https://github.com/electrocoder/iotHook/issues
  sayfasindan veya Me■e Bili■im den yard■m alabilirsiniz.

  Yayin : http://mesebilisim.com

  Licensed under the Apache License, Version 2.0 (the "License").
  You may not use this file except in compliance with the License.
  A copy of the License is located at

  http://www.apache.org/licenses/

*/

using System;
using System.IO;
using System.Net;

namespace Iothook
{
    class Program
    {
        static void Main(string[] args)
        {
            string url = "";
            url = "https://iothook.com/api/v1.2/datas/?data=all"; // for all data

            var webRequest = (HttpWebRequest)WebRequest.Create(url);

            webRequest.Method = "GET";
            webRequest.ContentType = "application/json";
            webRequest.UserAgent = "Mozilla/5.0 (Windows NT 5.1; rv:28.0) Gecko/20100101 Fir
            webRequest.ContentLength = 0;

            string autorization = "USERNAME" + ":" + "PASSWORD";
            byte[] binaryAuthorization = System.Text.Encoding.UTF8.GetBytes(autorization);
            autorization = Convert.ToBase64String(binaryAuthorization);
            autorization = "Basic " + autorization;
            webRequest.Headers.Add("AUTHORIZATION", autorization);

            var webResponse = (HttpWebResponse)webRequest.GetResponse();

            if (webResponse.StatusCode != HttpStatusCode.OK)
```

```
                Console.WriteLine(webResponse.Headers.ToString());

            using (StreamReader reader = new StreamReader(webResponse.GetResponseStream()))
            {
                Console.WriteLine(reader.ReadToEnd());
                reader.Close();
                webRequest.Abort();
            }

            Console.ReadLine();
        }
    }
}
```

Email Besleme

Iot cihazlardan email alma

Iot cihazlardan email almak için ‘Kanal Ekle’ menüsünden kanal olu■turulurken ‘Email feed’ seçene■inin aktif edilmesi gerekir. Kanal olu■turulduktan sonrada email alma seçene■i de■i■tirilebilir. Güncelleme için ‘Kanal Liste’ menüsünden ‘Düzenle’ seçene■i alt■ndan yap■labilir.

Email besleme plan■

Iot cihaz■n■zdan veri geldi■inde email ile besleme almak için ‘STUDENT’, ‘PRO’, veya ‘ULTRA’ planlardan birisini tercih etmelisiniz.

Plan de■i■ikli■i için Ödeme sayfas■ndan size uygun plan■ seçerek email besleme alabilirsiniz.

Email besleme süresi

- Free plan email besleme süresi: 8 email, ~180 dakika aral■k ile
- Student plan email besleme süresi: 10 email, ~144 dakika aral■k ile
- Pro plan email besleme süresi: 15 email, ~96 dakika aral■k ile
- Ultra plan email besleme süresi: 100 email, ~14 dakika aral■k ile

Iot Email Sms Alarm

Alarm nedir?

Iot Kanal alt■nda olu■turulan Elementlere alarm de■eri kurma i■lemidir. Alarm de■eri kurularak iot cihaz■dan her veri al■nd■■■nda operatör ile i■lem yap■larak sonuca göre alarm üretilir. Üretilen alarm abonelik tipine göre bir günde en fazla at■labilecek email ve sms plan■na göre belirlenir.

Operatörler

■■lem operatörleri a■a■■daki gibidir:

- < : Küçüktür operatörü. a < b. gelen_deger < alarm_degeri. Iot cihazdan gönderilen de■er ile alarm de■erini kar■■la■t■r■r. ■■lem sonucu do■ru (True) ise alarm üretilir.
- <= : Küçük e■ittir operatörü. a <= b. gelen_deger <= alarm_degeri. Iot cihazdan gönderilen de■er ile alarm de■erini kar■■la■t■r■r. ■■lem sonucu do■ru (True) ise alarm üretilir.
- == : E■ittir operatörü. a == b. gelen_deger == alarm_degeri. Iot cihazdan gönderilen de■er ile alarm de■erini kar■■la■t■r■r. ■■lem sonucu do■ru (True) ise alarm üretilir.

- != : E■it de■il operatörü. a != b. gelen_deger != alarm_degeri. Iot cihazdan gönderilen de■er ile alarm de■erini kar■■la■t■r■r. ■■lem sonucu do■ru (True) ise alarm üretilir.
- >= : Büyük e■it operatörü. a >= b. gelen_deger >= alarm_degeri. Iot cihazdan gönderilen de■er ile alarm de■erini kar■■la■t■r■r. ■■lem sonucu do■ru (True) ise alarm üretilir.
- > : Büyüktür operatörü. a > b. gelen_deger > alarm_degeri. Iot cihazdan gönderilen de■er ile alarm de■erini kar■■la■t■r■r. ■■lem sonucu do■ru (True) ise alarm üretilir.

Örnek operatör i■lemleri:

- < Küçüktür operatörü python örnek:

```
a = 5
b = 7
a < b
True
```

```
a = 9
b = 7
a < b
False
```

- <= Küçük e■ittir operatörü python örnek :

```
a = 5
b = 7
a <= b
True
```

```
a = 7
b = 7
a <= b
True
```

- == E■ittir operatörü python örnek :

```
a = 5
b = 7
a == b
False
```

```
a = 7
b = 7
a == b
True
```

- != E■it de■il operatörü python örnek :

```
a = 5
b = 7
a != b
True
```

```
a = 7
b = 7
a != b
False
```

- >= Büyük e■it operatörü python örnek :

```
a = 5
b = 7
```

```
a >= b
False
```

```
a = 7
b = 7
a >= b
True
```

- > Büyüktür operatörü python örnek :

```
a = 5
b = 7
a > b
False
```

```
a = 9
b = 7
a > b
True
```

Email Alarm nedir?

Iot Kanal/Element alarm i■lemi uyguland■■■nda gelen de■er ile alarm de■eri mant■ksal operatör i■lem sonucuna göre kanal yöneticisine email gönderilir. Kay■t olur iken kullan■lan email adresi geçerli email adresidir. Günlük (24 saat) email gönderilme say■s■ üyelik plan■na göre hesaplan■r.

SMS Alarm nedir?

Iot Kanal/Element alarm i■lemi uyguland■■■nda gelen de■er ile alarm de■eri mant■ksal operatör i■lem sonucuna göre kanal yöneticisine sms gönderilir. Sms mesaj gönderilebilmesi için kanal yöneticisinin cep telefonunun onayl■ olmas■ gerekir. Günlük (24 saat) sms gönderilme say■s■ üyelik plan■na göre hesaplan■r.

Iot Mqtt Nedir?

MQTT Message Queuing Telemetry Transport kelimelerinin ba■ harfleri ile tan■d■■■m■z bu teknoloji mesaj■n kar■■ tarafa ula■t■r■lmas■ için kullan■lan haberle■me protokolüdür. Haberle■me için mesaj yay■nlayan, mesaja abone olan ve mesaj trafi■ini kontrol eden yöneticiden olu■maktad■r.

Mesaj trafi■ini kontrol eden yöneticiye BROKER, mesaj yay■n■na PUBLISH ve aboneye SUBSCRIBE denir. Mesaj al■■veri■i publisher dan subscriber lara do■ru yani yay■nc■lardan abonelere do■ru olmaktad■r.

MQTT

Mqtt Protokolü Nas■ld■r?

MQTT de asenkron haberle■me protokolü kullan■lmaktad■r. Mesaj yay■nc■lar■ ve mesaj al■c■lar aras■nda e■zamans■z olarak veri ta■■nmaktad■r. Di■er haberle■me yap■lar■na göre basit olu■u ve minimum kaynak tüketmesi sebebiyle “machine-to-machine” (M2M) makineden makineye veri iletiminde ve (IOT) “Internet of Things” ■nternete ba■l■ nesnelerin mesajla■mas■nda tercih edilmektedir.

MQTT Temp Test Client

Iot MQTT Temp Test Client Mosquitto Brokera websocket ile gelen temp/random ba■l■■■n■ dinler.

Bu örnek ‘test.mosquitto.org’ sitesinden al■nm■■t■r. MQTT Temp örne■inin orjinal kayna■■na ‘desert-home.com’ adresinden ve Github üzerinden ula■abilirsiniz. MQTT Brokera nas■l mesaj gönderebilirim?

Iothook MQTT broker■na veri göndermek için “temp/random” ba■l■■■ gönderilmelidir. Gönderilen de■er -20 ile +50 aral■■■nda kayar noktal■ (float) veya tamsay■ (int) format■nda olmal■d■r.

MQTT Broker kimlik do■rulama ile çal■■■r.

Örnek -> Mesaj yay■nlama: mosquitto_pub -h iothook.com -p 1883 -t “temp/random” -m “6” -u pub_user -P iothook_pub_user

Örnek -> Mesaja abone olma: mosquitto_sub -h iothook.com -p 1883 -t “temp/random” -u pub_user -P iothook_pub_user

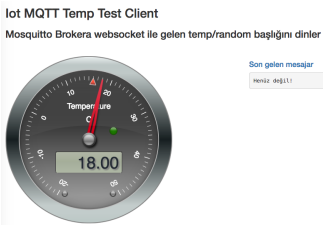
MQTT Brokera için Test Kullan■c■lar■:

Kullan■c■ Ad■ : pub_user ■ifre : iothook_pub_user

Kullan■c■ Ad■ : sub_user ■ifre : iothook_sub_user

Kullan■c■ Ad■ : pub_client ■ifre : iothook_pub_client

Kullan■c■ Ad■ : sub_client ■ifre : iothook_sub_client



MQTT Temp Test

Temp client sayfas■na https://iothook.com/mqtt/mqtt-temp-test/ adresinden ula■abilirsiniz.

MQTT Test Client Publisher Subscriber

MQTT Brokera Mesaj Gönderme ve Abone Olma Mosquitto Brokera websocket ile gelen temp/random ba■l■■■n■ dinler.

Iothook MQTT broker■na veri göndermek için “temp/random” ba■l■■■ gönderilmelidir. Gönderilen veri kayar noktal■ (float), tamsay■ (int) veya string (text) format■nda olabilir.

MQTT Broker kimlik do■rulama ile çal■■■r.

Örnek -> Mesaj yay■nlama: mosquitto_pub -h iothook.com -p 1883 -t “temp/random” -m “6” -u pub_user -P iothook_pub_user

Örnek -> Mesaja abone olma: mosquitto_sub -h iothook.com -p 1883 -t “temp/random” -u pub_user -P iothook_pub_user

MQTT Brokera için Test Kullan■c■lar■:

Kullan■c■ Ad■ : pub_user ■ifre : iothook_pub_user

Kullan■c■ Ad■ : sub_user ■ifre : iothook_sub_user

Kullan■c■ Ad■ : pub_client ■ifre : iothook_pub_client

Kullan■c■ Ad■ : sub_client ■ifre : iothook_sub_client

MQTT Test Publisher Subscriber

Temp client sayfas■na https://iothook.com/mqtt/mqtt-temp-test-pub-sub/ adresinden ula■abilirsiniz.

Full Featured MQTT Client

MQTT Brokera Mesaj Gönderme ve Alma

MQTT Broker kimlik do■rulama ile çal■■■r.

Örnek -> Mesaj yay■nlama: mosquitto_pub -h iothook.com -p 1883 -t “temp/random” -m “6” -u pub_user -P iothook_pub_user

Örnek -> Mesaja abone olma: mosquitto_sub -h iothook.com -p 1883 -t “temp/random” -u pub_user -P iothook_pub_user

MQTT Brokera için Test Kullan■c■lar■:

Kullan■c■ Ad■ : pub_user ■ifre : iothook_pub_user

Kullan■c■ Ad■ : sub_user ■ifre : iothook_sub_user

Kullan■c■ Ad■ : pub_client ■ifre : iothook_pub_client

Kullan■c■ Ad■ : sub_client ■ifre : iothook_sub_client

Websockets Client Uygulamas■ Apache License Version 2.0 ile HiveMQ http://www.hivemq.com/ taraf■ndan da■■t■lmaktad■r. Lisans hakk■nda ayr■ca bilgi al■n■z.

Temp client sayfas■na https://iothook.com/mqtt/full-featured-mqtt-client/ adresinden ula■abilirsiniz.

IHook Nedir?

Iot Dashboard Nedir?

Banana Pi, NanoPC, Intel Edison, Parallella, Raspberry Pi gibi tek kart bilgisayarlarda çal■■an Python/Django REST framework ile geli■tirilmi■ Web Api servisidir. IOT cihazlar ile ileti■ime geçerek Web Api sayesinde GET, POST, PUT ve DELETE i■lemlerini kolayca yapabilmek için tasarlanm■■t■r. Iotdashboard tüm cihazlar■n■z aras■nda kesintisiz veri aktar■m■ yapan, internete ba■l■ nesnelerin kolayca ula■abilece■i ileti■im protokollerini destekler. Google developer chart apileri ile entegre olarak verileri gerçek zamanl■ izleme olana■■ sa■lar. Proje iOTHOOK taraf■ndan aç■k kaynak olarak geli■tirilmi■ ve MIT lisans■ ile da■■t■lmaktad■r. Kaynak kodlara http://electrocoder.github.io/iotdashboard/ buradan ula■abilirsiniz.

IHook GITHUB

Iot Dashboard GITHUB Iot dashboard projesi Raspberry Pi türevi tek kart bilgisayarlar için geli■tirilmi■ Django Rest framewok server projesidir. Proje iOTHOOK taraf■ndan aç■k kaynak olarak geli■tirilmi■ ve MIT lisans■ ile da■■t■lmaktad■r. Kaynak kodlara http://electrocoder.github.io/iotdashboard/ buradan ula■abilirsiniz.