IoThook Documentation

Sürüm 1.3

electrocoder

Eyl 20, 2017

İçindekiler

1 Quick reference **1**

1.1 Iot Nedir 1

1.2 Kanal Aç 2

1.3 Veri Gönder 3

1.4 Veri Al 11

1.5 Email Besleme 19

1.6 Iot Email Sms Alarm 19

1.7 Iot Mqtt Nedir? 22

1.8 Mqtt Protokolü Nasıldır? 22

1.9 MQTT Temp Test Client 22

1.10 MQTT Test Client Publisher Subscriber 23

1.11 Full Featured MQTT Client 23

1.12 IHook Nedir? 24

BÖLÜM 1

Quick reference

Iot Nedir

Iot Nedir?

Nesnelerin interneti “internet of things” 1999 yılında Kevin Ashton tarafından kullanılan bir kavramdır ve teknolojideki gelişmeler ile birlikte bugünkü haline gelmiştir. RFID teknolojisi için üretilen bu kavram günümüzde tüm elektronik cihazlara uygulanmaktadır.

=0.65=0.65@default

Şekil 1.1: *IoT - Nesnelerin İnterneti*

Iothook nedir?

Iothook internete bağlı nesneler (iot) arasında veri transferi yapan web servis ağı 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 tabanlı sistemleri birbirine bağlar.

Niçin Iothook?

- Iothook hızlıdır,
- Sınırsız kanal oluşturabilirsin,
- Sınırsız element ekleyebilirsin,
- Tüm cihazların ile kolayca veri gönderebilirsin (post),
- Tüm iot cihazlarından kolayca veri alabilirsin (get),
- Datalarını gerçek zamanlı takip edebilirsin,

- Dataların için gerçek zamanlı grafik oluşturabilirsin,

Iothook' un sunduğu avantajlar:

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

Iothook tüm cihazlarınız arasında kesintisiz veri aktarımı yapan, internete bağlı nesnelerin kolayca ulaşabileceği iletişim protokollerini destekler.

Google developer chart apileri ile entegre olarak verileri gerçek zamanlı izleme olanağı sağlar.

Kanal Aç

Iothook kanal; internete bağlı nesneler arasında veri iletimini sağlamak için oluşturulmuş kanca sistemidir. Kanal ile iot sistemleri veri paylaşımı yapılabilir, veri gönderim işlemleri tanımlanır.

Iothook web servislerini kullanabilmek için üye olunmalıdır. Üyelik seçenekleri 'Free', 'Student', 'Pro' ve 'Ultra' olmak üzere 4 kullanım planı vardır. Üye olmak için adrese gidiniz.

Üyelik adımından sonra yönetim paneli aracılığı ile 'Kanal Ekle' ekranına girilir.

Kanal Ekle

Kanal ekleme adımları şu şekildedir:

- **Form Metod: Http (Hyper Text Transfer Protocol) de veriler TCP/IP metodu ile iletilmektedir. Http protokolü üzerinden**
    - Post: Verilerin iot cihazda mesaj gövdesine yerleştirilerek gönderilme işlemidir.
    - Get: Verilerin iot cihaz ile sorgulanma ve cevap alınma talebidir.
    - Post/Get: Veri aktarımının iot nesnesi ile server arasında çift taraflı olacağını gösterir.
- Form enctype: "application/x-www-form-urlencoded" ile iot cihazından gönderilen karakterlerin gönderilmeden önce kodlanacağını belirtir. "mutlipart/form-data" ise verilerin içerisinde ASCII olmayan verilerin olduğunu dosya veya image formatında veri olduğunu belirtir.
- Aygıt türü: Iot cihazın türünü belirler. Arduino, Raspberry Pi... gibi
- Kanal adı: Verilerin toplanacağı kanalın adı.
- Web site: Veriler bir web sitesinde kullanılacak ise web site adresi girilmelidir.

- Email ile haber ver: Veri alındığında kayıtlı olan mail adresine mesaj gonderir. Aktif edilirse 15dk. da bir veri gönderilmesi gerekir.
- Verileri kaydet: Iot nesnesinden gelen verilerin iothook veritabanında saklanması için gereklidir.
- Resim: Kanal tanıtım resmi olarak kullanılır.
- Açıklama: Kanal bilgileri girilmelidir.
- Is public POST: Bu kanal genel kullanıma açık ve veri eklenmesine açıktır.
- Is public GET: Bu kanal genel kullanıma açık ve verilerin okunmasına izin verir.
- Yayındamı: Kanalı aktif et.

Element Ekle

Iot cihazınız için kanal oluşturduktan sonra kanalda bulunmasını istediğiniz veri alanlarını oluşturmalısınız. Bu alanlar veri almaya başlamak için eklenir. Element verilerine POST veta GET metodu ile ulaşabilirsiniz.

Element ayarları:

- Kanal adı: Elementin hangi kanala veri aktaracağı seçilir.
- Grafik türü: Toplanan verilerin çizileceği grafik türünü belirler.
- Element tipi: Verilerin depolanacağı alan tipini belirler. Grafik çizimi sadece "number" veri tipinde yapılmalıdır.
- Kanal adı: Verilerin toplanacağı kanalın adı.
- Element adı: Verilerin tutulacağı element adı.
- Yayındamı: Elementi aktif et.

Veri Gönder

Veri göndermek için öncelikle kanal ve element eklemeniz gerekir. Kanal oluşturulduğunda size özel "api_key" üretilerek belirlenen erişim metoduna göre (POST, GET, POST/GET) veri işlemi gerçekleştirilir.

Örneğin; Kanalımız ısı, ışık, hareket, bar ve nem değerlerini alan bir yapıda olsun. Kanal içerisinde bulunacak iot cihazlarımız bizlere bu dataları 15 sn. yede bir 100 kere göndersin.

Oluşturulan "API_KEY" Key Yöneticisi sayfasından görülebilir.

Python ile JSON Veri Gönderme

Python ile Json Post Örneği:

Bu örneği http://bit.ly/2jI1FNQ sayfasından indirebilirsiniz.

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{\PYGZsh{} \PYGZhy{}*\PYGZhy{} coding: utf\PYGZhy{}8 \PYGZhy{}*\PYGZhy{}}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}
\PYG{l+s+sd}{  Python ile IoThook REST Api Testi}
```

```
\PYG{l+s+sd}{  Kod çalıştırıldığında APIKEY ile doğrulama gerçekleştirilir.}
\PYG{l+s+sd}{  Kanal api\PYGZus{}key ile ilgili kanal ve element değerleri IoThook a␣
↪post edilir.}

\PYG{l+s+sd}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{l+s+sd}{  testlerin yapilmasini amaclamaktadir.}

\PYG{l+s+sd}{  2 Eylul 27}
\PYG{l+s+sd}{  Sahin MERSIN}

\PYG{l+s+sd}{  Daha fazlasi icin}

\PYG{l+s+sd}{  http://www.iothook.com}
\PYG{l+s+sd}{  ve}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook}

\PYG{l+s+sd}{  sitelerine gidiniz.}

\PYG{l+s+sd}{  Sorular ve destek talepleri icin}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook/issues}
\PYG{l+s+sd}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}

\PYG{l+s+sd}{  Yayin : http://mesebilisim.com}

\PYG{l+s+sd}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{l+s+sd}{  You may not use this file except in compliance with the License.}
\PYG{l+s+sd}{  A copy of the License is located at}

\PYG{l+s+sd}{  http://www.apache.org/licenses/}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}

\PYG{k+kn}{import} \PYG{n+nn}{requests}
\PYG{k+kn}{import} \PYG{n+nn}{json}
\PYG{k+kn}{import} \PYG{n+nn}{random}
\PYG{k+kn}{import} \PYG{n+nn}{pprint}
\PYG{k+kn}{import} \PYG{n+nn}{time}



\PYG{n}{headers} \PYG{o}{=}␣
↪\PYG{p}{\PYGZob{}}\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{Content\PYGZhy{}type}\PYG{l+s+s1}{\PYGZsq{}}
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{application/
↪json}\PYG{l+s+s1}{\PYGZsq{}}\PYG{p}{\PYGZcb{}}

\PYG{n}{url} \PYG{o}{=} \PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{https://iothook.com/api/
↪latest/datas/}\PYG{l+s+s1}{\PYGZsq{}}

\PYG{k}{for} \PYG{n}{i} \PYG{o+ow}{in}␣
↪\PYG{n+nb}{range}\PYG{p}{(}\PYG{l+m+mi}{1}\PYG{p}{)}\PYG{p}{:}
    \PYG{n}{data}\PYG{o}{=}\PYG{p}{\PYGZob{}}

       ␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{api\PYGZus{}key}\PYG{l+s+s1}{\PYGZsq{}}\PYG{p}{:}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{519ac5d4\PYGZhy{}95a5\PYGZhy{}116e185a343eac447b}\PYG{l+s+s1}{
↪} \PYG{c+c1}{\PYGZsh{} demo hesap api\PYGZus{}key}
```

```
        ␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{value\PYGZus{}1}\PYG{l+s+s1}{\PYGZsq{}}\PYG{p}{:}␣
↪\PYG{n}{i}\PYG{o}{*}\PYG{l+m+mi}{1}\PYG{p}{,}
        \PYG{p}{\PYGZcb{}}

    \PYG{n}{data\PYGZus{}json} \PYG{o}{=} \PYG{n}{json}\PYG{o}{.
↪}\PYG{n}{dumps}\PYG{p}{(}\PYG{n}{data}\PYG{p}{)}

    \PYG{n}{response} \PYG{o}{=} \PYG{n}{requests}\PYG{o}{.
↪}\PYG{n}{post}\PYG{p}{(}\PYG{n}{url}\PYG{p}{,}␣
↪\PYG{n}{data}\PYG{o}{=}\PYG{n}{data\PYGZus{}json}\PYG{p}{,}␣
↪\PYG{n}{headers}\PYG{o}{=}\PYG{n}{headers}\PYG{p}{)}
    \PYG{n}{pprint}\PYG{o}{.}\PYG{n}{pprint}\PYG{p}{(}\PYG{n}{response}\PYG{o}{.
↪}\PYG{n}{json}\PYG{p}{(}\PYG{p}{)}\PYG{p}{)}
    \PYG{n}{time}\PYG{o}{.}\PYG{n}{sleep}\PYG{p}{(}\PYG{l+m+mi}{15}\PYG{p}{)}
```

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 Örneği:

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{\PYGZsh{} \PYGZhy{}*\PYGZhy{} coding: utf\PYGZhy{}8 \PYGZhy{}*\PYGZhy{}}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}
\PYG{l+s+sd}{  Python ile IoThook REST Api Testi}

\PYG{l+s+sd}{  Kod çalıştırıldığında APIKEY ile gonderim gerçekleştirilir.}
\PYG{l+s+sd}{  Kanal api\PYGZus{}key ile ilgili kanal ve element değerleri IoThook a␣
↪GET metodu ile gonderilir.}

\PYG{l+s+sd}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{l+s+sd}{  testlerin yapilmasini amaclamaktadir.}

\PYG{l+s+sd}{  11 Eylul 27}
\PYG{l+s+sd}{  Sahin MERSIN}

\PYG{l+s+sd}{  Daha fazlasi icin}

\PYG{l+s+sd}{  http://www.iothook.com}
\PYG{l+s+sd}{  ve}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook}

\PYG{l+s+sd}{  sitelerine gidiniz.}

\PYG{l+s+sd}{  Sorular ve destek talepleri icin}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook/issues}
\PYG{l+s+sd}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}

\PYG{l+s+sd}{  Yayin : http://mesebilisim.com}
```

```
\PYG{l+s+sd}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{l+s+sd}{  You may not use this file except in compliance with the License.}
\PYG{l+s+sd}{  A copy of the License is located at}

\PYG{l+s+sd}{  http://www.apache.org/licenses/}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}

\PYG{k+kn}{import} \PYG{n+nn}{requests}
\PYG{k+kn}{import} \PYG{n+nn}{json}
\PYG{k+kn}{import} \PYG{n+nn}{random}
\PYG{k+kn}{import} \PYG{n+nn}{pprint}
\PYG{k+kn}{import} \PYG{n+nn}{time}



\PYG{n}{headers} \PYG{o}{=}␣
↪\PYG{p}{\PYGZob{}}\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{Content\PYGZhy{}type}\PYG{l+s+s1}{\PYGZsq{}}\
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{application/
↪json}\PYG{l+s+s1}{\PYGZsq{}}\PYG{p}{\PYGZcb{}}

\PYG{n}{API\PYGZus{}KEY} \PYG{o}{=}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{85c4ba5f\PYGZhy{}96ae\PYGZhy{}11841415634e983487e}\PYG{l+s+s1}

\PYG{k}{for} \PYG{n}{i} \PYG{o+ow}{in}␣
↪\PYG{n+nb}{range}\PYG{p}{(}\PYG{l+m+mi}{1}\PYG{p}{)}\PYG{p}{:}
    \PYG{n}{url} \PYG{o}{=} \PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{https://iothook.com/
↪api/latest/datas/update?api\PYGZus{}key=}\PYG{l+s+s1}{\PYGZsq{}} \PYG{o}{+}␣
↪\PYG{n}{API\PYGZus{}KEY} \PYG{o}{+}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{\PYGZam{}value\PYGZus{}1=1
PYGZam{}value\PYGZus{}2=2\PYGZam{}value\PYGZus{}3=3}\PYG{l+s+s1}{\PYGZsq{}}

    \PYG{n}{response} \PYG{o}{=} \PYG{n}{requests}\PYG{o}{.
↪}\PYG{n}{get}\PYG{p}{(}\PYG{n}{url}\PYG{p}{)}
    \PYG{n}{data} \PYG{o}{=} \PYG{n}{response}\PYG{o}{.
↪}\PYG{n}{json}\PYG{p}{(}\PYG{p}{)}
    \PYG{k}{print} \PYG{n}{data}
    \PYG{n}{time}\PYG{o}{.}\PYG{n}{sleep}\PYG{p}{(}\PYG{l+m+mi}{15}\PYG{p}{)}
```

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 örneği verilmiştir:

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{// 18..27}
\PYG{c+c1}{// nodemcu ile sicaklik ve nem takibi}
\PYG{c+c1}{// electrocoder@gmail.com}
\PYG{c+c1}{// sahin mersin}
\PYG{c+c1}{// v1}
```

```
\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZlt{}ESP8266WiFi.h\PYGZgt{} ␣
↪        //https://github.com/esp8266/Arduino}

\PYG{c+c1}{//needed for library}
\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZlt{}DNSServer.h\PYGZgt{}}
\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZlt{}ESP8266WebServer.
↪h\PYGZgt{}}
\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZlt{}WiFiManager.h\PYGZgt{} ␣
↪        //https://github.com/tzapu/WiFiManager}

\PYG{c+c1}{//for LED status}
\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZlt{}Ticker.h\PYGZgt{}}

\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZlt{}ESP8266HTTPClient.
↪h\PYGZgt{}}

\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{include} \PYG{c+cpf}{\PYGZdq{}DHT.h\PYGZdq{}}

\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{define DHTPIN 4     }\PYG{c+c1}{// what digital pin␣
↪we\PYGZsq{}re connected to   // D2 \PYGZhy{} GPIO4}
\PYG{c+cp}{\PYGZsh{}}\PYG{c+cp}{define DHTTYPE DHT11}

\PYG{n}{DHT} \PYG{n+nf}{dht}\PYG{p}{(}\PYG{n}{DHTPIN}\PYG{p}{,}␣
↪\PYG{n}{DHTTYPE}\PYG{p}{)}\PYG{p}{;}

\PYG{n}{Ticker} \PYG{n}{ticker}\PYG{p}{;}

\PYG{k+kt}{void} \PYG{n+nf}{tick}\PYG{p}{(}\PYG{p}{)}
\PYG{p}{\PYGZob{}}
  \PYG{c+c1}{//toggle state}
  \PYG{k+kt}{int} \PYG{n}{state} \PYG{o}{=}␣
↪\PYG{n}{digitalRead}\PYG{p}{(}\PYG{n}{BUILTIN\PYGZus{}LED}\PYG{p}{)}\PYG{p}{;} ␣
↪\PYG{c+c1}{// get the current state of GPIO1 pin}
  \PYG{n}{digitalWrite}\PYG{p}{(}\PYG{n}{BUILTIN\PYGZus{}LED}\PYG{p}{,} \PYG{o}{!
↪}\PYG{n}{state}\PYG{p}{)}\PYG{p}{;}     \PYG{c+c1}{// set pin to the opposite state}
\PYG{p}{\PYGZcb{}}

\PYG{c+c1}{//gets called when WiFiManager enters configuration mode}
\PYG{k+kt}{void} \PYG{n+nf}{configModeCallback} \PYG{p}{(}\PYG{n}{WiFiManager}␣
↪\PYG{o}{*}\PYG{n}{myWiFiManager}\PYG{p}{)} \PYG{p}{\PYGZob{}}
  \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{println}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{Entered config␣
↪mode}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
  \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{println}\PYG{p}{(}\PYG{n}{WiFi}\PYG{p}{.
↪}\PYG{n}{softAPIP}\PYG{p}{(}\PYG{p}{)}\PYG{p}{)}\PYG{p}{;}
  \PYG{c+c1}{//if you used auto generated SSID, print it}
  \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{println}\PYG{p}{(}\PYG{n}{myWiFiManager}\PYG{o}{\PYGZhy{}}\PYG{o}{\PYGZgt{}}\PYG{n}{getCon
↪}
  \PYG{c+c1}{//entered config mode, make led toggle faster}
  \PYG{n}{ticker}\PYG{p}{.}\PYG{n}{attach}\PYG{p}{(}\PYG{l+m+mf}{2}\PYG{p}{,}␣
↪\PYG{n}{tick}\PYG{p}{)}\PYG{p}{;}
\PYG{p}{\PYGZcb{}}



\PYG{k+kt}{void} \PYG{n+nf}{setup}\PYG{p}{(}\PYG{p}{)} \PYG{p}{\PYGZob{}}
```

```
  \PYG{c+c1}{// put your setup code here, to run once:}
  \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{begin}\PYG{p}{(}\PYG{l+m+mi}{1152}\PYG{p}{)}\PYG{p}{;}

  \PYG{c+c1}{//set led pin as output}
  \PYG{n}{pinMode}\PYG{p}{(}\PYG{n}{BUILTIN\PYGZus{}LED}\PYG{p}{,}␣
↪\PYG{n}{OUTPUT}\PYG{p}{)}\PYG{p}{;}

  \PYG{c+c1}{// start ticker with 5̇ because we start in AP mode and try to connect}
  \PYG{n}{ticker}\PYG{p}{.}\PYG{n}{attach}\PYG{p}{(}\PYG{l+m+mf}{6̇}\PYG{p}{,}␣
↪\PYG{n}{tick}\PYG{p}{)}\PYG{p}{;}

  \PYG{c+c1}{//WiFiManager}
  \PYG{c+c1}{//Local intialization. Once its business is done, there is no need to␣
↪keep it around}
  \PYG{n}{WiFiManager} \PYG{n}{wifiManager}\PYG{p}{;}
  \PYG{c+c1}{//reset settings \PYGZhy{} for testing}
  \PYG{c+c1}{//wifiManager.resetSettings();}

  \PYG{c+c1}{//set callback that gets called when connecting to previous WiFi fails,␣
↪and enters Access Point mode}
  \PYG{n}{wifiManager}\PYG{p}{.
↪}\PYG{n}{setAPCallback}\PYG{p}{(}\PYG{n}{configModeCallback}\PYG{p}{)}\PYG{p}{;}

  \PYG{c+c1}{//fetches ssid and pass and tries to connect}
  \PYG{c+c1}{//if it does not connect it starts an access point with the specified␣
↪name}
  \PYG{c+c1}{//here  \PYGZdq{}AutoConnectAP\PYGZdq{}}
  \PYG{c+c1}{//and goes into a blocking loop awaiting configuration}
  \PYG{k}{if} \PYG{p}{(}\PYG{o}{!}\PYG{n}{wifiManager}\PYG{p}{.
↪}\PYG{n}{autoConnect}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{MeseIoT}\PYG{l+s}{\PYGZdq{}}\PYG{p}{,
↪} \PYG{l+s}{\PYGZdq{}}\PYG{l+s}{MeseIoT**}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{)}␣
↪\PYG{p}{\PYGZob{}}
    \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{println}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{failed to connect and hit␣
↪timeout}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
    \PYG{c+c1}{//reset and try again, or maybe put it to deep sleep}
    \PYG{n}{ESP}\PYG{p}{.}\PYG{n}{reset}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
    \PYG{n}{delay}\PYG{p}{(}\PYG{l+m+mi}{1}\PYG{p}{)}\PYG{p}{;}
  \PYG{p}{\PYGZcb{}}

  \PYG{c+c1}{//if you get here you have connected to the WiFi}
  \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{println}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{connected...yeey␣
↪:)}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
  \PYG{n}{ticker}\PYG{p}{.}\PYG{n}{detach}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
  \PYG{c+c1}{//keep LED on}
  \PYG{n}{digitalWrite}\PYG{p}{(}\PYG{n}{BUILTIN\PYGZus{}LED}\PYG{p}{,}␣
↪\PYG{n}{LOW}\PYG{p}{)}\PYG{p}{;}

  \PYG{n}{dht}\PYG{p}{.}\PYG{n}{begin}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
\PYG{p}{\PYGZcb{}}



\PYG{k+kt}{void} \PYG{n+nf}{loop}\PYG{p}{(}\PYG{p}{)} \PYG{p}{\PYGZob{}}
```

```
 \PYG{c+c1}{// Wait a few seconds between measurements.}
 \PYG{n}{delay}\PYG{p}{(}\PYG{l+m+mi}{2}\PYG{p}{)}\PYG{p}{;}

 \PYG{c+c1}{// Reading temperature or humidity takes about 25 milliseconds!}
 \PYG{c+c1}{// Sensor readings may also be up to 2 seconds \PYGZsq{}old\PYGZsq{}␣
↪(its a very slow sensor)}
 \PYG{k+kt}{float} \PYG{n}{h} \PYG{o}{=} \PYG{n}{dht}\PYG{p}{.
↪}\PYG{n}{readHumidity}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
 \PYG{c+c1}{// Read temperature as Celsius (the default)}
 \PYG{k+kt}{float} \PYG{n}{t} \PYG{o}{=} \PYG{n}{dht}\PYG{p}{.
↪}\PYG{n}{readTemperature}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
 \PYG{c+c1}{// Read temperature as Fahrenheit (isFahrenheit = true)}
 \PYG{k+kt}{float} \PYG{n}{f} \PYG{o}{=} \PYG{n}{dht}\PYG{p}{.
↪}\PYG{n}{readTemperature}\PYG{p}{(}\PYG{n+nb}{true}\PYG{p}{)}\PYG{p}{;}

 \PYG{c+c1}{// Check if any reads failed and exit early (to try again).}
 \PYG{k}{if} \PYG{p}{(}\PYG{n}{isnan}\PYG{p}{(}\PYG{n}{h}\PYG{p}{)}␣
↪\PYG{o}{\textbar{}}\PYG{o}{\textbar{}} \PYG{n}{isnan}\PYG{p}{(}\PYG{n}{t}\PYG{p}{)}␣
↪\PYG{o}{\textbar{}}\PYG{o}{\textbar{}}␣
↪\PYG{n}{isnan}\PYG{p}{(}\PYG{n}{f}\PYG{p}{)}\PYG{p}{)} \PYG{p}{\PYGZob{}}
   \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{println}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{Failed to read from DHT␣
↪sensor!}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
   \PYG{k}{return}\PYG{p}{;}
 \PYG{p}{\PYGZcb{}}

 \PYG{c+c1}{// Compute heat index in Fahrenheit (the default)}
 \PYG{k+kt}{float} \PYG{n}{hif} \PYG{o}{=} \PYG{n}{dht}\PYG{p}{.
↪}\PYG{n}{computeHeatIndex}\PYG{p}{(}\PYG{n}{f}\PYG{p}{,}␣
↪\PYG{n}{h}\PYG{p}{)}\PYG{p}{;}
 \PYG{c+c1}{// Compute heat index in Celsius (isFahreheit = false)}
 \PYG{k+kt}{float} \PYG{n}{hic} \PYG{o}{=} \PYG{n}{dht}\PYG{p}{.
↪}\PYG{n}{computeHeatIndex}\PYG{p}{(}\PYG{n}{t}\PYG{p}{,} \PYG{n}{h}\PYG{p}{,}␣
↪\PYG{n+nb}{false}\PYG{p}{)}\PYG{p}{;}

 \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{Humidity:␣
↪}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{n}{h}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{␣
↪\PYGZpc{}}\PYG{l+s+se}{\PYGZbs{}t}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{Temperature:␣
↪}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{n}{t}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{ *C␣
↪}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{n}{f}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{␣
↪*F}\PYG{l+s+se}{\PYGZbs{}t}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{Heat␣
↪index: }\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{n}{hic}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{ *C␣
↪}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
 \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{print}\PYG{p}{(}\PYG{n}{hif}\PYG{p}{)}\PYG{p}{;}
```

```
  \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{println}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{␣
↪*F}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}

  \PYG{c+c1}{///}
  \PYG{n}{HTTPClient} \PYG{n}{http}\PYG{p}{;}

  \PYG{c+c1}{// configure server and url}
  \PYG{n}{http}\PYG{p}{.}\PYG{n}{begin}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{http://
↪iothook.com/api/latest/datas/update/?api\PYGZus{}key=5c75f\PYGZhy{}9c4
PYGZhy{}11e144d3e\PYGZam{}value\PYGZus{}1=}\PYG{l+s}{\PYGZdq{}} \PYG{o}{+}␣
↪\PYG{n}{String}\PYG{p}{(}\PYG{n}{t}\PYG{p}{)} \PYG{o}{+}␣
↪\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{\PYGZam{}value\PYGZus{}2=}\PYG{l+s}{\PYGZdq{}}␣
↪\PYG{o}{+} \PYG{n}{String}\PYG{p}{(}\PYG{n}{h}\PYG{p}{)} \PYG{o}{+}␣
↪\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
  \PYG{c+c1}{//http.begin(\PYGZdq{}192.168.1.12\PYGZdq{}, 8 \PYGZdq{}/test.
↪html\PYGZdq{});}

  \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{[HTTP] GET...
↪}\PYG{l+s+se}{\PYGZbs{}n}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}
  \PYG{c+c1}{// start connection and send HTTP header}
  \PYG{k+kt}{int} \PYG{n}{httpCode} \PYG{o}{=} \PYG{n}{http}\PYG{p}{.
↪}\PYG{n}{GET}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
  \PYG{k}{if} \PYG{p}{(}\PYG{n}{httpCode} \PYG{o}{\PYGZgt{}} \PYG{l+m+mi}{}\PYG{p}{)}␣
↪\PYG{p}{\PYGZob{}}
    \PYG{c+c1}{// HTTP header has been send and Server response header has been␣
↪handled}
    \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{printf}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{[HTTP] GET... code:␣
↪\PYGZpc{}d}\PYG{l+s+se}{\PYGZbs{}n}\PYG{l+s}{\PYGZdq{}}\PYG{p}{,}␣
↪\PYG{n}{httpCode}\PYG{p}{)}\PYG{p}{;}

    \PYG{c+c1}{// file found at server}
    \PYG{k}{if} \PYG{p}{(}\PYG{n}{httpCode} \PYG{o}{=}\PYG{o}{=}␣
↪\PYG{n}{HTTP\PYGZus{}CODE\PYGZus{}OK}\PYG{p}{)} \PYG{p}{\PYGZob{}}

      \PYG{c+c1}{// get lenght of document (is \PYGZhy{}1 when Server sends no␣
↪Content\PYGZhy{}Length header)}
      \PYG{k+kt}{int} \PYG{n}{len} \PYG{o}{=} \PYG{n}{http}\PYG{p}{.
↪}\PYG{n}{getSize}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}

      \PYG{c+c1}{// create buffer for read}
      \PYG{k+kt}{uint8\PYGZus{}t} \PYG{n}{buff}\PYG{p}{[}\PYG{l+m+mi}{128}\PYG{p}{]}␣
↪\PYG{o}{=} \PYG{p}{\PYGZob{}} \PYG{l+m+mi}{} \PYG{p}{\PYGZcb{}}\PYG{p}{;}

      \PYG{c+c1}{// get tcp stream}
      \PYG{n}{WiFiClient} \PYG{o}{*} \PYG{n}{stream} \PYG{o}{=} \PYG{n}{http}\PYG{p}{.
↪}\PYG{n}{getStreamPtr}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}

      \PYG{c+c1}{// read all data from server}
      \PYG{k}{while} \PYG{p}{(}\PYG{n}{http}\PYG{p}{.
↪}\PYG{n}{connected}\PYG{p}{(}\PYG{p}{)} \PYG{o}{\PYGZam{}}\PYG{o}{\PYGZam{}}␣
↪\PYG{p}{(}\PYG{n}{len} \PYG{o}{\PYGZgt{}} \PYG{l+m+mi}{}␣
↪\PYG{o}{\textbar{}}\PYG{o}{\textbar{}} \PYG{n}{len} \PYG{o}{=}\PYG{o}{=}␣
↪\PYG{o}{\PYGZhy{}}\PYG{l+m+mi}{1}\PYG{p}{)}\PYG{p}{)} \PYG{p}{\PYGZob{}}
        \PYG{c+c1}{// get available data size}
    \PYG{k+kt}{size\PYGZus{}t} \PYG{n}{size} \PYG{o}{=}␣
↪\PYG{n}{stream}\PYG{o}{\PYGZhy{}}\PYG{o}{\PYGZgt{}}\PYG{n}{available}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;
↪}

    \PYG{k}{if} \PYG{p}{(}\PYG{n}{size}\PYG{p}{)} \PYG{p}{\PYGZob{}}
     \PYG{c+c1}{// read up to 128 byte}
     \PYG{k+kt}{int} \PYG{n}{c} \PYG{o}{=}␣
↪\PYG{n}{stream}\PYG{o}{\PYGZhy{}}\PYG{o}{\PYGZgt{}}\PYG{n}{readBytes}\PYG{p}{(}\PYG{n}{buff}\PYG{p
↪} \PYG{p}{(}\PYG{p}{(}\PYG{n}{size} \PYG{o}{\PYGZgt{}}␣
↪\PYG{k}{sizeof}\PYG{p}{(}\PYG{n}{buff}\PYG{p}{)}\PYG{p}{)} \PYG{o}{?}␣
↪\PYG{k}{sizeof}\PYG{p}{(}\PYG{n}{buff}\PYG{p}{)} \PYG{o}{:}␣
↪\PYG{n}{size}\PYG{p}{)}\PYG{p}{)}\PYG{p}{;}

     \PYG{c+c1}{// write it to Serial}
     \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{write}\PYG{p}{(}\PYG{n}{buff}\PYG{p}{,}␣
↪\PYG{n}{c}\PYG{p}{)}\PYG{p}{;}

     \PYG{k}{if} \PYG{p}{(}\PYG{n}{len} \PYG{o}{\PYGZgt{}}␣
↪\PYG{l+m+mi}{}\PYG{p}{)} \PYG{p}{\PYGZob{}}
      \PYG{n}{len} \PYG{o}{\PYGZhy{}}\PYG{o}{=} \PYG{n}{c}\PYG{p}{;}
     \PYG{p}{\PYGZcb{}}
    \PYG{p}{\PYGZcb{}}
    \PYG{n}{delay}\PYG{p}{(}\PYG{l+m+mi}{1}\PYG{p}{)}\PYG{p}{;}
   \PYG{p}{\PYGZcb{}}

   \PYG{n}{Serial}\PYG{p}{.}\PYG{n}{println}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
   \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{print}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{[HTTP] connection closed or␣
↪file end.}\PYG{l+s+se}{\PYGZbs{}n}\PYG{l+s}{\PYGZdq{}}\PYG{p}{)}\PYG{p}{;}

  \PYG{p}{\PYGZcb{}}
 \PYG{p}{\PYGZcb{}} \PYG{k}{else} \PYG{p}{\PYGZob{}}
  \PYG{n}{Serial}\PYG{p}{.
↪}\PYG{n}{printf}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}}\PYG{l+s}{[HTTP] GET... failed,␣
↪error: \PYGZpc{}s}\PYG{l+s+se}{\PYGZbs{}n}\PYG{l+s}{\PYGZdq{}}\PYG{p}{,}␣
↪\PYG{n}{http}\PYG{p}{.
↪}\PYG{n}{errorToString}\PYG{p}{(}\PYG{n}{httpCode}\PYG{p}{)}\PYG{p}{.
↪}\PYG{n}{c\PYGZus{}str}\PYG{p}{(}\PYG{p}{)}\PYG{p}{)}\PYG{p}{;}
 \PYG{p}{\PYGZcb{}}

 \PYG{n}{http}\PYG{p}{.}\PYG{n}{end}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
 \PYG{c+c1}{////}
 \PYG{n}{delay}\PYG{p}{(}\PYG{l+m+mi}{13}\PYG{p}{)}\PYG{p}{;}
\PYG{p}{\PYGZcb{}}
```

Veri Al

Iot cihazından gönderilen ısı, nem, voltaj, ışık gibi değerleri iothook data merkezinden çekebilmek için öncelikle kanal üye kullanıcı adı ve giriş şifresineihtiyaç vardır. Iot cihazından gelen veriler, Android, iOS gibi mobil cihazınızdan veya web sitenizden izlenebilir. Kanalınızın kullanımı genel kullanıma açık ise diğer kullanıcılar ile de bu verileri paylaşabilirsiniz.

Örneğin; Kanal adımız "Temperature sensor" olarak belirlenmiş ve kanal içerisinde bulunacak iot cihazımızdan "temperature" ve "humidity" element verileri gönderiliyor olsun.

Python 2, Python 3 Json ile Veri Alma

Python Json ile Get Örneği:

Bu örneği http://bit.ly/2jI1FNQ sayfasından inceleyebilirsiniz.

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{\PYGZsh{} \PYGZhy{}*\PYGZhy{} coding: utf\PYGZhy{}8 \PYGZhy{}*\PYGZhy{}}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}
\PYG{l+s+sd}{  Python 2, Python 3 ile IoThook REST Api Testi}

\PYG{l+s+sd}{  Kod çalıştırıldığında \PYGZsq{}data\PYGZsq{} değişkenine verilen␣
↪\PYGZsq{}all\PYGZsq{} değişkeni ile}
\PYG{l+s+sd}{  auth sahipliğindeki tüm veriler alınır.}

\PYG{l+s+sd}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{l+s+sd}{  testlerin yapilmasini amaclamaktadir.}

\PYG{l+s+sd}{  1 Mayıs 27}
\PYG{l+s+sd}{  Sahin MERSIN}

\PYG{l+s+sd}{  Daha fazlasi icin}

\PYG{l+s+sd}{  http://www.iothook.com}
\PYG{l+s+sd}{  ve}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook}

\PYG{l+s+sd}{  sitelerine gidiniz.}

\PYG{l+s+sd}{  Sorular ve destek talepleri icin}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook/issues}
\PYG{l+s+sd}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}

\PYG{l+s+sd}{  Yayin : http://mesebilisim.com}

\PYG{l+s+sd}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{l+s+sd}{  You may not use this file except in compliance with the License.}
\PYG{l+s+sd}{  A copy of the License is located at}

\PYG{l+s+sd}{  http://www.apache.org/licenses/}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}

\PYG{k+kn}{import} \PYG{n+nn}{requests}

\PYG{n}{API\PYGZus{}KEY} \PYG{o}{=}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{511b73\PYGZhy{}95a5\PYGZhy{}11c814c2297e434c}\PYG{l+s+s1}{\PYGZ

\PYG{n}{url} \PYG{o}{=} \PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{https://iothook.com/api/
↪latest/datas/?api\PYGZus{}key=}\PYG{l+s+s1}{\PYGZsq{}} \PYG{o}{+}␣
↪\PYG{n}{API\PYGZus{}KEY}
```

```
\PYG{n}{response} \PYG{o}{=} \PYG{n}{requests}\PYG{o}{.
↪}\PYG{n}{get}\PYG{p}{(}\PYG{n}{url}\PYG{p}{)}
\PYG{n}{data} \PYG{o}{=} \PYG{n}{response}\PYG{o}{.}\PYG{n}{json}\PYG{p}{(}\PYG{p}{)}
\PYG{k}{print} \PYG{n}{data}
```

Python 2, Python 3 Json ile İlk Veriyi Alma

Python İlk Veriyi Alma, Json ile Get Örneği:

Bu örneği http://bit.ly/2jI1FNQ sayfasından inceleyebilirsiniz.

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{\PYGZsh{} \PYGZhy{}*\PYGZhy{} coding: utf\PYGZhy{}8 \PYGZhy{}*\PYGZhy{}}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}
\PYG{l+s+sd}{  Python 2 ile IoThook REST Api Testi}

\PYG{l+s+sd}{  Kod çalıştırıldığında \PYGZsq{}data\PYGZsq{} değişkenine verilen␣
↪\PYGZsq{}first\PYGZsq{} değişkeni ile}
\PYG{l+s+sd}{  auth sahipliğindeki ilk veri alınır. \PYGZsq{}channel\PYGZsq{}␣
↪değişkeni Iothook dashboard}
\PYG{l+s+sd}{  Kanal oluşturma sırasında otomatik verilen id numarasıdır.}

\PYG{l+s+sd}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{l+s+sd}{  testlerin yapilmasini amaclamaktadir.}

\PYG{l+s+sd}{  1 Mayıs 27}
\PYG{l+s+sd}{  Sahin MERSIN}

\PYG{l+s+sd}{  Daha fazlasi icin}

\PYG{l+s+sd}{  http://www.iothook.com}
\PYG{l+s+sd}{  ve}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook}

\PYG{l+s+sd}{  sitelerine gidiniz.}

\PYG{l+s+sd}{  Sorular ve destek talepleri icin}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook/issues}
\PYG{l+s+sd}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}

\PYG{l+s+sd}{  Yayin : http://mesebilisim.com}

\PYG{l+s+sd}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{l+s+sd}{  You may not use this file except in compliance with the License.}
\PYG{l+s+sd}{  A copy of the License is located at}

\PYG{l+s+sd}{  http://www.apache.org/licenses/}
```

```
\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}

\PYG{k+kn}{import} \PYG{n+nn}{requests}

\PYG{n}{API\PYGZus{}KEY} \PYG{o}{=}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{511b73\PYGZhy{}95a5\PYGZhy{}11c814c2297e434c}\PYG{l+s+s1}{\PYGZ

\PYG{n}{url} \PYG{o}{=} \PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{https://iothook.com/api/
↪latest/datas/?data=first\PYGZam{}api\PYGZus{}key=}\PYG{l+s+s1}{\PYGZsq{}}␣
↪\PYG{o}{+} \PYG{n}{API\PYGZus{}KEY}

\PYG{n}{response} \PYG{o}{=} \PYG{n}{requests}\PYG{o}{.
↪}\PYG{n}{get}\PYG{p}{(}\PYG{n}{url}\PYG{p}{)}
\PYG{n}{data} \PYG{o}{=} \PYG{n}{response}\PYG{o}{.}\PYG{n}{json}\PYG{p}{(}\PYG{p}{)}
\PYG{k}{print} \PYG{n}{data}
```

Python 2, Python 3 Json ile Son Veriyi Alma

Python Son Veriyi Alma, Json ile Get Örneği:

Bu örneği http://bit.ly/2jI1FNQ sayfasından inceleyebilirsiniz.

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{\PYGZsh{} \PYGZhy{}*\PYGZhy{} coding: utf\PYGZhy{}8 \PYGZhy{}*\PYGZhy{}}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}
\PYG{l+s+sd}{  Python 2 ile IoThook REST Api Testi}

\PYG{l+s+sd}{  Kod çalıştırıldığında \PYGZsq{}data\PYGZsq{} değişkenine verilen␣
↪\PYGZsq{}last\PYGZsq{} değişkeni ile}
\PYG{l+s+sd}{  auth sahipliğindeki en son veri alınır. \PYGZsq{}channel\PYGZsq{}␣
↪değişkeni Iothook dashboard}
\PYG{l+s+sd}{  Kanal oluşturma sırasında otomatik verilen id numarasıdır.}

\PYG{l+s+sd}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{l+s+sd}{  testlerin yapilmasini amaclamaktadir.}

\PYG{l+s+sd}{  1 Mayıs 27}
\PYG{l+s+sd}{  Sahin MERSIN}

\PYG{l+s+sd}{  Daha fazlasi icin}

\PYG{l+s+sd}{  http://www.iothook.com}
\PYG{l+s+sd}{  ve}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook}

\PYG{l+s+sd}{  sitelerine gidiniz.}

\PYG{l+s+sd}{  Sorular ve destek talepleri icin}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook/issues}
\PYG{l+s+sd}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}
```

```
\PYG{l+s+sd}{  Yayin : http://mesebilisim.com}

\PYG{l+s+sd}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{l+s+sd}{  You may not use this file except in compliance with the License.}
\PYG{l+s+sd}{  A copy of the License is located at}

\PYG{l+s+sd}{  http://www.apache.org/licenses/}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}

\PYG{k+kn}{import} \PYG{n+nn}{requests}


\PYG{n}{API\PYGZus{}KEY} \PYG{o}{=}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{516b73\PYGZhy{}95a5\PYGZhy{}11c814c2597e434c}\PYG{l+s+s1}{\PYGZ

\PYG{n}{url} \PYG{o}{=} \PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{https://iothook.com/api/
↪latest/datas/?data=last\PYGZam{}api\PYGZus{}key=}\PYG{l+s+s1}{\PYGZsq{}} \PYG{o}{+}␣
↪\PYG{n}{API\PYGZus{}KEY}

\PYG{n}{response} \PYG{o}{=} \PYG{n}{requests}\PYG{o}{.
↪}\PYG{n}{get}\PYG{p}{(}\PYG{n}{url}\PYG{p}{)}
\PYG{n}{data} \PYG{o}{=} \PYG{n}{response}\PYG{o}{.}\PYG{n}{json}\PYG{p}{(}\PYG{p}{)}
\PYG{k}{print} \PYG{n}{data}
```

Python 2, Python 3 Json ile Veriye Ait Detay Alma

Python veriye ait detay alma örneği:

Bu örneği http://bit.ly/2jI1FNQ sayfasından inceleyebilirsiniz.

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+c1}{\PYGZsh{} \PYGZhy{}*\PYGZhy{} coding: utf\PYGZhy{}8 \PYGZhy{}*\PYGZhy{}}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}
\PYG{l+s+sd}{  Python 2, 3 ile IoThook REST Api Testi}

\PYG{l+s+sd}{  Kod çalıştırıldığında datas url yapısına parametre olarak verilen}
\PYG{l+s+sd}{  değer Kanal ve Element içerisinde tanımlı datanın ayrıntılarını␣
↪getirir.}

\PYG{l+s+sd}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{l+s+sd}{  testlerin yapilmasini amaclamaktadir.}

\PYG{l+s+sd}{  1 Mayıs 27}
\PYG{l+s+sd}{  Sahin MERSIN}

\PYG{l+s+sd}{  Daha fazlasi icin}
```

```
\PYG{l+s+sd}{  http://www.iothook.com}
\PYG{l+s+sd}{  ve}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook}

\PYG{l+s+sd}{  sitelerine gidiniz.}

\PYG{l+s+sd}{  Sorular ve destek talepleri icin}
\PYG{l+s+sd}{  https://github.com/electrocoder/iotHook/issues}
\PYG{l+s+sd}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}

\PYG{l+s+sd}{  Yayin : http://mesebilisim.com}

\PYG{l+s+sd}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{l+s+sd}{  You may not use this file except in compliance with the License.}
\PYG{l+s+sd}{  A copy of the License is located at}

\PYG{l+s+sd}{  http://www.apache.org/licenses/}

\PYG{l+s+sd}{\PYGZdq{}\PYGZdq{}\PYGZdq{}}

\PYG{k+kn}{import} \PYG{n+nn}{requests}

\PYG{n}{API\PYGZus{}KEY} \PYG{o}{=}␣
↪\PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{516b73\PYGZhy{}95a5\PYGZhy{}11c814c2297e434c}\PYG{l+s+s1}{\PYGZ

\PYG{n}{url} \PYG{o}{=} \PYG{l+s+s1}{\PYGZsq{}}\PYG{l+s+s1}{https://iothook.com/api/
↪latest/datas/4545/?api\PYGZus{}key=}\PYG{l+s+s1}{\PYGZsq{}} \PYG{o}{+}␣
↪\PYG{n}{API\PYGZus{}KEY}

\PYG{n}{response} \PYG{o}{=} \PYG{n}{requests}\PYG{o}{.
↪}\PYG{n}{get}\PYG{p}{(}\PYG{n}{url}\PYG{p}{)}
\PYG{n}{data} \PYG{o}{=} \PYG{n}{response}\PYG{o}{.}\PYG{n}{json}\PYG{p}{(}\PYG{p}{)}
\PYG{k}{print} \PYG{n}{data}
```

C# ile Json Formatında Veri Alma

IoThook kullanıcısının tüm kanallarına ait veriyi alabilmesi için GET metodu ile '?data=last' değişkeninin gönderilmesi gereklidir. Aynı örnekdeki data değişkeninin alabileceği değerler:

- ?data=all : Kullanıcının tüm datalarını getir
- ?data=first : Kullanıcının ilk datasını getir
- ?data=last : Kullanıcının son datasını getir

CSharp örneğinde *HttpWebRequest* metodu kullanılmıştır.

Bu örneği http://bit.ly/2jI1FNQ Github sayfasından inceleyebilirsiniz.

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{c+cm}{/*}
\PYG{c+cm}{  C\PYGZsh{} ile IoThook REST Api Testi}
```

```
\PYG{c+cm}{  Bu örnek ile CSharp ve Request metodu ile kullanıcının datalarının get␣
↪metodu ile alınması gerçekleştirilmiştir.}
\PYG{c+cm}{  \PYGZsq{}autorization\PYGZsq{} ile kullanıcı adı ve parola değeri␣
↪verilmelidir.}

\PYG{c+cm}{  Bu ornek IotHook servisine veri almak/gondermek icin baslangic␣
↪seviyesinde}
\PYG{c+cm}{  testlerin yapilmasini amaclamaktadir.}

\PYG{c+cm}{  29 Temmuz 27}
\PYG{c+cm}{  Sahin MERSIN}

\PYG{c+cm}{  Daha fazlasi icin}

\PYG{c+cm}{  http://www.iothook.com}
\PYG{c+cm}{  ve}
\PYG{c+cm}{  https://github.com/electrocoder/iotHook}

\PYG{c+cm}{  sitelerine gidiniz.}

\PYG{c+cm}{  Sorular ve destek talepleri icin}
\PYG{c+cm}{  https://github.com/electrocoder/iotHook/issues}
\PYG{c+cm}{  sayfasindan veya Meşe Bilişim den yardım alabilirsiniz.}

\PYG{c+cm}{  Yayin : http://mesebilisim.com}

\PYG{c+cm}{  Licensed under the Apache License, Version 2. (the␣
↪\PYGZdq{}License\PYGZdq{}).}
\PYG{c+cm}{  You may not use this file except in compliance with the License.}
\PYG{c+cm}{  A copy of the License is located at}

\PYG{c+cm}{  http://www.apache.org/licenses/}

\PYG{c+cm}{*/}

\PYG{k}{using} \PYG{n+nn}{System}\PYG{p}{;}
\PYG{k}{using} \PYG{n+nn}{System.IO}\PYG{p}{;}
\PYG{k}{using} \PYG{n+nn}{System.Net}\PYG{p}{;}

\PYG{k}{namespace} \PYG{n+nn}{Iothook}
\PYG{p}{\PYGZob{}}
    \PYG{k}{class} \PYG{n+nc}{Program}
    \PYG{p}{\PYGZob{}}
        \PYG{k}{static} \PYG{k}{void}␣
↪\PYG{n+nf}{Main}\PYG{p}{(}\PYG{k+kt}{string}\PYG{p}{[}\PYG{p}{]}␣
↪\PYG{n}{args}\PYG{p}{)}
        \PYG{p}{\PYGZob{}}
            \PYG{k+kt}{string} \PYG{n}{url} \PYG{p}{=}␣
↪\PYG{l+s}{\PYGZdq{}\PYGZdq{}}\PYG{p}{;}
            \PYG{n}{url} \PYG{p}{=} \PYG{l+s}{\PYGZdq{}https://iothook.com/api/v1.2/
↪datas/?data=all\PYGZdq{}}\PYG{p}{;} \PYG{c+c1}{// for all data}

            \PYG{k+kt}{var} \PYG{n}{webRequest} \PYG{p}{=}␣
↪\PYG{p}{(}\PYG{n}{HttpWebRequest}\PYG{p}{)}\PYG{n}{WebRequest}\PYG{p}{.
↪}\PYG{n}{Create}\PYG{p}{(}\PYG{n}{url}\PYG{p}{)}\PYG{p}{;}
```

```
            \PYG{n}{webRequest}\PYG{p}{.}\PYG{n}{Method} \PYG{p}{=}␣
↪\PYG{l+s}{\PYGZdq{}GET\PYGZdq{}}\PYG{p}{;}
            \PYG{n}{webRequest}\PYG{p}{.}\PYG{n}{ContentType} \PYG{p}{=}␣
↪\PYG{l+s}{\PYGZdq{}application/json\PYGZdq{}}\PYG{p}{;}
            \PYG{n}{webRequest}\PYG{p}{.}\PYG{n}{UserAgent} \PYG{p}{=}␣
↪\PYG{l+s}{\PYGZdq{}Mozilla/5. (Windows NT 5.1; rv:28. Gecko/21 Firefox/28.
PYGZdq{}}\PYG{p}{;}
            \PYG{n}{webRequest}\PYG{p}{.}\PYG{n}{ContentLength} \PYG{p}{=}␣
↪\PYG{l+m}{}\PYG{p}{;}

            \PYG{k+kt}{string} \PYG{n}{autorization} \PYG{p}{=}␣
↪\PYG{l+s}{\PYGZdq{}USERNAME\PYGZdq{}} \PYG{p}{+} \PYG{l+s}{\PYGZdq{}:\PYGZdq{}}␣
↪\PYG{p}{+} \PYG{l+s}{\PYGZdq{}PASSWORD\PYGZdq{}}\PYG{p}{;}
            \PYG{k+kt}{byte}\PYG{p}{[}\PYG{p}{]} \PYG{n}{binaryAuthorization}␣
↪\PYG{p}{=} \PYG{n}{System}\PYG{p}{.}\PYG{n}{Text}\PYG{p}{.
↪}\PYG{n}{Encoding}\PYG{p}{.}\PYG{n}{UTF8}\PYG{p}{.
↪}\PYG{n}{GetBytes}\PYG{p}{(}\PYG{n}{autorization}\PYG{p}{)}\PYG{p}{;}
            \PYG{n}{autorization} \PYG{p}{=} \PYG{n}{Convert}\PYG{p}{.
↪}\PYG{n}{ToBase64String}\PYG{p}{(}\PYG{n}{binaryAuthorization}\PYG{p}{)}\PYG{p}{;}
            \PYG{n}{autorization} \PYG{p}{=} \PYG{l+s}{\PYGZdq{}Basic \PYGZdq{}}␣
↪\PYG{p}{+} \PYG{n}{autorization}\PYG{p}{;}
            \PYG{n}{webRequest}\PYG{p}{.}\PYG{n}{Headers}\PYG{p}{.
↪}\PYG{n}{Add}\PYG{p}{(}\PYG{l+s}{\PYGZdq{}AUTHORIZATION\PYGZdq{}}\PYG{p}{,}␣
↪\PYG{n}{autorization}\PYG{p}{)}\PYG{p}{;}

            \PYG{k+kt}{var} \PYG{n}{webResponse} \PYG{p}{=}␣
↪\PYG{p}{(}\PYG{n}{HttpWebResponse}\PYG{p}{)}\PYG{n}{webRequest}\PYG{p}{.
↪}\PYG{n}{GetResponse}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}

            \PYG{k}{if} \PYG{p}{(}\PYG{n}{webResponse}\PYG{p}{.}\PYG{n}{StatusCode}␣
↪\PYG{p}{!}\PYG{p}{=} \PYG{n}{HttpStatusCode}\PYG{p}{.}\PYG{n}{OK}\PYG{p}{)}
                \PYG{n}{Console}\PYG{p}{.
↪}\PYG{n}{WriteLine}\PYG{p}{(}\PYG{n}{webResponse}\PYG{p}{.}\PYG{n}{Headers}\PYG{p}{.
↪}\PYG{n}{ToString}\PYG{p}{(}\PYG{p}{)}\PYG{p}{)}\PYG{p}{;}

            \PYG{k}{using} \PYG{p}{(}\PYG{n}{StreamReader} \PYG{n}{reader} \PYG{p}{=}␣
↪\PYG{k}{new} \PYG{n}{StreamReader}\PYG{p}{(}\PYG{n}{webResponse}\PYG{p}{.
↪}\PYG{n}{GetResponseStream}\PYG{p}{(}\PYG{p}{)}\PYG{p}{)}\PYG{p}{)}
            \PYG{p}{\PYGZob{}}
                \PYG{n}{Console}\PYG{p}{.
↪}\PYG{n}{WriteLine}\PYG{p}{(}\PYG{n}{reader}\PYG{p}{.
↪}\PYG{n}{ReadToEnd}\PYG{p}{(}\PYG{p}{)}\PYG{p}{)}\PYG{p}{;}
                \PYG{n}{reader}\PYG{p}{.}\PYG{n}{Close}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
                \PYG{n}{webRequest}\PYG{p}{.
↪}\PYG{n}{Abort}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
            \PYG{p}{\PYGZcb{}}

            \PYG{n}{Console}\PYG{p}{.}\PYG{n}{ReadLine}\PYG{p}{(}\PYG{p}{)}\PYG{p}{;}
        \PYG{p}{\PYGZcb{}}
    \PYG{p}{\PYGZcb{}}
\PYG{p}{\PYGZcb{}}
```

Email Besleme

Iot cihazlardan email alma

Iot cihazlardan email almak için 'Kanal Ekle' menüsünden kanal oluşturulurken 'Email feed' seçeneğinin aktif edilmesi gerekir. Kanal oluşturulduktan sonrada email alma seçeneği değiştirilebilir. Güncelleme için 'Kanal Liste' menüsünden 'Düzenle' seçeneği altından yapılabilir.

Email besleme planı

Iot cihazınızdan veri geldiğinde email ile besleme almak için 'STUDENT', 'PRO', veya 'ULTRA' planlardan birisini tercih etmelisiniz.

Plan değişikliği için Ödeme sayfasından size uygun planı seçerek email besleme alabilirsiniz.

Email besleme süresi

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

Iot Email Sms Alarm

Alarm nedir?

Iot Kanal altında oluşturulan Elementlere alarm değeri kurma işlemidir. Alarm değeri kurularak iot cihazıdan her veri alındığında operatör ile işlem yapılarak sonuca göre alarm üretilir. Üretilen alarm abonelik tipine göre bir günde en fazla atılabilecek email ve sms planına göre belirlenir.

Operatörler

İşlem operatörleri aşağıdaki gibidir:

- < : Küçüktür operatörü. a < b. gelen_deger < alarm_degeri. Iot cihazdan gönderilen değer ile alarm değerini karşılaştırır. İşlem sonucu doğru (True) ise alarm üretilir.
- <= : Küçük eşittir operatörü. a <= b. gelen_deger <= alarm_degeri. Iot cihazdan gönderilen değer ile alarm değerini karşılaştırır. İşlem sonucu doğru (True) ise alarm üretilir.
- == : Eşittir operatörü. a== b. gelen_deger== alarm_degeri. Iot cihazdan gönderilen değer ile alarm değerini karşılaştırır. İşlem sonucu doğru (True) ise alarm üretilir.
- != : Eşit değil operatörü. a != b. gelen_deger != alarm_degeri. Iot cihazdan gönderilen değer ile alarm değerini karşılaştırır. İşlem sonucu doğru (True) ise alarm üretilir.
- >= : Büyük eşit operatörü. a >= b. gelen_deger >= alarm_degeri. Iot cihazdan gönderilen değer ile alarm değerini karşılaştırır. İşlem sonucu doğru (True) ise alarm üretilir.

- > : Büyüktür operatörü. a > b. gelen_deger > alarm_degeri. Iot cihazdan gönderilen değer ile alarm değerini karşılaştırır. İşlem sonucu doğru (True) ise alarm üretilir.

Örnek operatör işlemleri:

- < Küçüktür operatörü python örnek:

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{5}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZlt{}} \PYG{n}{b}
\PYG{n+nb+bp}{True}
```

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{9}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZlt{}} \PYG{n}{b}
\PYG{n+nb+bp}{False}
```

- <= Küçük eşittir operatörü python örnek :

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{5}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZlt{}}\PYG{o}{=} \PYG{n}{b}
\PYG{n+nb+bp}{True}
```

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZlt{}}\PYG{o}{=} \PYG{n}{b}
\PYG{n+nb+bp}{True}
```

- == Eşittir operatörü python örnek :

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{5}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{==} \PYG{n}{b}
\PYG{n+nb+bp}{False}
```

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
```

```
\PYG{n}{a} \PYG{o}{==} \PYG{n}{b}
\PYG{n+nb+bp}{True}
```

- != Eşit değil operatörü python örnek :

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{5}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{!=} \PYG{n}{b}
\PYG{n+nb+bp}{True}
```

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{!=} \PYG{n}{b}
\PYG{n+nb+bp}{False}
```

- >= Büyük eşit operatörü python örnek :

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{5}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZgt{}}\PYG{o}{=} \PYG{n}{b}
\PYG{n+nb+bp}{False}
```

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZgt{}}\PYG{o}{=} \PYG{n}{b}
\PYG{n+nb+bp}{True}
```

- > Büyüktür operatörü python örnek :

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{5}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZgt{}} \PYG{n}{b}
\PYG{n+nb+bp}{False}
```

```
=
{}=
{}@default
=0=1=2=0=1=2\PYG{n}{a} \PYG{o}{=} \PYG{l+m+mi}{9}
\PYG{n}{b} \PYG{o}{=} \PYG{l+m+mi}{7}
\PYG{n}{a} \PYG{o}{\PYGZgt{}} \PYG{n}{b}
```

```
\PYG{n+nb+bp}{True}
```

Email Alarm nedir?

Iot Kanal/Element alarm işlemi uygulandığında gelen değer ile alarm değeri mantıksal operatör işlem sonucuna göre kanal yöneticisine email gönderilir. Kayıt olur iken kullanılan email adresi geçerli email adresidir. Günlük (24 saat) email gönderilme sayısı üyelik planına göre hesaplanır.

SMS Alarm nedir?

Iot Kanal/Element alarm işlemi uygulandığında gelen değer ile alarm değeri mantıksal operatör işlem sonucuna göre kanal yöneticisine sms gönderilir. Sms mesaj gönderilebilmesi için kanal yöneticisinin cep telefonunun onaylı olması gerekir. Günlük (24 saat) sms gönderilme sayısı üyelik planına göre hesaplanır.

Iot Mqtt Nedir?

MQTT Message Queuing Telemetry Transport kelimelerinin baş harfleri ile tanıdığımız bu teknoloji mesajın karşı tarafa ulaştırılması için kullanılan haberleşme protokolüdür. Haberleşme için mesaj yayınlayan, mesaja abone olan ve mesaj trafiğini kontrol eden yöneticiden oluşmaktadır.

Mesaj trafiğini kontrol eden yöneticiye BROKER, mesaj yayınına PUBLISH ve aboneye SUBSCRIBE denir. Mesaj alışverişi publisher dan subscriber lara doğru yani yayıncılardan abonelere doğru olmaktadır.

=0.65=0.65@default

Şekil 1.2: *MQTT*

Mqtt Protokolü Nasıldır?

MQTT de asenkron haberleşme protokolü kullanılmaktadır. Mesaj yayıncıları ve mesaj alıcılar arasında eşzamansız olarak veri taşınmaktadır. Diğer haberleşme yapılarına göre basit oluşu ve minimum kaynak tüketmesi sebebiyle “machine-to-machine” (M2M) makineden makineye veri iletiminde ve (IOT) “Internet of Things” İnternete bağlı nesnelerin mesajlaşmasında tercih edilmektedir.

MQTT Temp Test Client

Iot MQTT Temp Test Client Mosquitto Brokera websocket ile gelen temp/random başlığını dinler.

Bu örnek ‘test.mosquitto.org’ sitesinden alınmıştır. MQTT Temp örneğinin orjinal kaynağına ‘desert-home.com’ adresinden ve Github üzerinden ulaşabilirsiniz. MQTT Brokera nasıl mesaj gönderebilirim?

Iothook MQTT brokerına veri göndermek için “temp/random” başlığı gönderilmelidir. Gönderilen değer -20 ile +50 aralığında kayar noktalı (float) veya tamsayı (int) formatında olmalıdır.

MQTT Broker kimlik doğrulama ile çalışır.

Örnek -> Mesaj yayınlama: 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 Kullanıcıları:

Kullanıcı Adı : pub_user Şifre : iothook_pub_user

Kullanıcı Adı : sub_user Şifre : iothook_sub_user

Kullanıcı Adı : pub_client Şifre : iothook_pub_client

Kullanıcı Adı : sub_client Şifre : iothook_sub_client

=0.65=0.65@default

Şekil 1.3: *MQTT Temp Test*

Temp client sayfasına https://iothook.com/mqtt/mqtt-temp-test/ adresinden ulaşabilirsiniz.

MQTT Test Client Publisher Subscriber

MQTT Brokera Mesaj Gönderme ve Abone Olma Mosquitto Brokera websocket ile gelen temp/random başlığını dinler.

Iothook MQTT brokerına veri göndermek için "temp/random" başlığı gönderilmelidir. Gönderilen veri kayar noktalı (float), tamsayı (int) veya string (text) formatında olabilir.

MQTT Broker kimlik doğrulama ile çalışır.

Örnek -> Mesaj yayınlama: 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 Kullanıcıları:

Kullanıcı Adı : pub_user Şifre : iothook_pub_user

Kullanıcı Adı : sub_user Şifre : iothook_sub_user

Kullanıcı Adı : pub_client Şifre : iothook_pub_client

Kullanıcı Adı : sub_client Şifre : iothook_sub_client

=0.65=0.65@default

Şekil 1.4: *MQTT Test Publisher Subscriber*

Temp client sayfasına https://iothook.com/mqtt/mqtt-temp-test-pub-sub/ adresinden ulaşabilirsiniz.

Full Featured MQTT Client

MQTT Brokera Mesaj Gönderme ve Alma

MQTT Broker kimlik doğrulama ile çalışır.

Örnek -> Mesaj yayınlama: 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 Kullanıcıları:

Kullanıcı Adı : pub_user Şifre : iothook_pub_user

Kullanıcı Adı : sub_user Şifre : iothook_sub_user

Kullanıcı Adı : pub_client Şifre : iothook_pub_client

Kullanıcı Adı : sub_client Şifre : iothook_sub_client

Websockets Client Uygulaması Apache License Version 2.0 ile HiveMQ http://www.hivemq.com/ tarafından dağıtılmaktadır. Lisans hakkında ayrıca bilgi alınız.

Temp client sayfasına https://iothook.com/mqtt/full-featured-mqtt-client/ adresinden ulaşabilirsiniz.

IHook Nedir?

Iot Dashboard Nedir?

Banana Pi, NanoPC, Intel Edison, Parallella, Raspberry Pi gibi tek kart bilgisayarlarda çalışan Python/Django REST framework ile geliştirilmiş Web Api servisidir. IOT cihazlar ile iletişime geçerek Web Api sayesinde GET, POST, PUT ve DELETE işlemlerini kolayca yapabilmek için tasarlanmıştır. Iotdashboard tüm cihazlarınız arasında kesintisiz veri aktarımı yapan, internete bağlı nesnelerin kolayca ulaşabileceği iletişim protokollerini destekler. Google developer chart apileri ile entegre olarak verileri gerçek zamanlı izleme olanağı sağlar. Proje iOTHOOK tarafından açık kaynak olarak geliştirilmiş ve MIT lisansı ile dağıtılmaktadır. Kaynak kodlara http://electrocoder.github.io/iotdashboard/ buradan ulaşabilirsiniz.

IHook GITHUB

Iot Dashboard GITHUB Iot dashboard projesi Raspberry Pi türevi tek kart bilgisayarlar için geliştirilmiş Django Rest framewok server projesidir. Proje iOTHOOK tarafından açık kaynak olarak geliştirilmiş ve MIT lisansı ile dağıtılmaktadır. Kaynak kodlara http://electrocoder.github.io/iotdashboard/ buradan ulaşabilirsiniz.