

La inteligencia de las cosas

Introducción a IoT utilizando IBM IoT Foundation

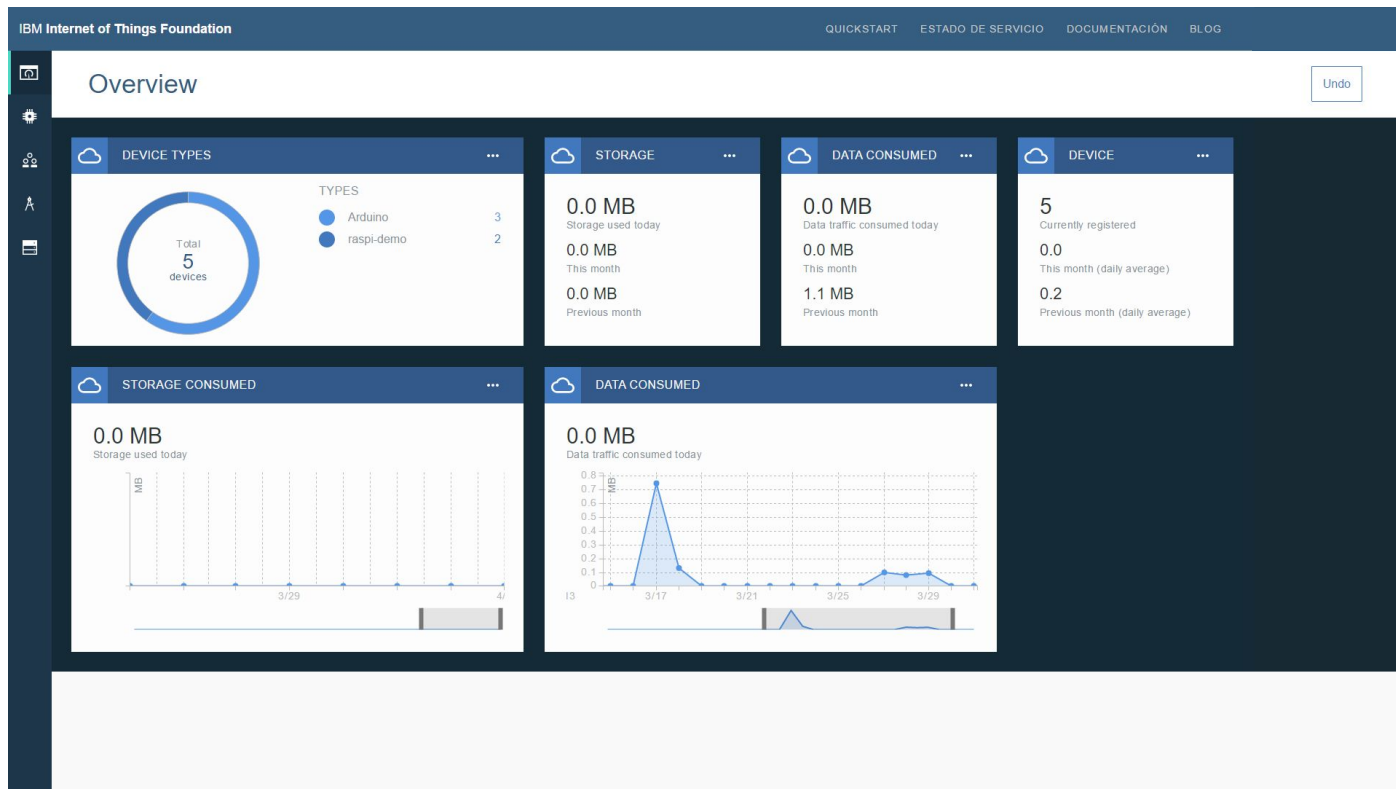
Internet of Things

- Es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet
- Objetos como termostatos, iluminación, vehículos o frigoríficos estarán conectados a internet para permitirnos conocer y modificar su estado de forma remota

IBM IoT Foundation

- Herramienta para facilitar el trabajo en IoT
 - gestión de dispositivos
 - gestión de aplicaciones
 - vista rápida de consumo






Vista previa



Dispositivos

IBM Internet of Things Foundation











QUICKSTARTESTADO DE SERVICIODOCUMENTACIÓNBLOG



Dispositivos

examinar | diagnosticar | acción | tipos de dispositivo






+ Añadir dispositivo

ID de dispositivo	Tipo de dispositivo	ID de clase	Fecha en la que se ha añadido
Resultados 1-5 de 5			
  light-sensor	Arduino	Dispositivo	28 de mar. de 2016 16:11:27
  raspi-demo	raspi-demo	Dispositivo	27 de mar. de 2016 12:47:13
  raspi2-demo	raspi-demo	Dispositivo	28 de mar. de 2016 16:10:43
  button	Arduino	Dispositivo	28 de mar. de 2016 16:12:21
  temp-sensor	Arduino	Dispositivo	28 de mar. de 2016 16:11:55

Acceso

IBM Internet of Things Foundation









QUICKSTARTESTADO DE SERVICIODOCUMENTACIÓNBLOG



Acceso

miembros | invitados | **claves api** | aplicaciones bluemix

+ Generar clave API

	Clave	Comentario	
	 a-35oqk3-evelbb3hym	-	 
	 a-35oqk3-wwftekup8a	demo	 

¡Hora de dar inteligencia a objetos!



Desarrollando para IoT Foundation

- Usando Node-RED
- Con código (<https://docs.internetofthings.ibmcloud.com/>)
 - Llamadas HTTP a la API
 - Protocolo MQTT
 - Librerías en:
 - Python
 - NodeJS
 - Java
 - C#
 - En dispositivos también disponible Embedded C y mBed C++

Conectando dispositivos

1

Crear nueva categoría para el dispositivo

Conectando dispositivos

1

Crear nueva categoría para el dispositivo

2

ID de organización	35oqk3
Tipo de dispositivo	raspi-demo
ID de dispositivo	sensor-demo
Método de autenticación	token
Señal de autenticación	eyJhbGciOiJIUzI1NiIsInR5cCI6Ikpz...

Conectando dispositivos

1

Crear nueva categoría para el dispositivo

2

ID de organización	35oqk3
Tipo de dispositivo	raspi-demo
ID de dispositivo	sensor-demo
Método de autenticación	token
Señal de autenticación	eyJhbGciOiJIUzI1NiIsInR5cCI6Ikpz...

3

Fichero de configuración (*device.cfg*)

```
[device]
org=$orgId
type=$myDeviceType
id=$myDeviceId
auth-method=token
auth-token=$token
```

Conectando dispositivos

1

Crear nueva categoría para el dispositivo

2

ID de organización	35oqk3
Tipo de dispositivo	raspi-demo
ID de dispositivo	sensor-demo
Método de autenticación	token
Señal de autenticación	...

3

Fichero de configuración (*device.cfg*)

```
[device]
org=$orgId
type=$myDeviceType
id=$myDeviceId
auth-method=token
auth-token=$token
```

4

Script de conexión

```
import ibmiotf.device

try:
    options = ibmiotf.device.ParseConfigFile('device.cfg')
    client = ibmiotf.device.Client(options)
    client.connect()
except ibmiotf.ConnectionException as e:
    print e
```

Enviando datos

```
import ibmiotf.device
import psutil
import time

def publishStatus():
    cpu_percent = psutil.cpu_percent(interval=None)
    virtual_memory_percent = psutil.virtual_memory().percent
    disk_usage = psutil.disk_usage('/').percent
    myData={'cpu_percent' : cpu_percent, 'virtual_memory_percent' : virtual_memory_percent, 'disk_usage' : disk_usage}
    return myData

try:
    options = ibmiotf.device.ParseConfigFile('device.cfg')
    client = ibmiotf.device.Client(options)
    client.connect()

    while True:
        myData = publishStatus()
        client.publishEvent("device_status", "json", myData)
        time.sleep(1)

    client.disconnect()
except ibmiotf.ConnectionException as e:
    print e
```

Enviando datos

Sucesos recientes			
Suceso	Formato	Hora de recepción	
device_status	json	28 de mar. de 2016 18:52:27	
device_status	json	28 de mar. de 2016 18:52:28	
device_status	json	28 de mar. de 2016 18:52:29	
device_status	json	28 de mar. de 2016 18:52:30	
device_status	json	28 de mar. de 2016 18:52:31	
device_status	json	28 de mar. de 2016 18:52:32	
device_status	json	28 de mar. de 2016 18:52:33	
device_status	json	28 de mar. de 2016 18:52:34	
device_status	json	28 de mar. de 2016 18:52:35	
device_status	json	28 de mar. de 2016 18:52:36	
Información del sensor			
Suceso	Punto de datos	Valor	Hora de recepción
device_status	disk_usage	94.3	28 de mar. de 2016 18:52:36
device_status	virtual_memory_percent	15.8	28 de mar. de 2016 18:52:36
device_status	cpu_percent	1	28 de mar. de 2016 18:52:36

Enviando datos



Recibiendo datos

1

ave API a-35oqk3-9qunbp7g9l

Señal de autenticación 

Las señales de autenticación no son recuperables. Si pierde esta señal, deberá volver a registrar la clave API para generar una señal de autenticación nueva.

Comentario Ejemplo de clave

Este comentario será visible para todos los invitados y miembros de esta organización y debe describir el uso de la clave.

Recibiendo datos

1

Clave API

Señal de autenticación

Las señales de autenticación no son recuperables. Si pierde esta señal, deberá volver a registrar la clave API para generar una señal de autenticación nueva.

Comentario

Este comentario será visible para todos los invitados y miembros de esta organización y debe describir el uso de la clave.

2


Fichero de configuración (*app.cfg*)

```
[device]
org=$orgId
type=$myDeviceType
id=$myDeviceId
auth-method=token
auth-token=$token
```

Recibiendo datos

1

ave API a-35oqk3-9qunbp7g9l

Señal de autenticación 

Las señales de autenticación no son recuperables. Si pierde esta señal, deberá volver a registrar la clave API para generar una señal de autenticación nueva.

Comentario Ejemplo de clave

Este comentario será visible para todos los invitados y miembros de esta organización y debe describir el uso de la clave.

2

Fichero de configuración (*app.cfg*)

```
[device]
org=$orgId
type=$myDeviceType
id=$myDeviceId
auth-method=token
auth-token=$token
```

3

```
import ibmiotf.application

def eventCallBack(event):
    print event.data

try:
    options = ibmiotf.application.ParseConfigFile( 'app.cfg')
    client = ibmiotf.application.Client(options)

    client.connect()
    client.deviceEventCallback = eventCallBack
    client.subscribeToDeviceEvents()

    while True:
        pass

    client.disconnect()

except ibmiotf.ConnectionException as e:
    print e
```

Recibiendo datos



Enviando comandos

Script para enviar comandos (app de control)

```
import ibmiotf.application

try:
    options = ibmiotf.application.ParseConfigFile( 'app.cfg')
    client = ibmiotf.application.Client(options)

    client.connect()

    data={'delay' : 1}
    client.publishCommand("raspi-demo", "raspi-demo", "reboot",
"json", data)

    client.disconnect()

except ibmiotf.ConnectionException as e:
    print e
```

Script para recibir comandos (device)

```
import ibmiotf.device

def commandCallback(cmd):
    if cmd.command == "reboot":
        if 'delay' not in cmd.data:
            restart(0)
        else:
            restart(cmd.data['delay'])

def restart(time):
    # Code to restart the system in 'time' minutes

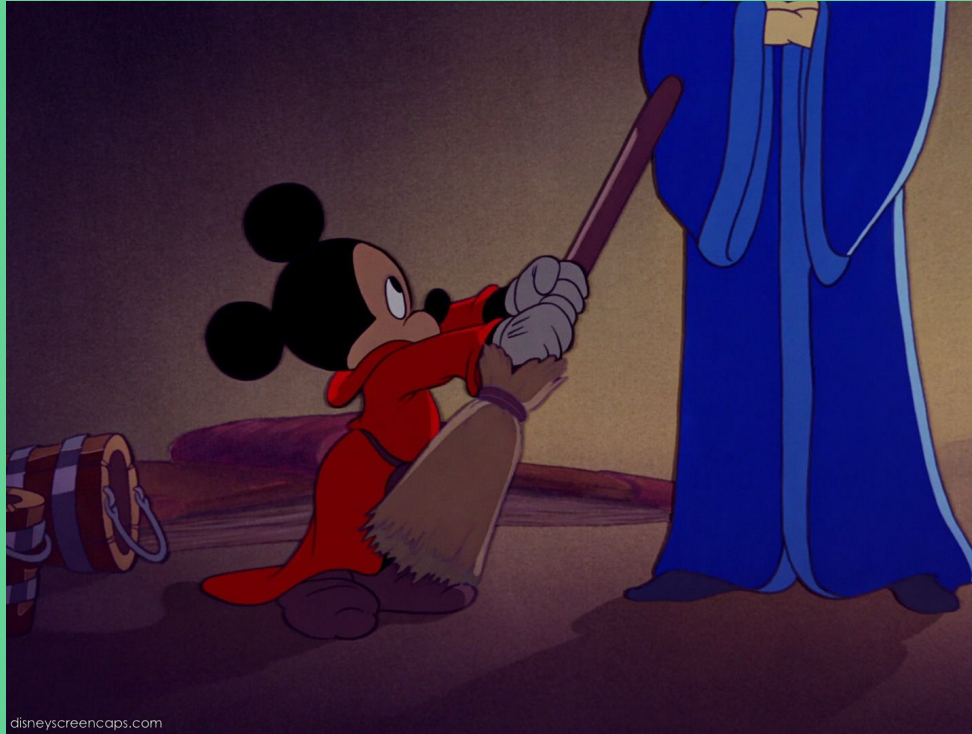
try:
    options = ibmiotf.device.ParseConfigFile( 'device.cfg')
    client = ibmiotf.device.Client(options)
    client.connect()
    client.commandCallback = commandCallback

except ibmiotf.ConnectionException as e:
    print e
```

Enviando comandos



¿Preguntas?



Recursos

- [IBM Bluemix](#)
- [IBM IoT Foundation](#)
- [Código de la demo](#)
- [Post en Betabeers](#)
- [IBM IoT Developer Receipes](#)
- [IBM Watson IoT](#)