

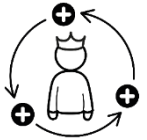
# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?

MIGUEL ANGEL GOMEZ  
SAP Presales Specialist. SAP España



#SAPForum

## Starter Kit for the SAP HANA Cloud Platform Internet of Things (IoT) Services



HCP IoT Services Overview



Getting Started in HCP



Sending messages from the device  
(simulation)



Pushing messages to the device  
(simulation)



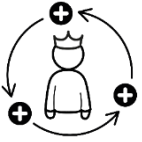
Integrated examples for IoT Devices



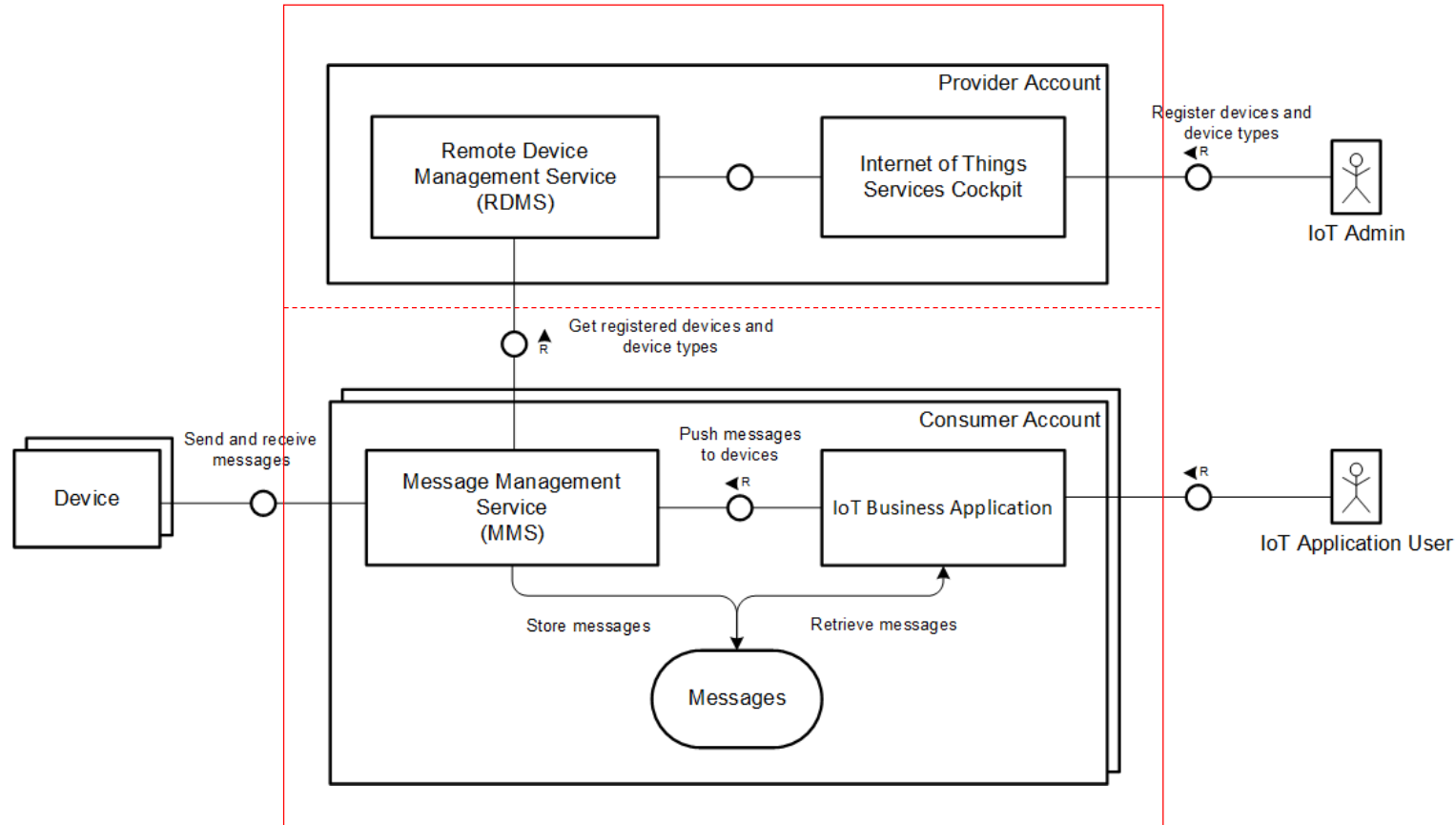
What comes next



# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## HCP IoT Services Overview



# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Getting Started in HCP

1. Get HANA Cloud Platform Developer Account -> <https://account.hanatrial.ondemand.com>

**Registro**

Datos personales

Nombre

Apellido 1 \*

Correo electrónico \*

**Establecer contraseña**

Clave de acceso \*

Vuelva a introducir la contraseña \*

**Preferencias de contacto**

Además de las comunicaciones como resultado del registro, ¿desea recibir también noticias y notificaciones de eventos de SAP que puedan ser de su interés?

Por correo electrónico\*

☐ Sí ☐ No

Por teléfono\*

☐ Sí ☐ No

**Términos y condiciones**

☐ He leído y acepto la [SAP declaración de protección de datos\\*](#)

☐ He leído y entendido las condiciones comerciales de SAP HANA Cloud Platform.\*

\*Obligatorio

**Registrar**

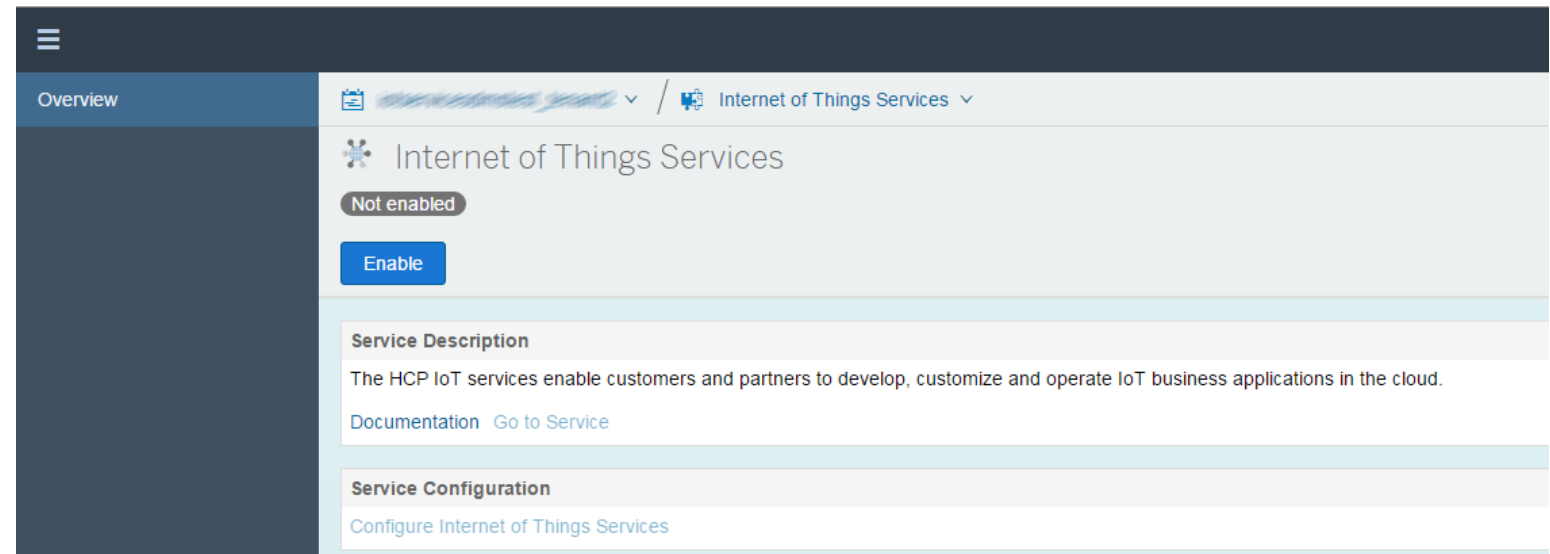
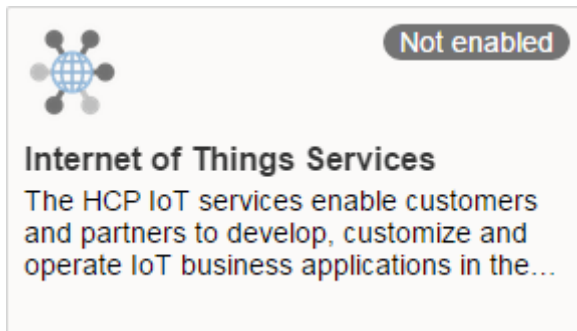
Powered by Cloud Identity

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Getting Started in HCP

1. Get HANA Cloud Platform Developer Account -> <https://account.hanatrial.ondemand.com>
2. Enable Internet of Things Services



# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Getting Started in HCP

1. Get HANA Cloud Platform Developer Account -> <https://account.hanatrial.ondemand.com>
2. Enable Internet of Things Services
3. Deploy the Message Management Service (MMS)

Deploy Message Management Service

The Message Management Service (MMS) receives and processes messages sent from devices. In addition, it provides interfaces to push messages to devices. The component will be deployed on the HCP user account specified.

ACCOUNT SETTINGS

Host <https://hanatrial.ondemand.com>

Account ID [p194173953trial](#)

USER SETTINGS

User Name [P1941739532](#)

Password \*\*\*\*\*

Deploy

Internet of Things Services Cockpit

Device Management

- 1 Device Types (All Device Types)
- 2 Message Types (All Message Types)
- 1 Devices (All Registered Devices)

Message Management

- Deploy Message Management Service (MMS Deployment)
- View messages received, use sample clients, etc. (MMS Cockpit)

Legal Disclosure Privacy Terms of Use Copyright Trademark SAP.com

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Getting Started in HCP

1. Get HANA Cloud Platform Developer Account -> <https://account.hanatrial.ondemand.com>
2. Enable Internet of Things Services
3. Deploy the Message Management Service (MMS)
4. Create Device Information in Internet of Things Services Cockpit

The image displays four overlapping screenshots from the SAP IoT Services Cockpit interface:

- Top Left:** 'Device Types' view showing a search bar and a list with 'No data'.
- Bottom Left:** 'New Device Type' form. The 'Information' section has '\*Name: GrovePi'. The 'Fields' section contains a table with 3 rows: (1, 'sensor', 'string'), (2, 'value', 'string'), and (3, 'timestamp', 'date').
- Top Right:** 'Devices' view showing a search bar and a list with 'MyGrovePi' and 'MyGrovePi\_SW'. The 'MyGrovePi\_SW' device is selected, showing its 'Information' and 'Authentication' tabs.
- Bottom Right:** 'New Message Type' form. The 'Information' section has '\*Name: InboundMessage', '\*Device Type: GrovePi', and '\*Direction: To Device'. The 'Fields' section contains a table with 2 rows: (1, 'opcode', 'string') and (2, 'operand', 'string').

Overlaid on the bottom right screenshot is a modal dialog titled 'OAuth Access Token' with a checked checkbox. It displays the message: 'New OAuth access token for device "MyGrovePi\_SW" created. Token: 7a3c783e26db620d0a73ad3da26be0'. There is a 'Close' button at the bottom right of the dialog.

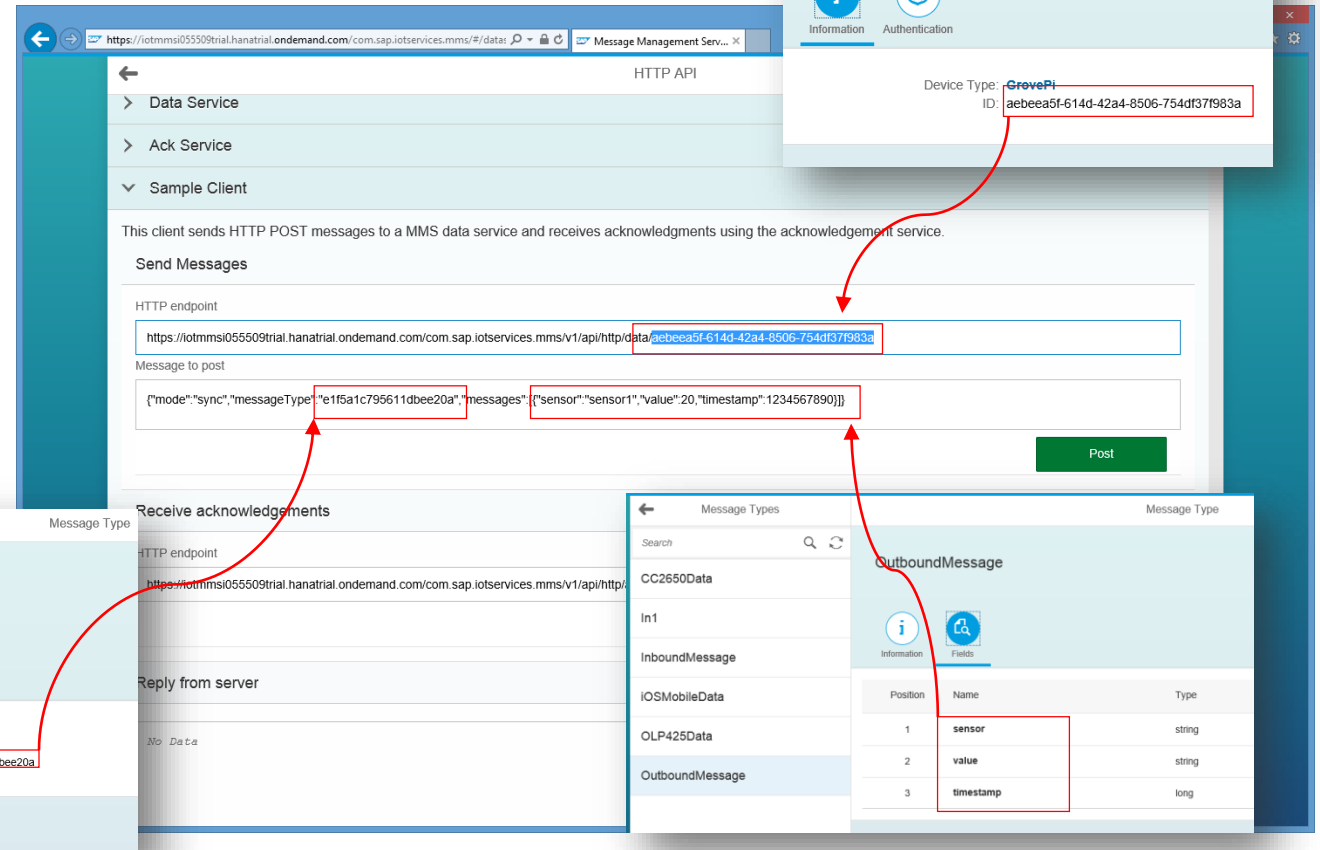
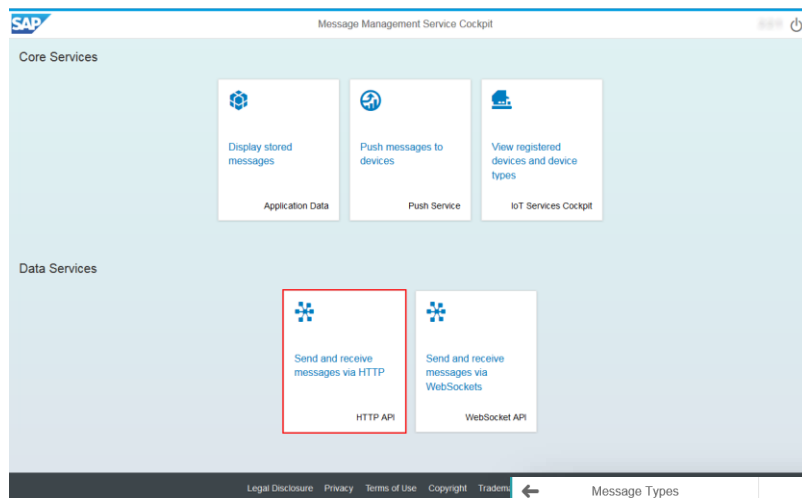


# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



Sending messages from the device (simulation)

1. Send messages with MMS embedded sample client using HTTP API





# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



Sending messages from the device (simulation)

## 1. Send messages with MMS embedded sample client using HTTP API (result)

The screenshot shows the 'Getting acknowledgements' section of the SAP IoT Services HTTP API documentation. A red box highlights the status codes and their meanings:

- 200 - request processed successfully. Final state.
- 202 - request accepted, info message gives insight if request is still to be validated (e.g. in case a synchronization with RDMS has to be done before) or if the request is already validated and in processing queue. Intermediate state.
- 400 - validation of request failed, info message gives insight about the type of request failure. Final state.
- 404 - either the given device or sequence is unknown.
- 413 - request too large, final state.
- 500 - processing failed, info message gives insight about type of server error. Final state.
- 503 - request cannot be processed at the moment, probably due to temporary overload. Final state.

The text explains that the HTTP response contains the status of the processing of the sequence as HTTP response code, and the JSON body contains additional information. It also provides the URL for sending messages and the format of the JSON response.

The screenshot shows the SAP IoT Services HTTP API interface. The 'Send Messages' section displays the HTTP endpoint and the message to post. The 'Receive acknowledgements' section displays the HTTP endpoint. The 'Reply from server' section shows the response from the server, which is highlighted with a red box:

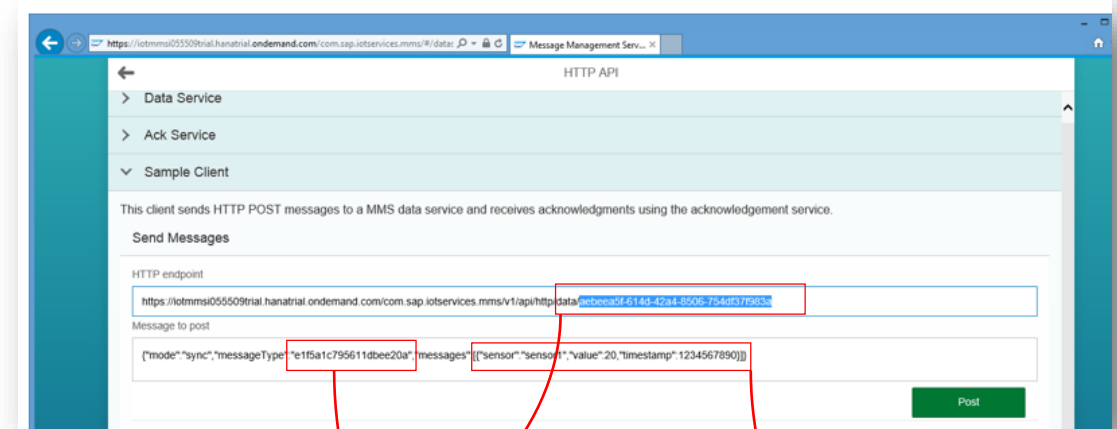
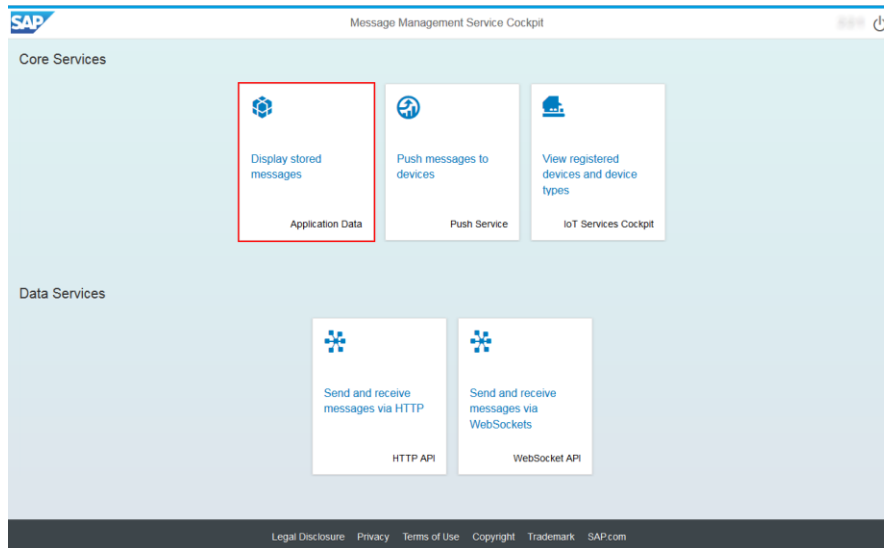
```
[200] [{"msg": "1 message(s) received from device [aeb5f-614d-42a4-8506-754df37f983a]"}]
```

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



Sending messages from the device (simulation)

2. Consume the messages from the device with MMS embedded client



G_DEVICE	G_CREATED	C_SENSOR	C_TIMESTAMP	C_VALUE
aebeea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 12:39:40 GMT+0100 (W. Europe Standard Time)	sensor1	1234567890	20
aebeea5f-614d-42a4-8506-754df37f983a	Mon Jan 25 2016 16:11:41 GMT+0100 (W. Europe Standard Time)	slider_desktop	1453734698	5
aebeea5f-614d-42a4-8506-754df37f983a	Mon Jan 25 2016 16:11:41 GMT+0100 (W. Europe Standard Time)	slider_desktop	1453734697	78
aebeea5f-614d-42a4-8506-754df37f983a	Mon Jan 25 2016 16:11:36 GMT+0100 (W. Europe Standard Time)	slider_desktop	1453734695	78

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?

SAP



Pushing messages to the device (simulation)

1. Push messages with MMS embedded sample client using HTTP and WebSocket API

Message Management Service Cockpit

Core Services

- Display stored messages (Application Data)
- Push messages to devices (Push Service)**
- View registered devices and device types (IoT Services Cockpit)

Data Services

- Send and receive messages via HTTP (HTTP API)
- Send and receive messages via WebSockets (WebSocket API)

Message Types

Search
CC2650Data
In1
<b>InboundMessage</b>
IOSMobileData
OLP425Data
OutboundMessage

InboundMessage

Device Type: GrovePi  
Direction: To Device  
ID: 329965b8028364528605

MyGrovePi\_SW

Information Authentication

Device Type: GrovePi  
ID: aebeea5f-614d-42a4-8506-754df37f983a

Push Service

Sample Client

This client sends push messages to a device.

Push messages

Device ID: aebeea5f-614d-42a4-8506-754df37f983a

Push via: http

Sender: IoT App -SIMULACION-

Message to push: {"messageType":"329965b8028364528605","messages":[{"opcode":"display","operand":"Esto funciona fenomenal!"}]}

Reply from server

No Data

Message Types

Search
CC2650Data
In1
<b>InboundMessage</b>
IOSMobileData
OLP425Data
OutboundMessage

InboundMessage

Position	Name	Type
1	opcode	string
2	operand	string

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



Pushing messages to the device (simulation)

## 1. Push messages with MMS embedded sample client using HTTP and WebSocket API (result)

**Getting acknowledgements**

In order to request acknowledgements for a given device and sequence, do a HTTP POST with an empty body to [https://iotmms055509trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/ack/\[device id\]/\[sequence id\]](https://iotmms055509trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/ack/[device id]/[sequence id]).

The HTTP response contains the status of the processing of the sequence as HTTP response code, the JSON body contains additional information. The status codes have the following semantics:

- 200 - request processed successfully. Final state.
- 202 - request accepted, info message gives insight if request is still to be validated (e.g. in case a synchronization with RDMS has to be done before) or if the request is already validated and in processing queue. Intermediate state.
- 400 - validation of request failed, info message gives insight about the type of request failure. Final state.
- 404 - either the given device or sequence is unknown
- 413 - request too large, final state.
- 500 - processing failed, info message gives insight about type of server error. Final state.
- 503 - request cannot be processed at the moment, probably due to temporary overload. Final state.

The codes 200, 400, 413, 500 and 503 mark final states. Information about a sequence being in one of these states is only delivered once, then it is deleted from the server. An attempt to query the ack service for the same sequence again will result in code 404.

In order to request all acknowledgements for a device do a HTTP POST with an empty body to [https://iotmms055509trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/ack/\[device id\]](https://iotmms055509trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/ack/[device id]).

The JSON response contains all pending acknowledgements containing sequence numbers together with the corresponding response code in the format:

```
{ "statusCode": 200, "messageType": "ack", "sequence": 1 }
```

**Sample Client**

This client sends HTTP POST messages to a MMS data service and receives acknowledgments using the acknowledgement service.

**Send Messages**

HTTP endpoint  
<https://iotmms055509trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/aebaea5f-614d-42a4-8506-754df37f983a>

Message to post  

```
{ "mode": "sync", "messageType": "e1f5a1c795611dbe20a", "messages": [{"sensor": "sensor1", "value": 20, "timestamp": 1234567890}] }
```

**Post**

**Push Service**

**Push Service**

**Sample Client**

This client sends push messages to a device.

**Push messages**

Device ID  
[aebaea5f-614d-42a4-8506-754df37f983a](https://iotmms055509trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/aebaea5f-614d-42a4-8506-754df37f983a)

Push via  
http

Sender  
IoT App -SIMULACION-

Message to push  

```
{ "messageType": "329965b8028364528605", "messages": [{"opcode": "display", "operand": "Esto funciona fenomenal!!"}] }
```

**Push**

**Reply from server**

[202] {"msg": "Message(s) pushed to device [aebaea5f-614d-42a4-8506-754df37f983a]"}]

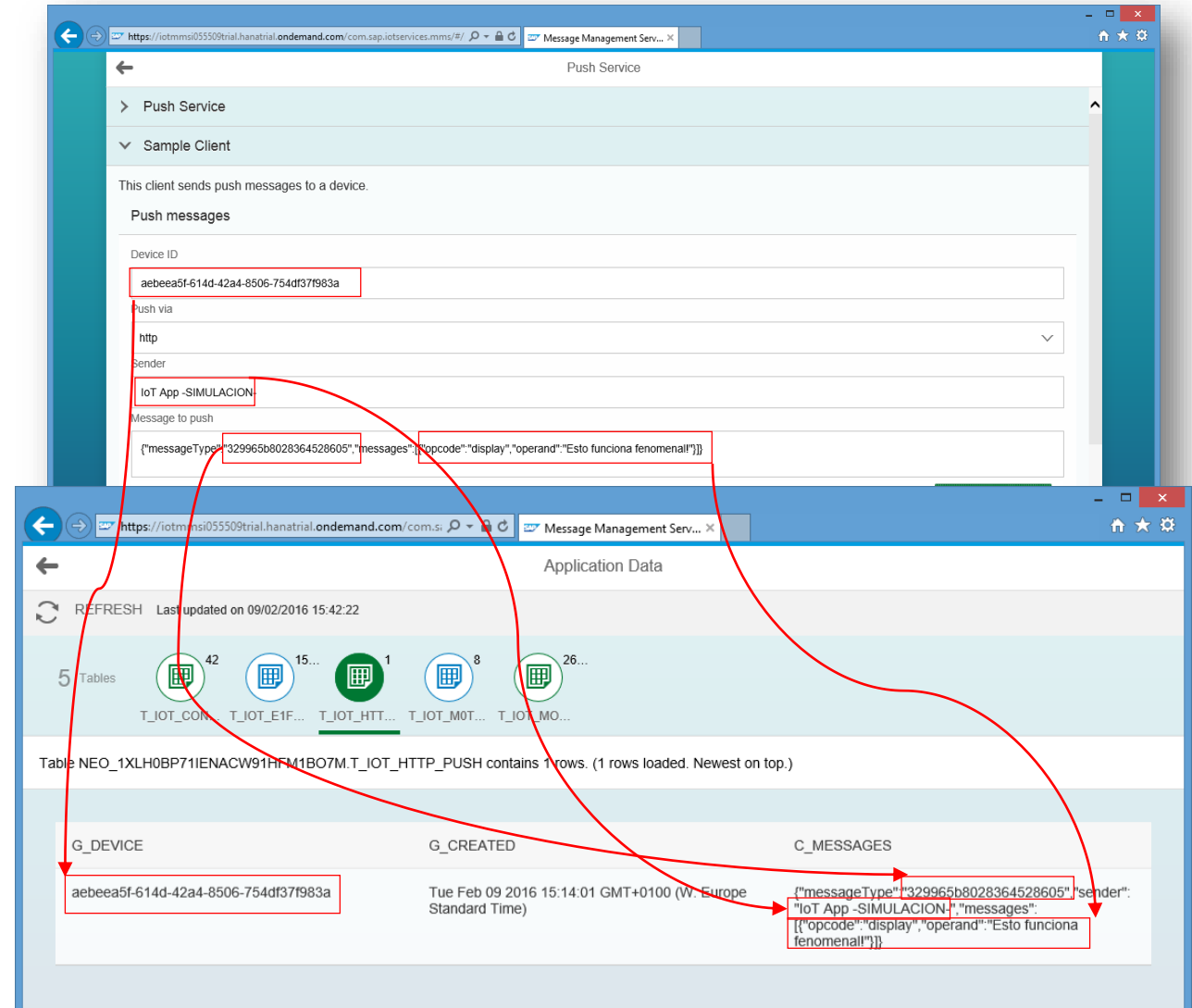
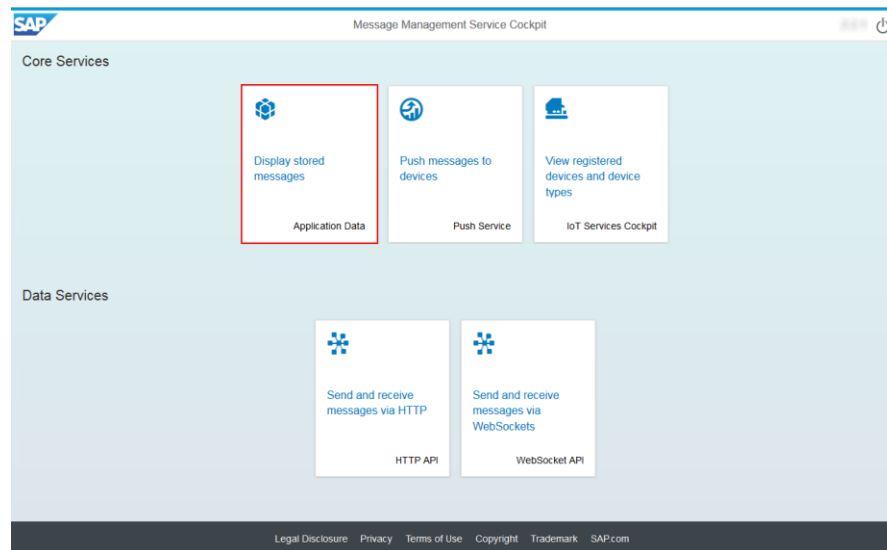


# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



Pushing messages to the device (simulation)

2. Receive the messages sent to the device with MMS embedded client

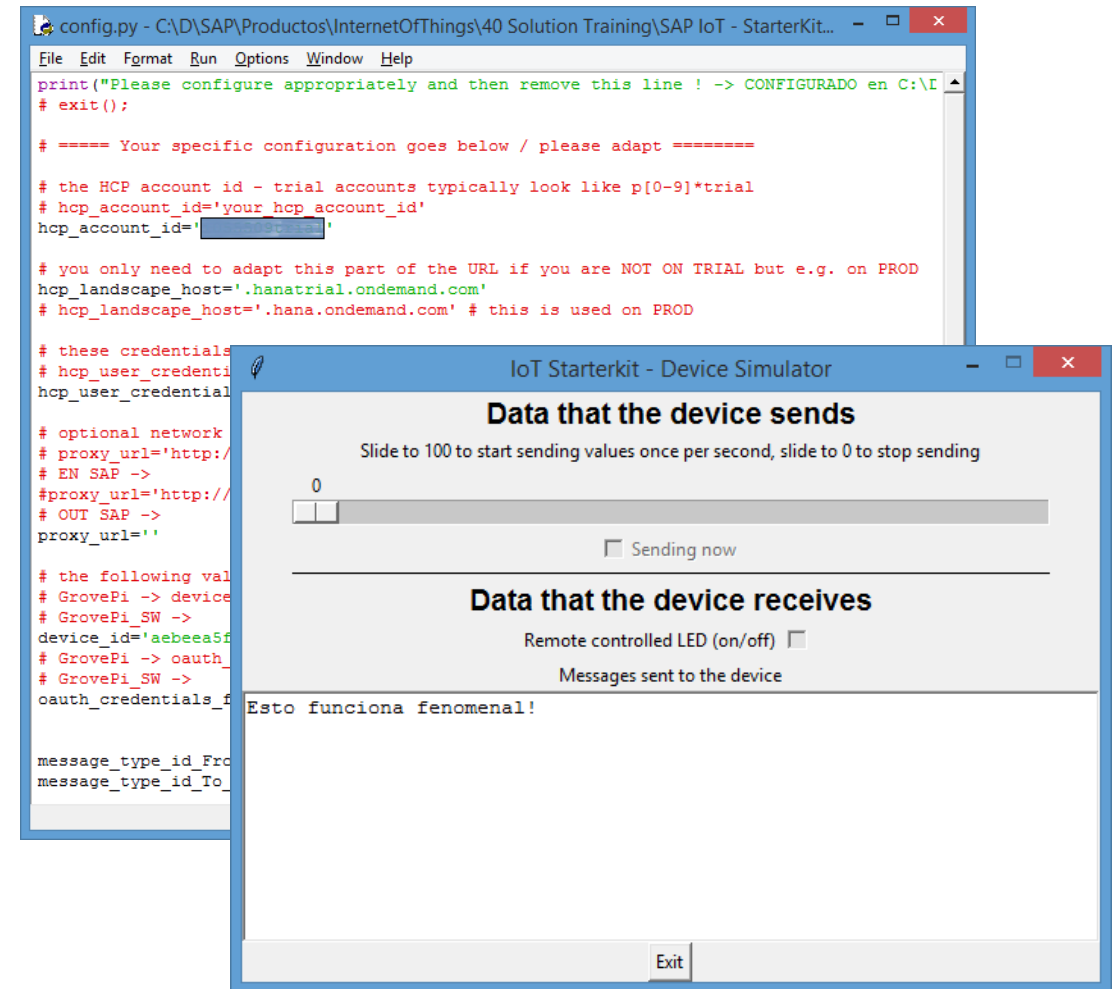
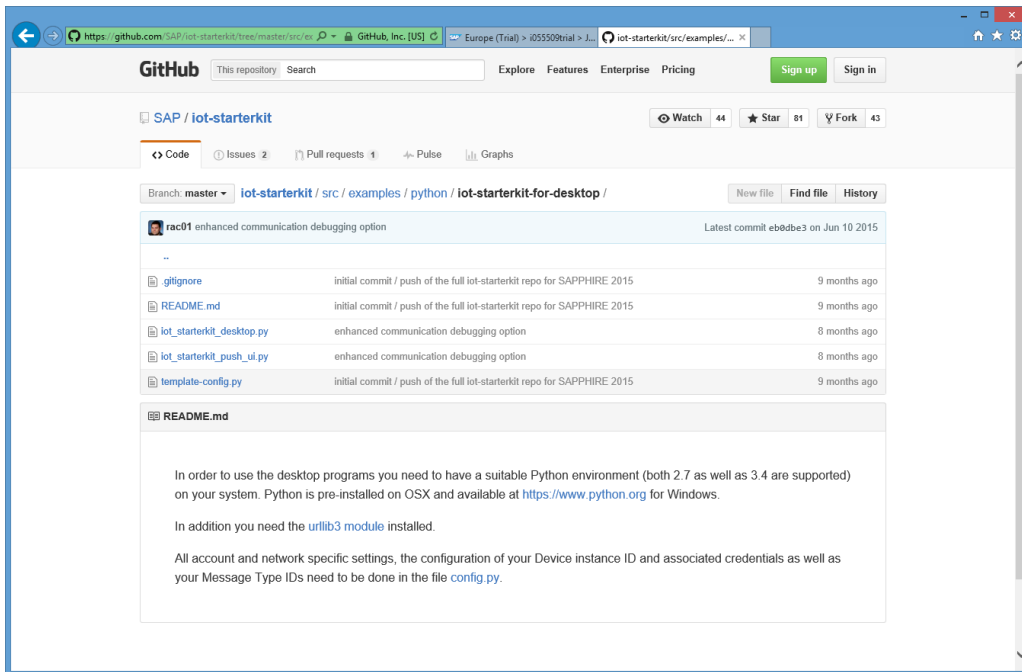


# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Integrated examples for IoT Devices

### 1. Working with device simulators (python)



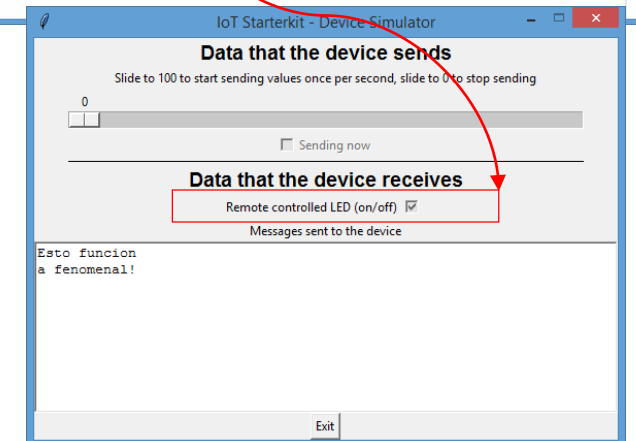
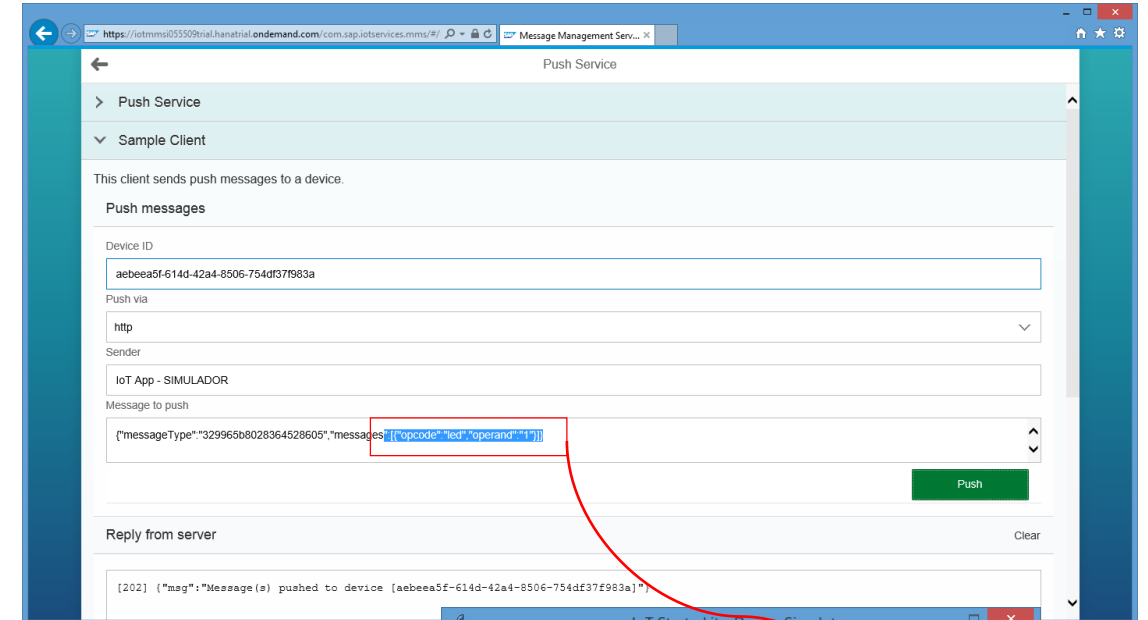
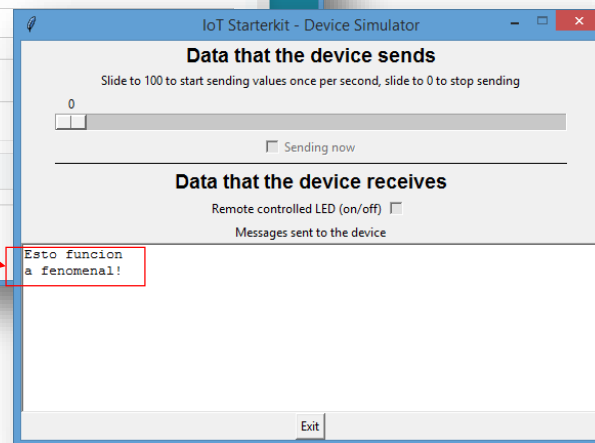
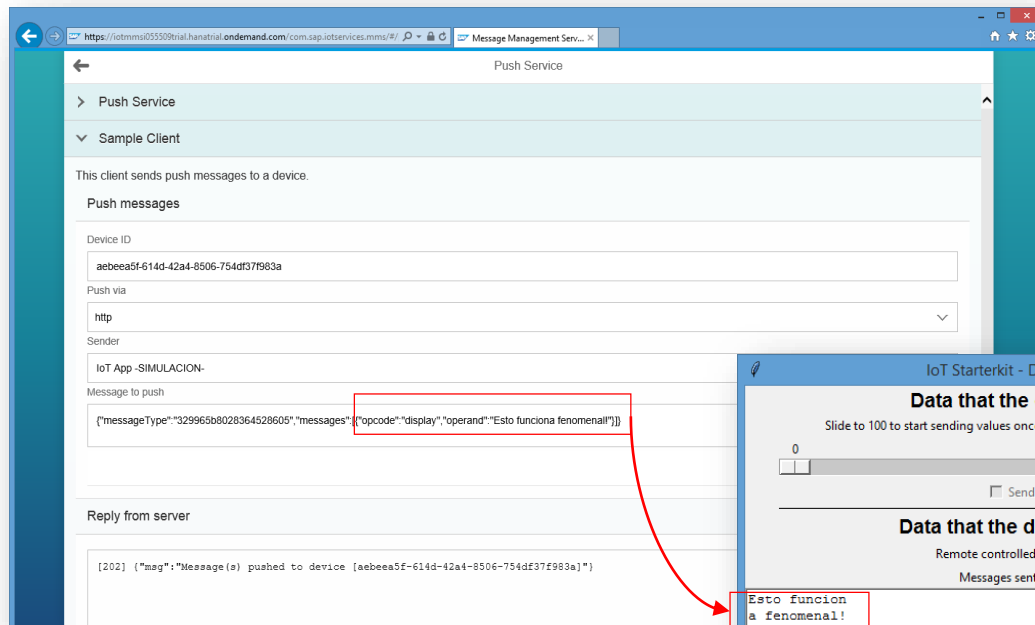
<https://github.com/SAP/iot-starterkit/tree/master/src/examples/python/iot-starterkit-for-desktop>

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Integrated examples for IoT Devices

2. Receive the messages sent to the device with device simulator (python)

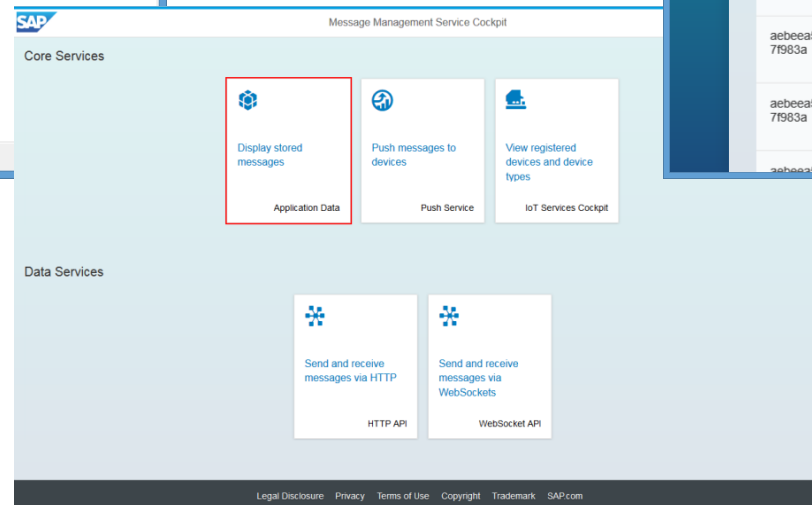
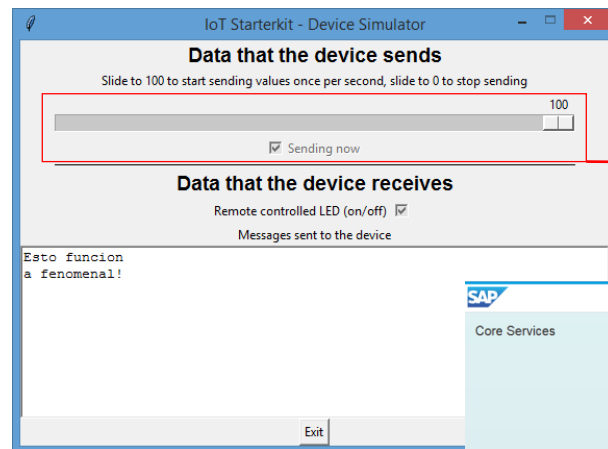


# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Integrated examples for IoT Devices

3. Consume the messages from the device simulator (python) with MMS embedded client



The screenshot shows the 'Message Management Service Cockpit' interface with the 'Application Data' table selected. The table displays data for the device 'aeb5ea5f-614d-42a4-8506-754df37f983a'. The table has five columns: 'G\_DEVICE', 'G\_CREATED', 'C\_SENSOR', 'C\_TIMESTAMP', and 'C\_VALUE'. The 'C\_VALUE' column is highlighted with a red box, and a red arrow points from the 'Sending now' checkbox in the device simulator to this column. The table contains 15 rows, with the first row showing a value of 74 and the subsequent rows showing a value of 100.

G_DEVICE	G_CREATED	C_SENSOR	C_TIMESTAMP	C_VALUE
aeb5ea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 16:42:19 GMT+0100 (W. Europe Standard Time)	slider_desktop	1455032538	74
aeb5ea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 16:42:19 GMT+0100 (W. Europe Standard Time)	slider_desktop	1455032536	74
aeb5ea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 16:42:19 GMT+0100 (W. Europe Standard Time)	slider_desktop	1455032535	100
aeb5ea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 16:42:14 GMT+0100 (W. Europe Standard Time)	slider_desktop	1455032533	100
aeb5ea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 16:42:14 GMT+0100 (W. Europe Standard Time)	slider_desktop	1455032532	100
aeb5ea5f-614d-42a4-8506-754df37f983a	Tue Feb 09 2016 16:42:14 GMT+0100 (W. Europe Standard Time)	slider_desktop	1455032531	100

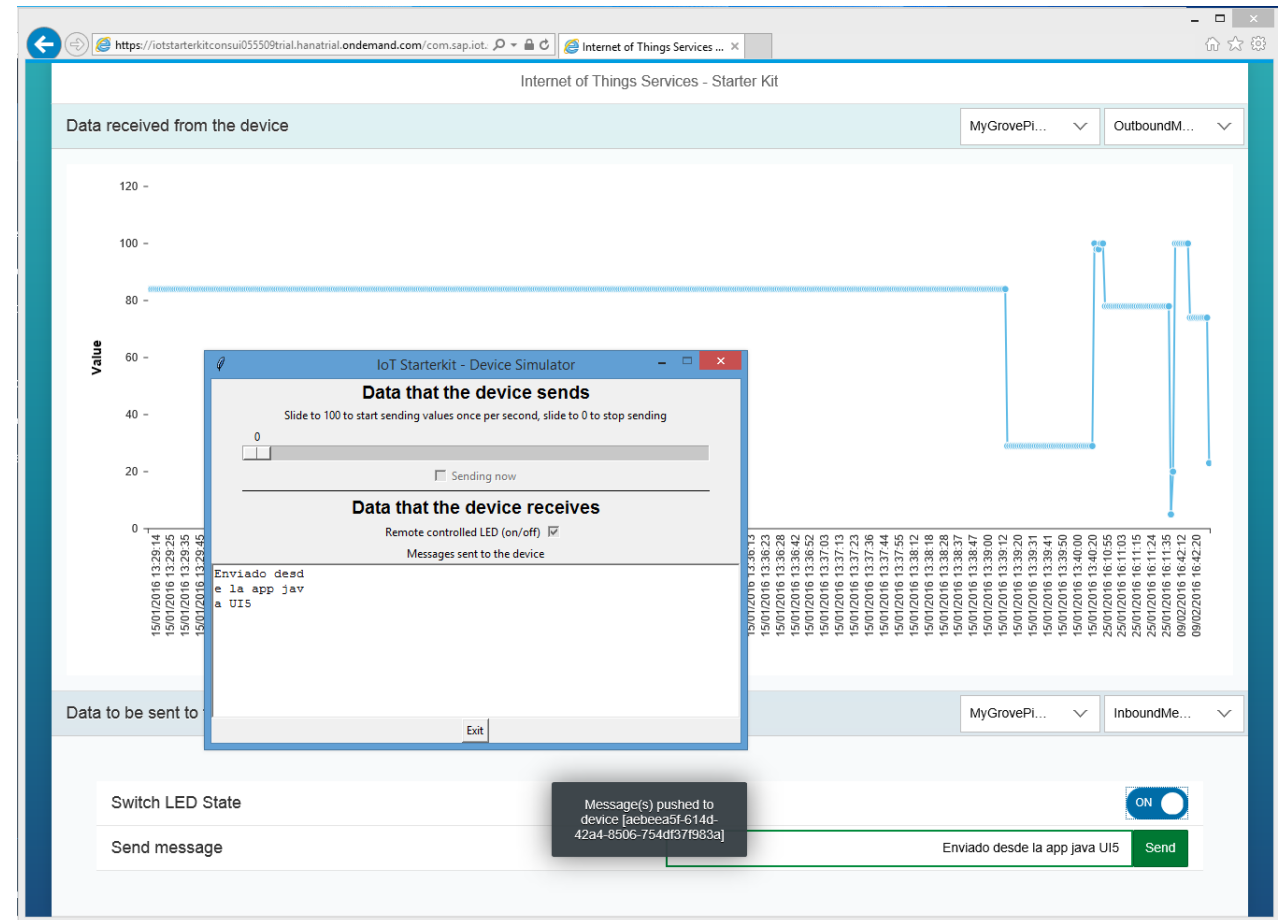
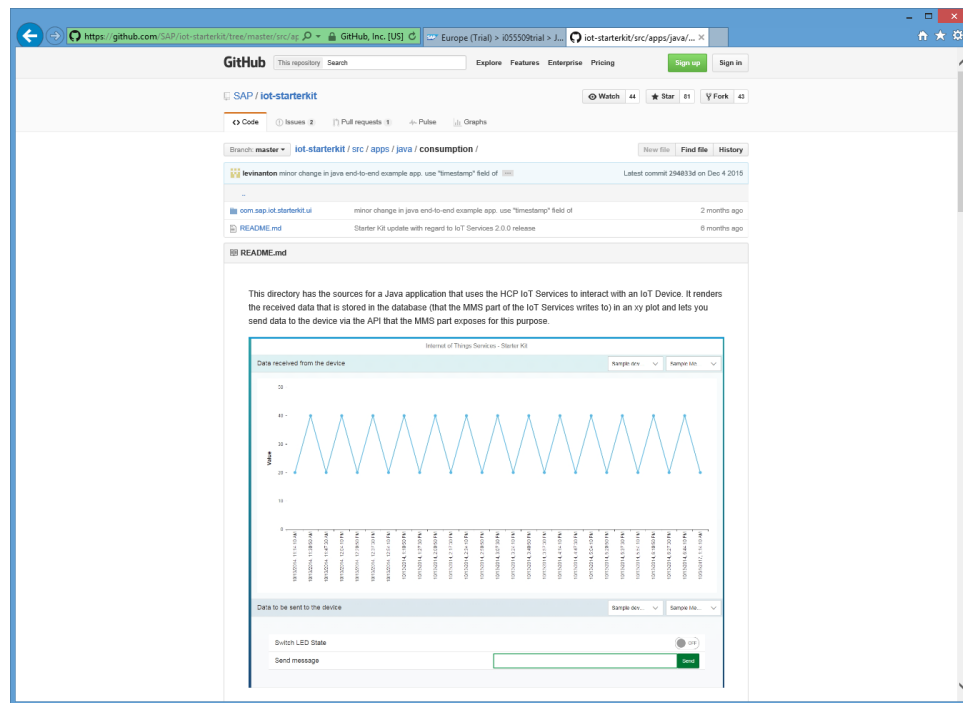


# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



## Integrated examples for IoT Devices

- Interact with Web Application based on Java and UI5 using HCP Persistence Service



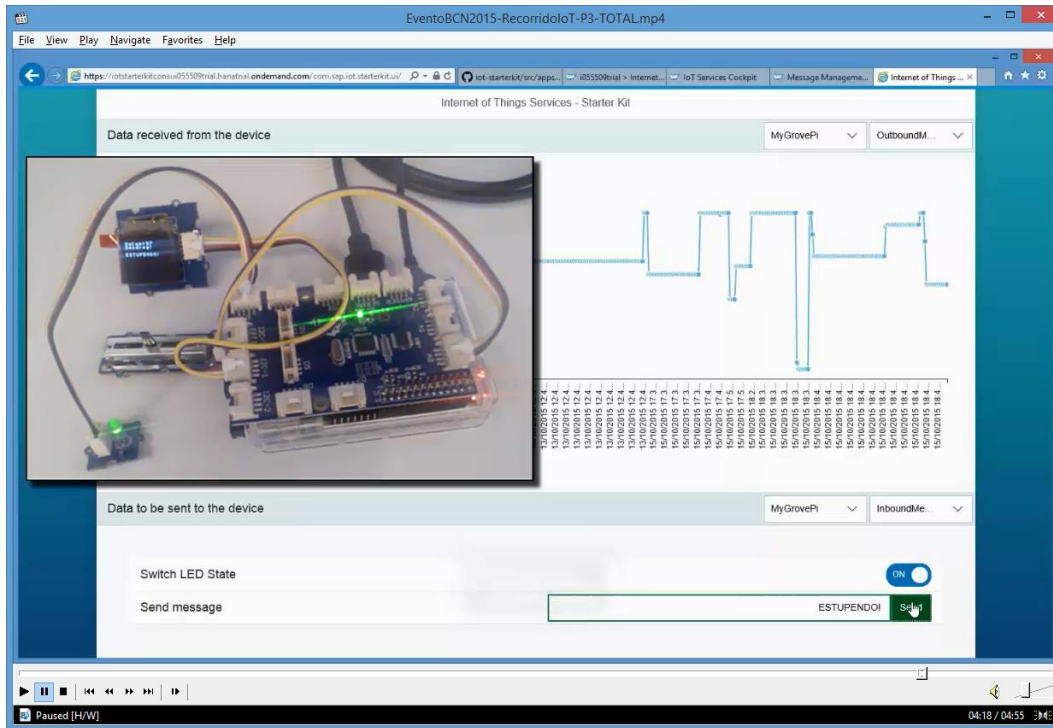
<https://github.com/SAP/iot-starterkit/tree/master/src/apps/java/consumption>

# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?



Integrated examples for IoT Devices

## 5. Working with real IoT hardware



Estamos en el área SAP – Booth 1 – escenarios HCP

<https://github.com/SAP/iot-starterkit/blob/master/src/hardware/raspberry-pi/README.md>

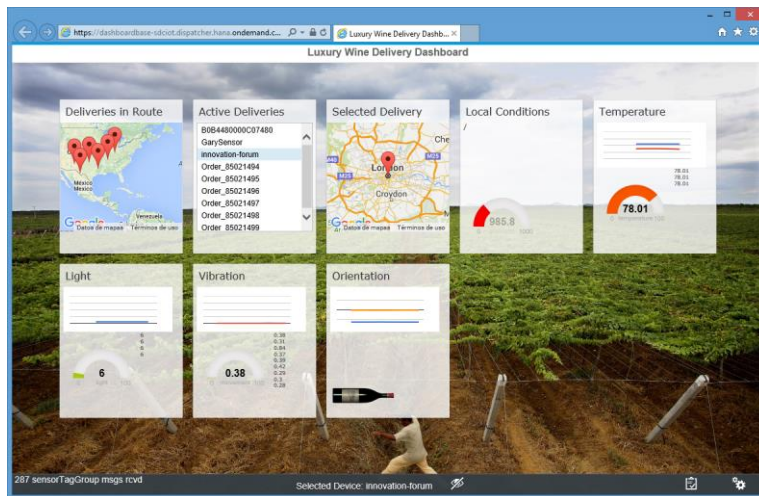


# ¿Cómo 'TRIUNFAR' con HCP IoT Services en 20'?

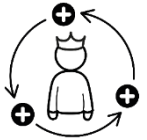


What comes next?

1. Try out the SAP HANA Cloud Platform Internet of Things (IoT) services (<http://scn.sap.com/docs/DOC-63811>)
2. Using the SAP HCP IoT Services (<http://scn.sap.com/community/developer-center/cloud-platform/blog/2015/06/09/using-the-sap-hcp-iot-services>)
3. RaspberryPi on SAP HANA Cloud Platform (<http://scn.sap.com/docs/DOC-56616>)
4. From zero to Smart Building IoT – SAP InnoJam story (<http://scn.sap.com/community/developer-center/cloud-platform/blog/2015/08/03/from-zero-to-smart-building-iot-sap-innojam-story>)



## Starter Kit for the SAP HANA Cloud Platform Internet of Things (IoT) Services



HCP IoT Services Overview



Getting Started in HCP



Sending messages from the device  
(simulation)



Pushing messages to the device  
(simulation)



Integrated examples for IoT Devices



What comes next





# GRACIAS

Miguel Angel Gómez

[m.gomez@sap.com](mailto:m.gomez@sap.com)

<http://es.linkedin.com/in/miguelangelgomezmoreno>



## #SAPForum

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.  
National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.