

SAP and Azure IoT Demo – „Hello World“

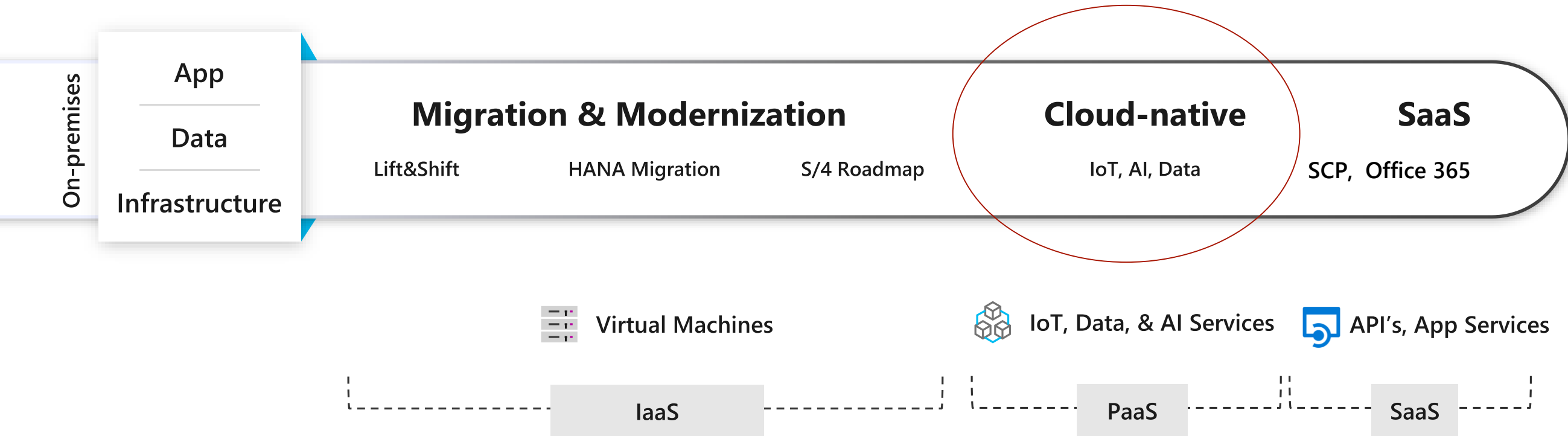
Roman Broich, Cloud Solution Architect – *SAP and Azure*

Demo introduction

GEWINNEN GEMEINSAM
EINFACH
MIT PARTNERN

DAS OCP PRINZIP

The SAP on Azure customer cloud journey



Demo motivation and positioning

- Demonstrate Lego-like integration between Azure IoT and SAP ECC based on SAP Gateway
- Public repeatable setup of the demo based on :
 - Public Azure Raspberry emulator
 - Public SAP demo system
 - Azure IoT Hub and Logic App
- Basic modification of existing Azure IoT tutorial to demonstrate SAP connectivity

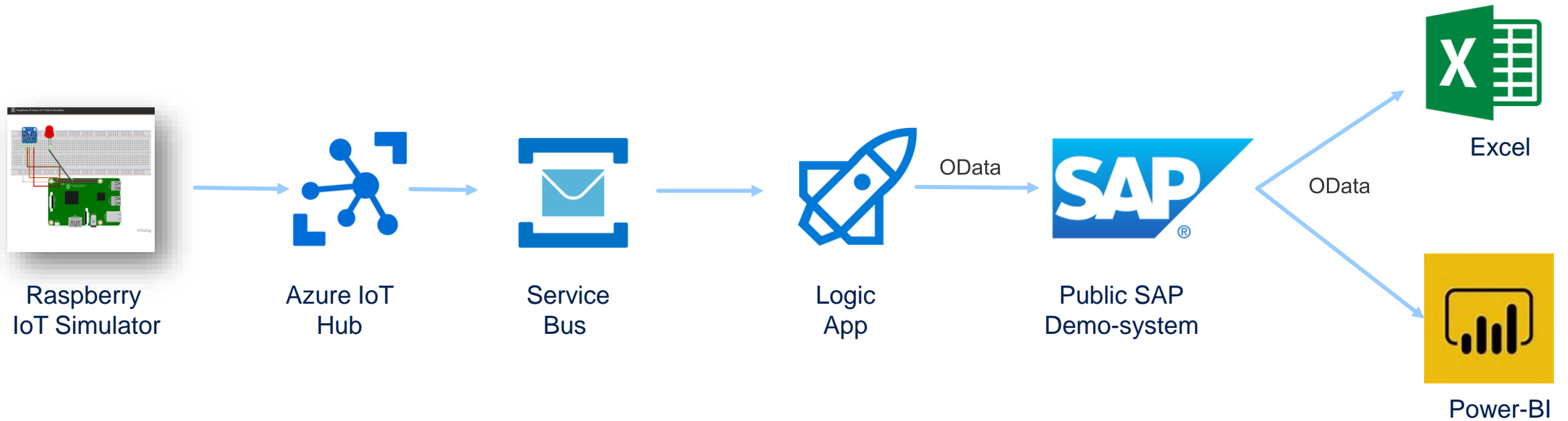


Demo scenario and basic storyline

- Imagine the Raspberry Emulator as an IoT device used in shipping of heat-sensitive vaccines
- If the temperature is measured above 30°, the vaccine is damaged and must be replaced
- To replace the vaccine, a sales order is automatically created in SAP ECC in case the measured temperature is above 30°
- For excel users the current sales order SAP ECC will be displayed in Excel



Demo scenario



Demo implementation

GEWINNEN GEMEINSAM
EINFACH
MIT PARTNERN

DAS OCP PRINZIP


Demo implementation

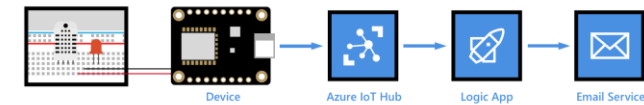


Demo preparation: Raspberry PI emulator and IoT Hub

- Implement Raspberry PI emulator tutorial ([Link](#))
- Implement IoT remote monitoring and notifications with Azure Logic Apps connecting your IoT hub and mailbox tutorial ([Link](#))
 - Stop at the paragraph “Configure the logic app trigger”

IoT remote monitoring and notifications with Azure Logic Apps connecting your IoT hub and mailbox

04/19/2019 • 8 minutes to read • Contributors  all



Note

Before you start this tutorial, complete the [Raspberry Pi online simulator](#) tutorial or one of the device tutorials; for example, [Raspberry Pi with node.js](#). In these articles, you set up your Azure IoT device and IoT hub, and you deploy a sample application to run on your device. The application sends collected sensor data to your IoT hub.

[Azure Logic Apps](#) can help you connect your IoT hub to other services and across various protocols. A logic app can be sequenced using built-in connectors. This solution for IoT monitoring scenarios can initiate logic app workflows.

Connect Raspberry Pi online simulator to Azure IoT Hub (Node.js)

04/11/2018 • 7 minutes to read • Contributors  all

Raspberry Pi web simulator ▾

In this tutorial, you begin by learning the basics of working with Raspberry Pi online simulator. You then learn how to seamlessly connect the Pi simulator to the cloud by using [Azure IoT Hub](#).

If you have physical devices, visit [Connect Raspberry Pi to Azure IoT Hub](#) to get started.

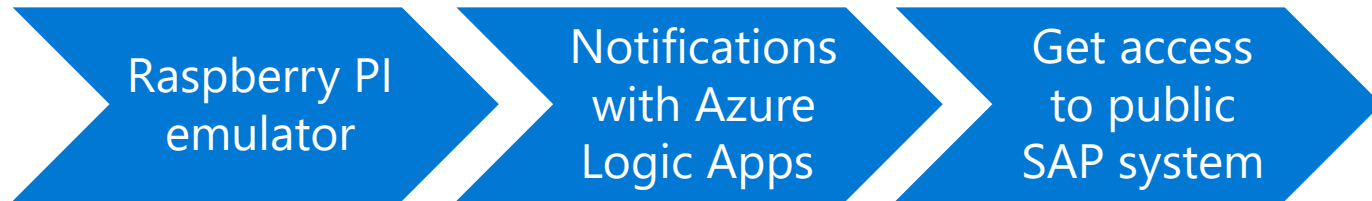


START RASPBERRY PI SIMULATOR


What you do

- Learn the basics of Raspberry Pi online simulator.
- Create an IoT hub.
- Register a device for Pi in your IoT hub.
- Run a sample application on Pi to send simulated sensor data to your IoT hub.

Demo implementation



Demo preparation: Access to public SAP demo system

- Get access to SAP Gateway demo system([Link](#))
- Get familiar with [GWSampleBasic Odata Service \(Visual-Code REST Client\)](#)
- Sample code: 

REST_CLIENT_SAP_ODATA.txt

```
# sapes5.sapdevcenter.com
# Authorization: Basic [REDACTED]

# @name login
GET https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/$metadata HTTP/1.1
Authorization: Basic [REDACTED]
X-CSRF-Token: fetch

@authToken = {{login.response.headers.X-CSRF-Token}}

# @name salesOrder
POST https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/SalesOrderSet HTTP/1.1
Authorization: B [REDACTED]
X-CSRF-Token: {{authToken}}
Content-Type: application/json
```

Technical Articles

Andre Fischer 

December 5, 2017 4 minute read

New SAP Gateway Demo System available

27 Likes 21,802 Views 98 Comments

Updates

- 23.01.2018 – Show how to enter a TCODE
- 29.05.2018 – Added the section “*What you can’t do with the SAP Gateway Demo System*”
- 26.09.2018 – Added links to the \$metadata documents of the SAP Fiori Reference Apps
- 10.05.2019 – Added link to reset your password

The beta phase for our new SAP Gateway Demo system that was announced by Former Member in his blog [Netweaver Gateway Demo – ES5 now in Beta!](#) is now over and we can thus announce the availability of our new “production system” ES5.

As Jonathan already wrote we look forward to your feedback and encourage you to use our new system since ES4 will be shutdown in the future.

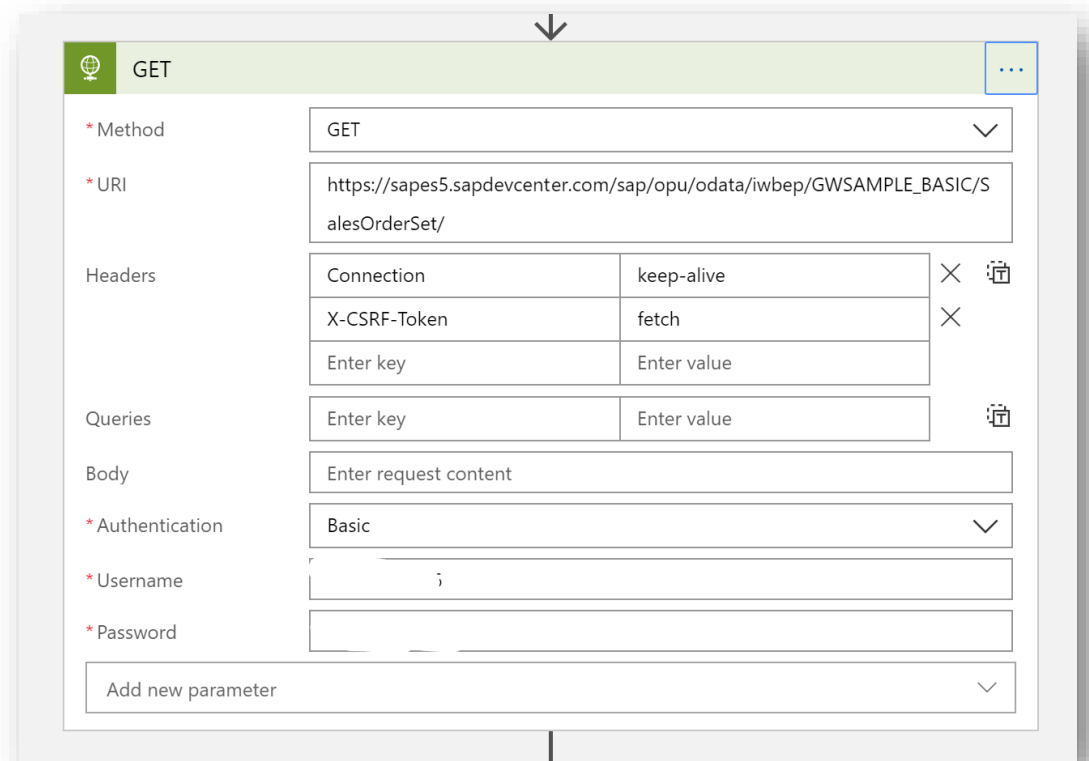
What you can’t do with the SAP Gateway Demo System

Demo implementation



Demo implementation: Logic App 1/2

- Finish the [tutorial](#) until : Configure the logic app trigger / 5. Create a service bus connection.
- Instead of creating an STMP connection, create a HTTP GET Action
- URI: [Link](#)
- Header parameter:
 - Connection: keep-alive
 - X-CSRF-Token: fetch
 - Authentication: Basic
 - User/Pwd: Credentials for SAP demo system



The screenshot shows the configuration for a 'GET' action in a Logic App. The interface includes a title bar with a globe icon and the text 'GET'. Below this, there are several sections for configuring the HTTP request:

- * Method:** A dropdown menu set to 'GET'.
- * URI:** A text box containing the URL: `https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/SalesOrderSet/`.
- Headers:** A table with two columns for key and value. It contains two entries: 'Connection' with value 'keep-alive' and 'X-CSRF-Token' with value 'fetch'. There are 'X' and 'Add' icons to the right of the table.
- Queries:** A table with two columns for key and value, both labeled 'Enter key' and 'Enter value' respectively. There is an 'Add' icon to the right.
- Body:** A text box labeled 'Enter request content'.
- * Authentication:** A dropdown menu set to 'Basic'.
- * Username:** A text box with a semicolon (;) as a placeholder.
- * Password:** A text box.
- Add new parameter:** A button at the bottom with a dropdown arrow.

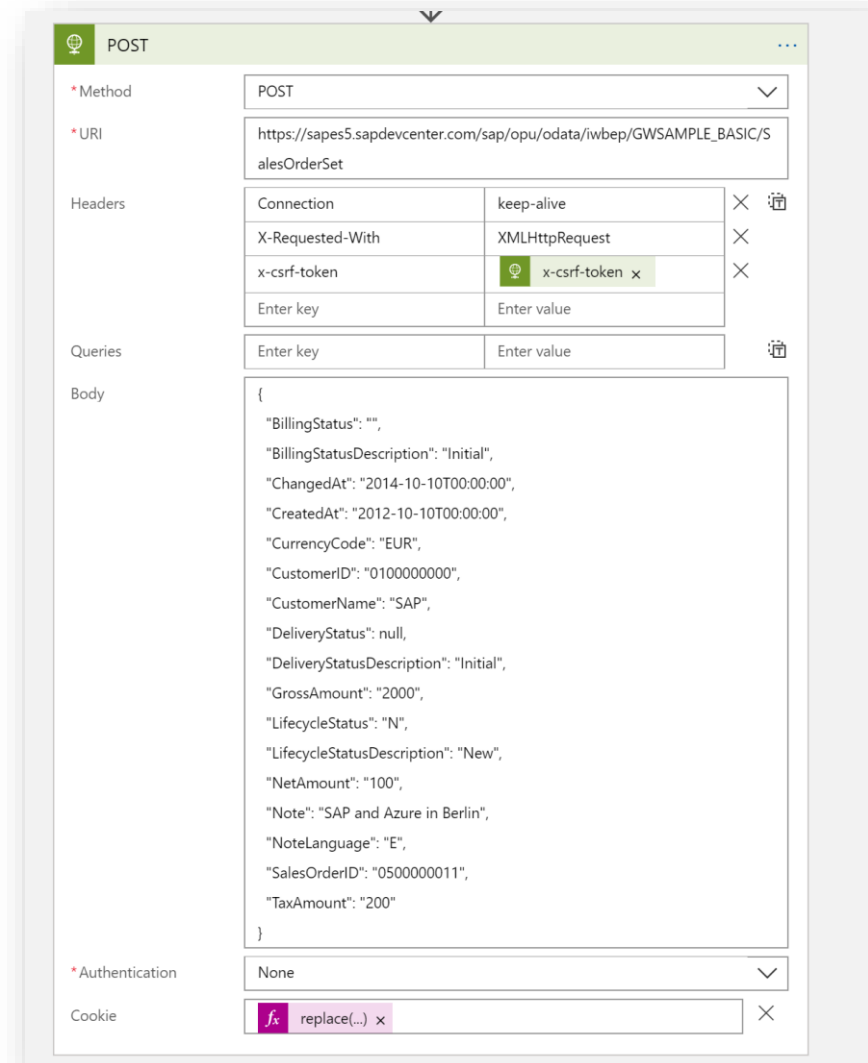
Demo implementation: Logic App 2/2

- Append a HTTP POST Action to the GET-request
- The logic app creates an SAP sales order via the Odata Gateway



LOGIC_APP_SAP_ODATA_POST.txt

- Sample code for Logic App:
- Important parameters:
 - X-csrf-token: @outputs('GET')['headers']['x-csrf-token']
 - Function for cookie:
replace(outputs('GET')['headers']['Set-cookie'], ',', ';')
- Further details in [Bartosz's Blog](#)

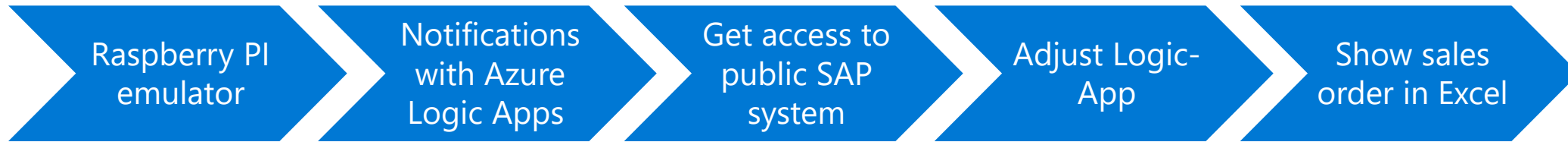


The screenshot displays the configuration for a POST action in a Logic App. The interface includes sections for Method, URI, Headers, Queries, Body, Authentication, and Cookie.

- Method:** POST
- URI:** https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/SalesOrderSet
- Headers:**
 - Connection: keep-alive
 - X-Requested-With: XMLHttpRequest
 - x-csrf-token: x-csrf-token (with a value input field)
- Queries:** (Empty table with Enter key and Enter value columns)
- Body:**

```
{  "BillingStatus": "",  "BillingStatusDescription": "Initial",  "ChangedAt": "2014-10-10T00:00:00",  "CreatedAt": "2012-10-10T00:00:00",  "CurrencyCode": "EUR",  "CustomerID": "0100000000",  "CustomerName": "SAP",  "DeliveryStatus": null,  "DeliveryStatusDescription": "Initial",  "GrossAmount": "2000",  "LifecycleStatus": "N",  "LifecycleStatusDescription": "New",  "NetAmount": "100",  "Note": "SAP and Azure in Berlin",  "NoteLanguage": "E",  "SalesOrderID": "0500000011",  "TaxAmount": "200"}
```
- Authentication:** None
- Cookie:** replace(...)

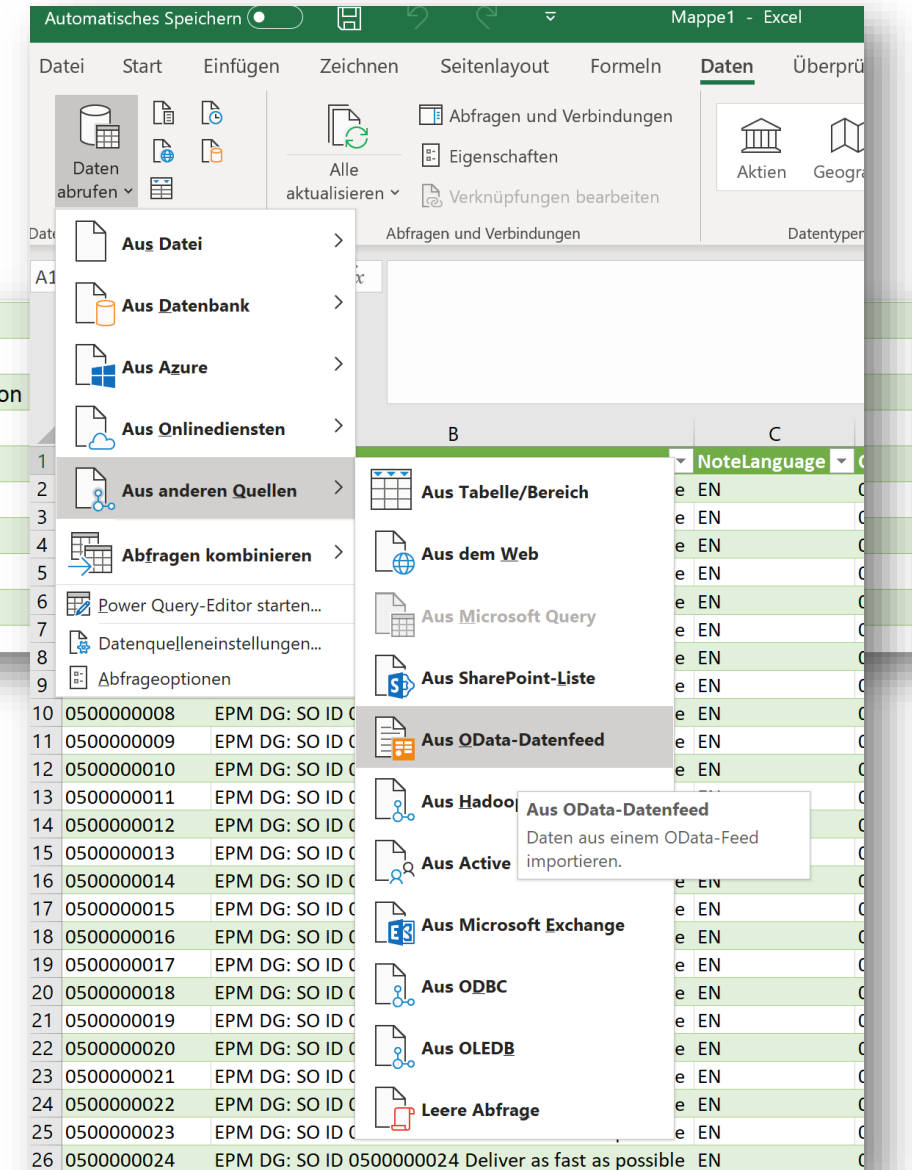
Demo implementation



Demo implementation: Show sales order in Excel

- Connect to Odata datssource
- URL: [Link](#)
- Display results:

0500000494	EPM DG: SO ID 0500000494 Deliver as fast as possible	EN	0100000043	Danish Fish Trading Company
0500000495	EPM DG: SO ID 0500000495 Deliver as fast as possible	EN	0100000044	Sorali
0500000496	EPM DG: SO ID 0500000496 Deliver as fast as possible	EN	0100000036	African Gold And Diamond Corporation
0500000497	EPM DG: SO ID 0500000497 Deliver as fast as possible	EN	0100000042	Siwusha
0500000498	EPM DG: SO ID 0500000498 Deliver as fast as possible	EN	0100000037	PicoBit
0500000499	EPM DG: SO ID 0500000499 Deliver as fast as possible	EN	0100000038	Bionic Research Lab
0500000500	SAP and Azure in Berlin	EN	0100000000	SAP
0500000501	SAP and Azure in Berlin	EN	0100000000	SAP
0500000502	SAP and Azure in Berlin	EN	0100000000	SAP
0500000503	SAP and Azure in Berlin	EN	0100000000	SAP



Run the demo

GEWINNEN GEMEINSAM
EINFACH
MIT PARTNERN

DAS OCP PRINZIP

Run the demo

Temperature in IoT simulator is above 30°

Stop Reset

```
>
Sending message: {"messageId":868,"deviceId":"Raspberry Pi Web Client","temperature":27.94813052000979,"humidity":68.5
>
Sending message: {"messageId":869,"deviceId":"Raspberry Pi Web Client","temperature":21.686533571725395,"humidity":75.
>
Sending message: {"messageId":870,"deviceId":"Raspberry Pi Web Client","temperature":31.179797765358543,"humidity":73.
>
Sending message: {"messageId":871,"deviceId":"Raspberry Pi Web Client","temperature":31.135097640075866,"humidity":75.
>
Sending message: {"messageId":872,"deviceId":"Raspberry Pi Web Client","temperature":21.88735663722798,"humidity":65.2
> []
```



Logic app creates new sales order

When a message is received in a queue (auto-complete) 10s

GET 2s

POST 1s

INPUTS

Method

POST

URI

https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/SalesOrderSet

Body

```
{
  "BillingStatus": "",
  "BillingStatusDescription": "Initial",
  "ChangedAt": "2014-10-10T00:00:00",
  "CreatedAt": "2012-10-10T00:00:00",
  "CurrencyCode": "EUR",
  "CustomerID": "0100000000",
  "CustomerName": "SAP"
}
```

Headers

```
"Connection": "keep-alive",
"X-Requested-With": "XMLHttpRequest",
"x-csrf-token": "TZub82bjxdYLD_7RGZWu8Q=="
```

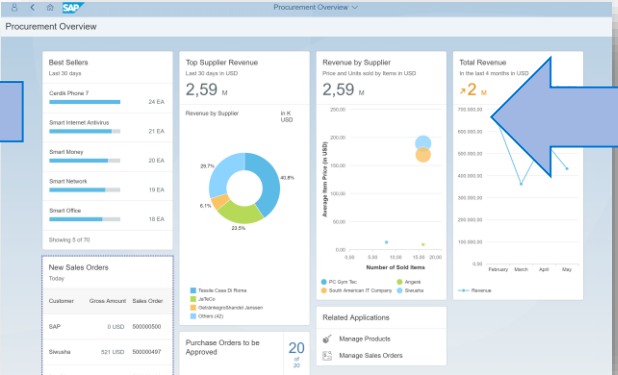
OUTPUTS

Status code

201



Sales order created in SAP



Display created sales order in Excel

0500000494	EPM DG: SO ID 0500000494 Deliver as fast as possible	EN
0500000495	EPM DG: SO ID 0500000495 Deliver as fast as possible	EN
0500000496	EPM DG: SO ID 0500000496 Deliver as fast as possible	EN
0500000497	EPM DG: SO ID 0500000497 Deliver as fast as possible	EN
0500000498	EPM DG: SO ID 0500000498 Deliver as fast as possible	EN
0500000499	EPM DG: SO ID 0500000499 Deliver as fast as possible	EN
0500000500	SAP and Azure in Berlin	EN
0500000501	SAP and Azure in Berlin	EN
0500000502	SAP and Azure in Berlin	EN
0500000503	SAP and Azure in Berlin	EN

Questions?



Thank You!

GEWINNEN GEMEINSAM EINFACH MIT PARTNERN

DAS OCP PRINZIP

Roman Broich

Cloud Solution Architect

[linkedin.com/in/roman-broich/](https://www.linkedin.com/in/roman-broich/)