

Contents

What you will learn	1
Prerequisites	1
Simulator Setup.....	2
Simulator Test.....	5
Takeaway	13

What you will learn: Simulator

Day 1 of 5. Topics covered :

SAP business objects simulator

- Publish SAP simulated events that represents mocked JSON structures
- Publish events from 5 different SAP object- Salesorders, Business Partners, Chart of Accounts, Material Master, and Notifications
- Test Simulated SAP events being published and subscribed too using the AEM broker and the SAP Ui5 dashboard.

Prerequisites

1: BTP subaccount with a developer space enabled in Cloud Foundry

2: CloudFoundry CLI installation (Only needed if upload doesn't work)

To start with, we will be installing the CloudFoundry CLI for the deployment process. Follow the steps mentioned over here [Installing the CF CLI](#) for detailed instructions on this.

3 : Downloading the deployable artifacts

Download the following files artefact files and save them in the same directory:

- capm-erp-simulation-exec.jar : <https://github.com/SolaceLabs/aem-sap-integration/blob/main/deployable/capm-erp-simulation-exec.jar>

- manifest.yml : <https://github.com/SolaceLabs/aem-sap-integration/blob/main/deployable/manifest.yml>

SAP Simulator setup

The SAP Cloud Application Programming Model (CAP) is a framework of languages, libraries, and tools for building enterprise-grade services and applications. It guides developers along a 'golden path' of proven best practices and a great wealth of out-of-the-box solutions to recurring tasks. CAP-based projects benefit from a primary focus on domain. Instead of delving into overly technical disciplines, we focus on accelerated development and safeguarding investments in a world of rapidly changing cloud technologies.

For more information on SAP CAP, you can refer to the link : [SAP Cloud Application Programming Model](#)

To showcase the integration capability of SAP CAP and AEM, we have created a CAP based Java microservice which will publish different SAP business object events into your AEM instance. This application can be deployed in your SAP CloudFoundry space.

1 : Identify CF Domain address

In order to deploy the simulator to your CloudFoundry space, you need to identify the domain address which is a part of the API endpoint.

- Navigate to your SAP BTP Sub account Overview page
- Copy the specified section of the API Endpoint in the Cloud Foundry Environment as shown below :

The screenshot shows the SAP BTP Cockpit interface. On the left, there's a sidebar with various menu items: Overview (highlighted with a red circle '1'), Services, Cloud Foundry, HTML Applications, Connectivity, Security, Entitlements, and Usage Analytics. The main content area is titled "Subaccount: trial - Overview". It has tabs for General, Cloud Foundry Environment, Kyma Environment, and Entitlements. Under General, it shows 75 Entitlements and 2 Instances and Subscriptions. In the Cloud Foundry Environment section, the API Endpoint is listed as "https://api.cl.us10-001.hana.ondemand.com" (highlighted with a red circle '2'). Below that, Org Name, Org ID, and Org Memory Limit (4,096MB) are shown. There's also a "Manage environment instance" button and a "Disable Cloud Foundry" button. At the bottom right, there's a "Create Space" button. The overall interface is light-colored with blue and grey accents.

2 : Update the manifest file

- Open the manifest.yml file which you downloaded earlier in a text editor
- Replace the placeholder text {API_ENDPOINT} on line number 12 with the value copied from the API Endpoint

```
1  # Generated manifest.yml based on template version 0.1.0
2  # appName = capm-erp-simulation
3  # language=java
4  # multitenancy=false
5  ---
6  applications:
7  # -----
8  # Backend Service
9  # -----
10 - name: capm-erp-simulation
11   routes:
12     - route: capm-erp-simulation-aem-workshop.cfapps.{API_ENDPOINT}
13       path: srv/target/capm-erp-simulation-exec.jar
14       memory: 1G
15       disk_quota: 512M
16   env:
17     JBP_CONFIG_SPRING_AUTO_RECONFIGURATION: '{ enabled: false }'
18   buildpack: sap_java_buildpack
19   # random-route: true
20
21
```

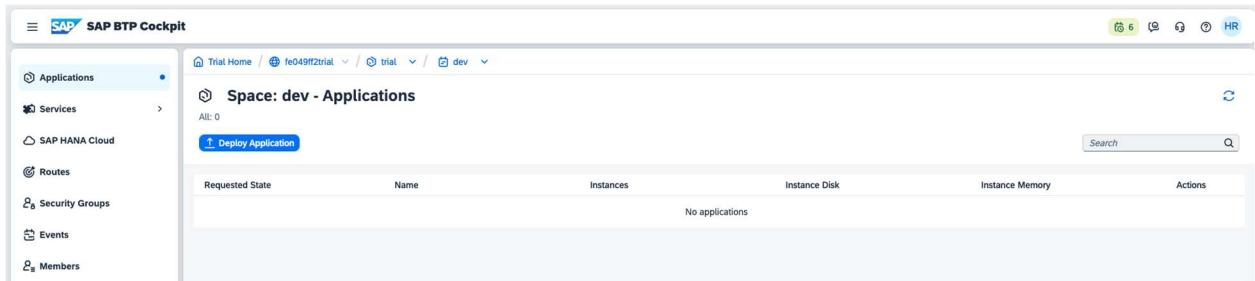
- After replacing your manifest file should look like this :

```
1 # Generated manifest.yml based on template version 0.1.0
2 # appName = capm-erp-simulation
3 # language=java
4 # multitenancy=false
5 ---
6 applications:
7 # -----
8 # Backend Service
9 #
10 - name: capm-erp-simulation
11   routes:
12     - route: capm-erp-simulation-aem-workshop.cfapps.us10-001.hana.ondemand.com
13       path: srv/target/capm-erp-simulation-exec.jar
14       memory: 1G
15       disk_quota: 512M
16     env:
17       JBP_CONFIG_SPRING_AUTO_RECONFIGURATION: '{ enabled: false }'
18     buildpack: sap_java_buildpack
19   # random-route: true
20
21
```

Save and close the file.

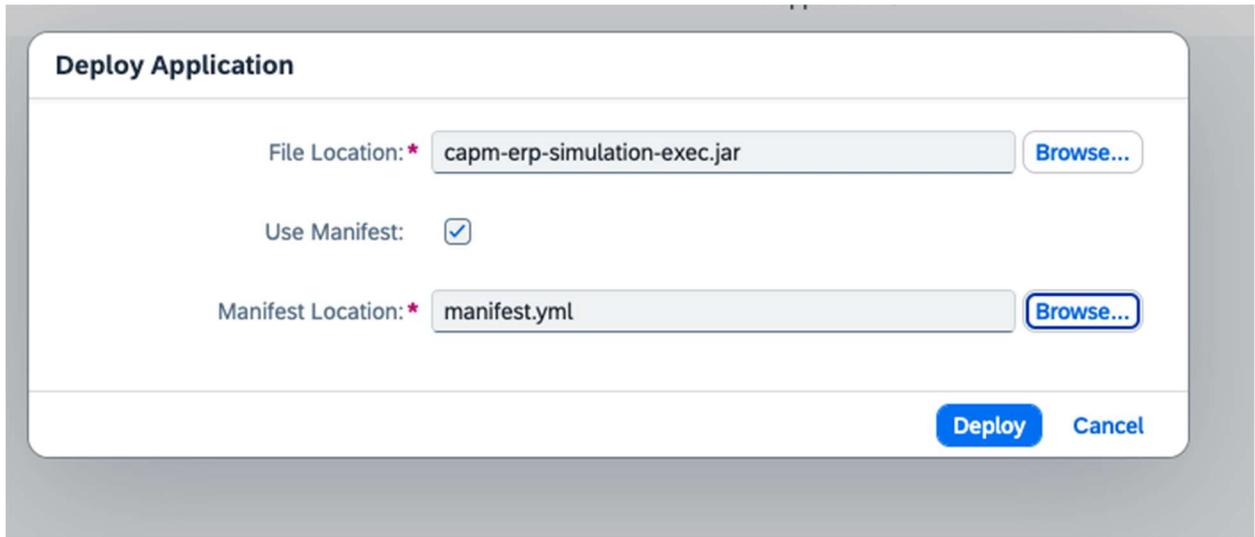
3 : Deploying the SAP Simulator application

- Navigate to the CloudFoundry space where you want to deploy the application and click on the **Deploy Application** button as below :



The screenshot shows the SAP BTP Cockpit interface. The left sidebar has navigation links for Applications, Services, SAP HANA Cloud, Routes, Security Groups, Events, and Members. The main area is titled 'Space: dev - Applications' and shows a table with columns: Requested State, Name, Instances, Instance Disk, Instance Memory, and Actions. A note at the bottom of the table says 'No applications'. At the top of the main area, there is a blue button labeled 'Deploy Application'. The top right corner of the screen shows some status icons.

- A **Deploy Application** modal window will be displayed where you can browse and select the **capm-erp-simulation-exec.jar** and **manifest.yml** as below



- Click on the deploy button **Note : this action will take some time to completely execute as it uploads the jar deployable and also start the application.**
- Once the upload is completely executed, you should be able to see the application deployed and running as below :

The screenshot shows the SAP BTP Cockpit interface. On the left, there's a sidebar with 'Applications', 'Services', 'SAP HANA Cloud', 'Routes', 'Security Groups', 'Events', and 'Members'. The main area is titled 'Space: dev - Applications' and shows a table with one row. The table columns are 'Requested State', 'Name', 'Instances', 'Instance Disk', 'Instance Memory', and 'Actions'. The row contains 'Started', 'capm-erp-simulation', '1/1', '512 MB', '1024 MB', and a set of small icons for more actions.

CMD Line Deployment Option

Login to CloudFoundry space

You can log in to the SAP CloudFoundry space in your account as below :

Use the command : cf login to log in, which will prompt for your SAP login credentials.

Once authenticated, the details of the default cloudfoundry space will be displayed.

Deploying the SAP Simulator application

Navigate to the directory where the above deployable artifact files are saved.

Run the command cf push --random-route which will upload the jar file and use the manifest.yml for properties. Note : this command will take some time to completely execute as it uploads the jar deployable and also start the application.

Once the command is completely executed, run the command cf apps to view a listing of the apps in your cloudfoundry space

Verify that the app capm-erp-simulation is deployed and started

Testing the Simulator

1 : Accessing the SAP Simulator application

- Navigate to the Cloud Foundry environment in your SAP BTP Cockpit
- You should see a screen like below

The screenshot shows the SAP BTP Cockpit interface for the 'Space: hari_cf_space - Applications' section. It displays a single application named 'capm-erp-simulation' which is currently 'Started'. The application has 1 instance, 512 MB of disk space, and 1024 MB of memory allocated. There are buttons for 'Deploy Application', 'Search', and 'Actions'.

- Click on the application name : **capm-erp-simulation** and enter the application overview screen.

The screenshot shows the 'Application: capm-erp-simulation - Overview' page. At the top, there are buttons for 'Started', 'Restart', 'Start', 'Stop', 'Instance', 'Delete', and 'Actions'. Below this, the 'Application Routes' section shows the URL <https://capm-erp-simulation.cfapps.us10-001.hana.ondemand.com>. The 'Application Information' section provides details about the instance: 1 instance, package uploaded on 27 Sept 2023, buildpack 'sap_java_buildpack', and stack 'Cloud Foundry Linux-based filesystem (Ubuntu 22.04) (cflinuxfs4)'. The 'Instance Details' section shows memory and disk usage. The 'Instances' section lists one instance as 'RUNNING'.

- Click on the application route as highlighted below. Note : this route url will differ from for different SAP BTP accounts.

Rapid Pilot Day 1

The screenshot shows the SAP Cloud Foundry Application Overview page for the application 'capm-erp-simulation'. At the top, there's a green 'Started' status bar with buttons for 'Restart', 'Start', 'Stop', '+ Instance', 'Instance', and 'Delete'. Below this, the 'Application Routes' section displays the URL <https://capm-erp-simulation.cfapps.us10-001.hana.ondemand.com>. The 'Application Information' section provides details about the instance: 1 instance, package uploaded on 27 Sept 2023, buildpack 'sap_java_buildpack', and stack 'Cloud Foundry Linux-based filesystem (Ubuntu 22.04) (cflinuxfs4)'. The 'Instance Details' section shows memory usage (1024 MB available, 3072 MB total), disk usage (512 MB), and a 'Change Instance Details' button. The 'Instances' section lists one instance as 'RUNNING' since 29 Sept 2023, 15:48:21, using 0.6% CPU, 181.2 MB memory, and 474.5 MB disk.

2 : Connecting to SAP AEM and running the simulator

- As you click on the above application route url, you will be redirected to the simulator screen as below

The screenshot shows the SAP Advanced Event Mesh - ERP Simulator interface. The header includes the 'solace.' logo, the title 'Welcome to the Advanced Event Mesh - ERP Simulator', and the SAP logo. Below the header is a form with fields for 'Host URL', 'VPN Name', 'Username', and 'Password', with a 'Connect to broker' button.

Here you can connect to your SAP AEM instance to publish events.

As long as both of your SAP AEM services are connected to the event mesh, messages will flow freely between the two of them. Due to this intelligent routing, you can connect the simulator to either of your AEM services created earlier.

- The connection parameters for the simulator can be captured from below:

The screenshot shows the SAP Mission Control interface for 'MontrealBroker-10.1'. The 'Status' tab is selected, and the 'Connect' button is highlighted with a red box. Other tabs include 'Manage', 'Monitoring', 'Configuration', and 'Try Me!'. The main area displays 'Messaging Activity' with three circular progress bars: 'Active Connections' at 20%, 'Guaranteed Messaging Endpoints' at 53%, and 'Queue Usage' at 1%. Below these are tables for 'Availability and Versioning' showing counts for AMQP, MQTT, SMF, REST, and Web protocols, and for 'Queues' and 'Topic Endpoints'.

Rapid Pilot Day 1

The screenshot shows the MontrealBroker-10.1 interface. At the top, there's a navigation bar with links for Status, Connect, Manage, Monitoring, Configuration, Try Me!, Open Broker Manager, and more. Below the navigation bar, a section titled "Connect Using a Supported Client Library" lists various client libraries. A dropdown menu labeled "View by: Language" is highlighted with a red box and a circled number 1. A "Try Me!" button is also visible. The list of libraries includes:

- Java (Solace Java, Solace JMS over SMF, Paho over MQTT, QPID JMS 1.1 over AMQP, QPID JMS 2.0 over AMQP)
- C (Solace C API over SMF, Paho over MQTT)
- Python (Solace Python API over SMF, Paho over MQTT)
- Go (Solace Go API over SMF)
- JavaScript (Solace Javascript API over SMF, Paho over MQTT)
- Node.js (Solace Node.js API over SMF, AMPQPI0 Open Source over AMQP)
- .NET (Solace .NET API over SMF, Paho over MQTT)
- Spring (Spring Cloud Stream, Spring Boot)

A red box highlights the "Spring" entry, which is circled with a number 2. Below this, a table shows the library and protocol mapping:

Library	Protocol
Spring Cloud Stream	SMF
Spring Boot Java API	SMF

A red box highlights the "Spring Boot Java API" entry, which is circled with a number 3. To the right of the table, there are two "Get Started" buttons, one for each row, both highlighted with red boxes and circled with a number 4.

Enter the appropriate value as specified below :

- Host URL : Public Endpoint
- VPN Name : Message VPN
- Username : Username
- Password : Password

Rapid Pilot Day 1

Select another API

Spring

[API Docs](#) [Download](#)

Spring Boot Java API

1 Get API **2 Connect to Service** **3 Learn More**

Here are a few easy ways to get the Get the API. If your environment differs then adjust the build instructions appropriately.

Get the API: I already have a Maven Spring Boot Project

Add the Solace Java Spring Boot Starter to your POM if using Maven.

```
<dependency>
  <groupId>com.solace.spring.boot</groupId>
  <artifactId>solace-java-spring-boot-starter</artifactId>
  <version>4.1.0</version>
</dependency>
```

Get the API: I already have a Gradle Spring Boot Project

Add the Solace Java Spring Boot Starter to your build.gradle if using Gradle.

```
compile group: 'com.solace.spring.boot', name: 'solace-java-spring-boot-starter'
```

Get the API: Starting from Scratch - Use Spring Initializr to bootstrap my project!

It is recommended to bootstrap your Spring Boot project using [Spring Initializr](#). Spring Initializr will generate your Spring Boot project skeleton for you based on a few simple choices, such as your preferred language, Spring Boot version, and then dependencies you need to build your microservice.

Once you bootstrap your Spring Boot project add the Solace Java Spring Boot Starter as shown in the previous section.

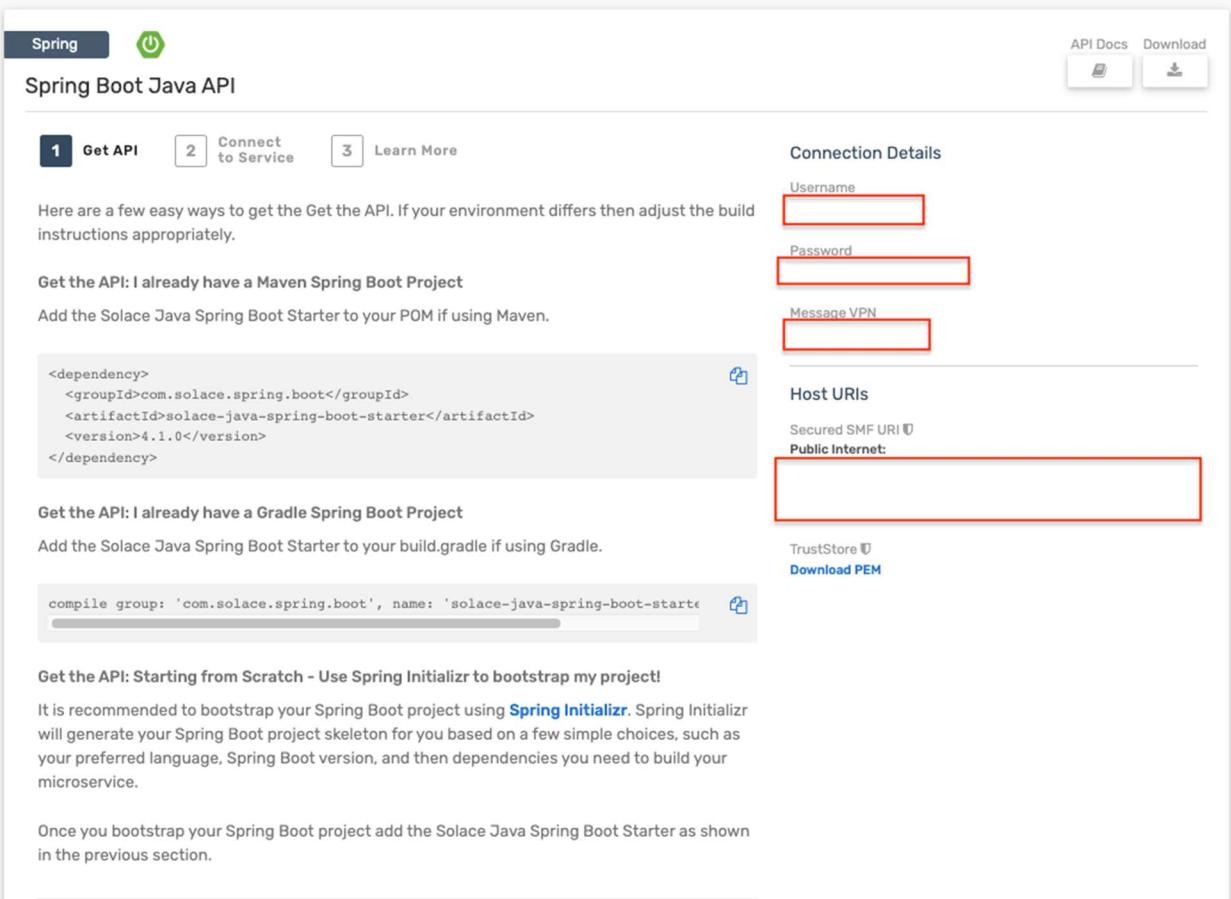
Connection Details

Username:
Password:
Message VPN:

Host URIs

Secured SMF URI:
Public Internet:

[TrustStore](#) [Download PEM](#)



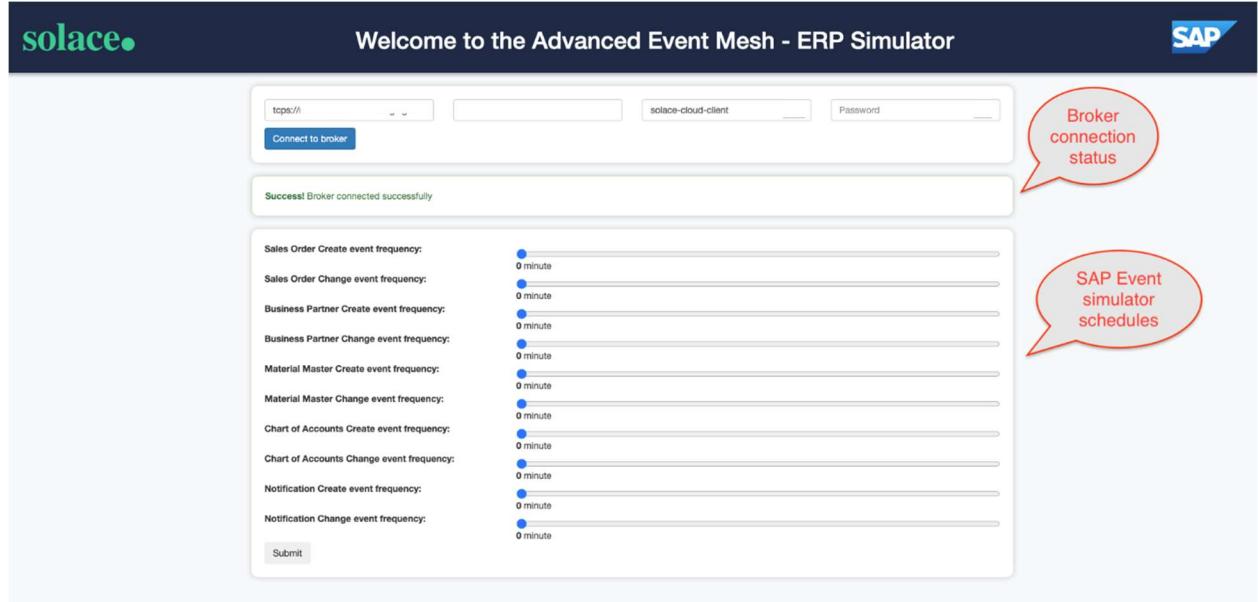
- Once the broker is successfully connected, you will be displayed a screen as below :

solace. **Welcome to the Advanced Event Mesh - ERP Simulator** **SAP**

Broker connection status: Success! Broker connected successfully

Sales Order Create event frequency: 0 minute
Sales Order Change event frequency: 0 minute
Business Partner Create event frequency: 0 minute
Business Partner Change event frequency: 0 minute
Material Master Create event frequency: 0 minute
Material Master Change event frequency: 0 minute
Chart of Accounts Create event frequency: 0 minute
Chart of Accounts Change event frequency: 0 minute
Notification Create event frequency: 0 minute
Notification Change event frequency: 0 minute

SAP Event simulator schedules



- You can choose which events to simulate and its frequency by using the sliders. As you change a schedule, the submit button in the bottom will be enabled.

The screenshot shows the SAP Advanced Event Mesh - ERP Simulator interface. At the top, there are input fields for broker URL (tcp://montreal/broker.messaging.sol), broker name (montrealbroker-10-1), client name (solace-cloud-client), and password. Below these are two buttons: 'Connect to broker' and 'Submit'. A message box displays 'Success! Broker connected successfully'. The main area contains a list of events with frequency sliders:

Event Type	Event Frequency
Sales Order Create	10 minute
Sales Order Change	6 minute
Business Partner Create	15 minute
Business Partner Change	4 minute
Material Master Create	1 minute
Material Master Change	0 minute
Chart of Accounts Create	0 minute
Chart of Accounts Change	0 minute
Notification Create	0 minute
Notification Change	0 minute

A green 'Submit' button is located at the bottom of the slider section.

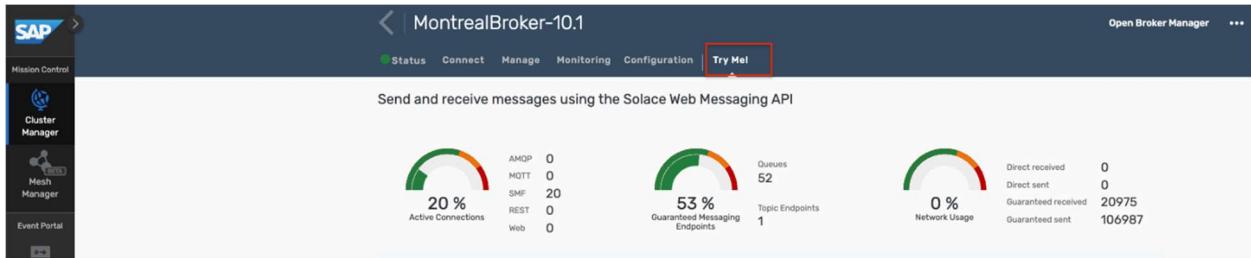
- In case you want to disable any of the events, then pull the slider to **0** and click submit and the event will be disabled immediately.

This screenshot is identical to the one above, but the 'Sales Order Create' event frequency slider has been moved to the 0 minute mark. A red box highlights the 0 minute value to indicate it has been disabled.

3 : Test the incoming events

You can easily test the simulator by using the **Cluster Manager - Try-Me** as below: > aside negative
 > As mentioned earlier due to the intelligent routing in the event mesh, you can connect the simulator and try-me to either of the two SAP AEM services in the event mesh and see the messages flowing freely.

Rapid Pilot Day 1



- Click on the **Connect** button in the **Subscriber** side of the panel as below:

The screenshot shows the MontrealBroker-10.1 interface with two main panels: Publisher and Subscriber. Both panels have "Endpoint Connectivity" dropdowns set to "Public Endpoint (Public Internet)". The Publisher panel shows step 1 "Establish connection" with a "Connect" button highlighted by a red box, and step 2 "Publish" with a topic "try-me" and message "Hello world!". The Subscriber panel shows step 1 "Establish connection" with a "Connect" button highlighted by a red box, and step 2 "Subscribe" with a "Subscribe" button highlighted by a red box. Both panels include "Result" tabs and "EDIT ON CODEPEN" buttons.

- Add topic subscription(s) to view incoming messages on the topic as below:

Publisher

Subscriber

Endpoint Connectivity: Public Endpoint (Public Internet)

1 Establish connection (Connected)

2 Publish

Topic: try-me

Message: Hello world!

Binary Text

Connect Disconnect Show advanced settings

EDIT ON CODEPEN

1 Establish connection (Connected)

2 Subscribe

Add Topic: sap.com/salesorder/create Subscribe

Subscribed Topics: Add a topic subscription to begin receiving messages.

Messages:

Green indicates "Connected"

Topic Subscription(s)

- You can use the below topic structures for different event types :

- **Sales Order :**
 - Create : sap.com/salesorder/create/ >
 - Change : sap.com/salesorder/change/ >
- **Business Partner :**
 - Create : sap.com/businesspartner/create/ >
 - Change : sap.com/businesspartner/change/ >
- **Chart of Accounts :**
 - Create : sap.com/chartofaccounts/create/ >
 - Change : sap.com/chartofaccounts/change/ >
- **Material Master :**
 - Create : sap.com/material/create/ >
 - Change : sap.com/material/change/ >

Rapid Pilot Day 1

- **Notifications :**
 - Create : `sap.com/notification/create/`
 - Change : `sap.com/notification/change/`
- As the simulator publishes events to the broker you should see events appearing in the subscribed topic(s)

Takeaways

- Deploy SAP Simulator in BTP
- Test Events with AEM Try Me Tab

