

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>

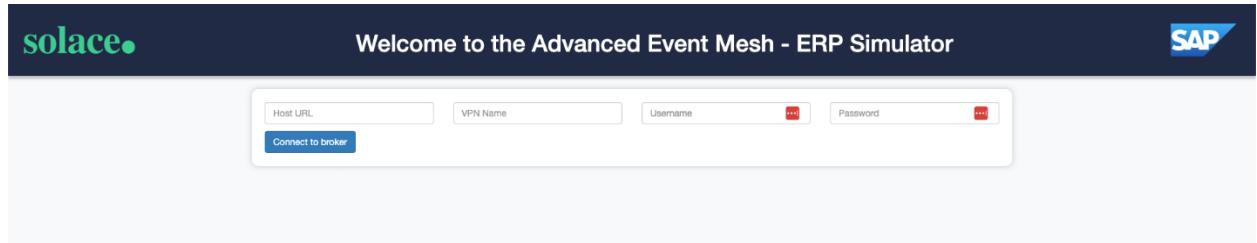
Testing the Simulator

1 : Accessing the SAP Simulator application

- Select 1 workshop attendee to be the Simulator Manager. If other users access the simulator page and modify settings it will overwrite the simulator instance.
- URL Provided in the workshop by presenter

2 : Connecting to SAP AEM and running the simulator

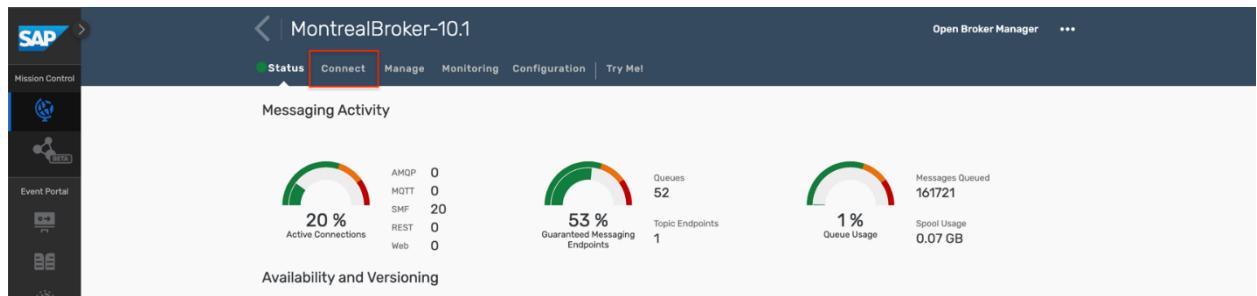
- As you click on the above application route url, you will be redirected to the simulator screen as below



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:



Rapid Pilot Part 2

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. A red circle labeled '1' highlights the 'View by:' dropdown menu, which is set to 'Language'. Below it, a link 'Expand all' is visible.

The main area is titled 'Connect Using a Supported Client Library' and instructs the user to select a connection point and supported client library to start messaging. A red circle labeled '2' highlights the 'Connect with Spring' option, which includes 'Spring Cloud Stream, Spring Boot'.

A table below lists supported libraries and protocols:

Library	Protocol
Spring Cloud Stream	SMF
Spring Boot Java API	SMF

Red circles labeled '3' highlight the 'Spring Cloud Stream' and 'Spring Boot Java API' rows in the table. Red circles labeled '4' highlight two 'Get Started' buttons located at the bottom right of the table.

Enter the appropriate value as specified below :

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

Rapid Pilot Part 2

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 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.

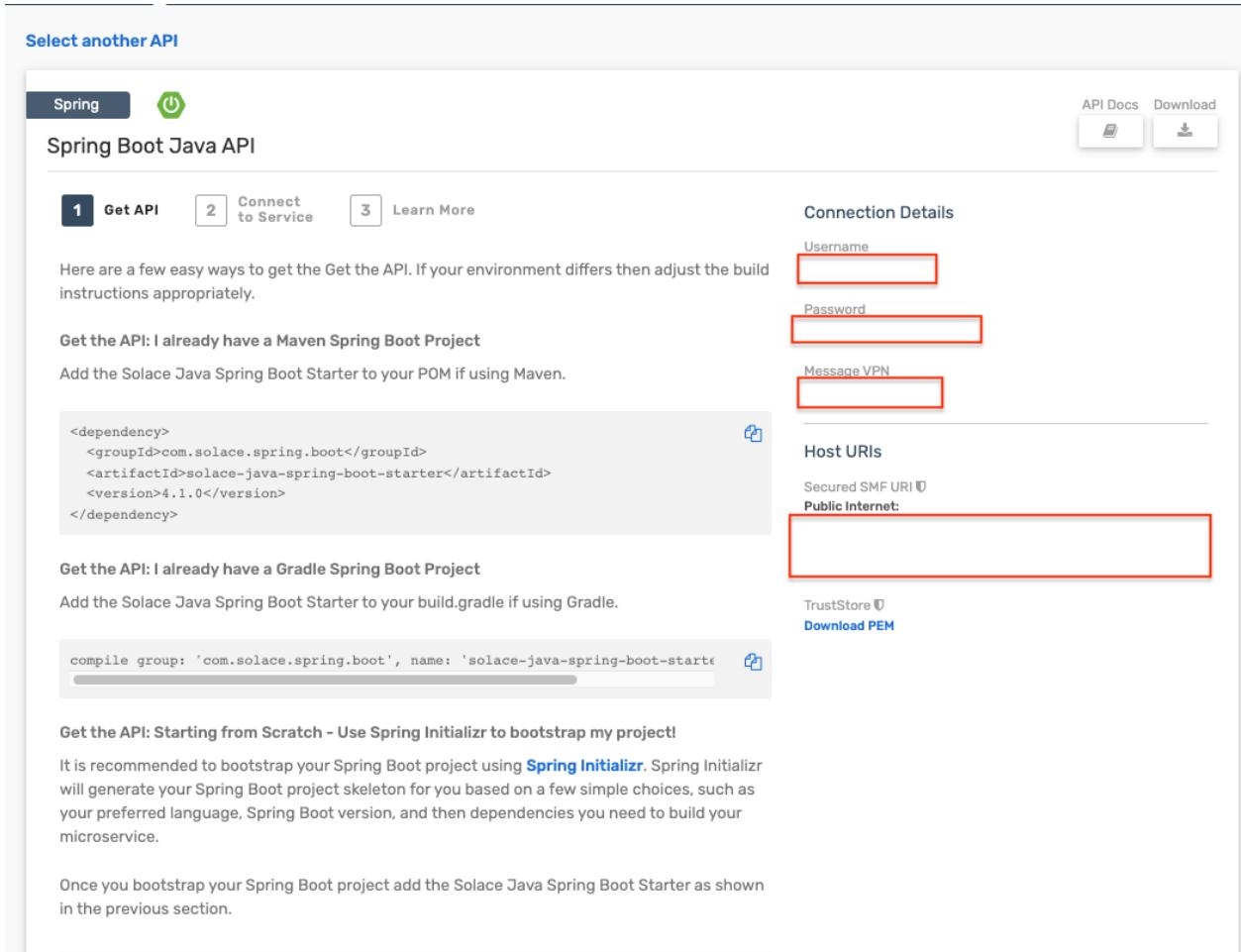
Connection Details

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 :

Welcome to the Advanced Event Mesh - ERP Simulator

Broker connection status

SAP Event simulator schedules

tcp:// solace-cloud-client Password

Connect to broker

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

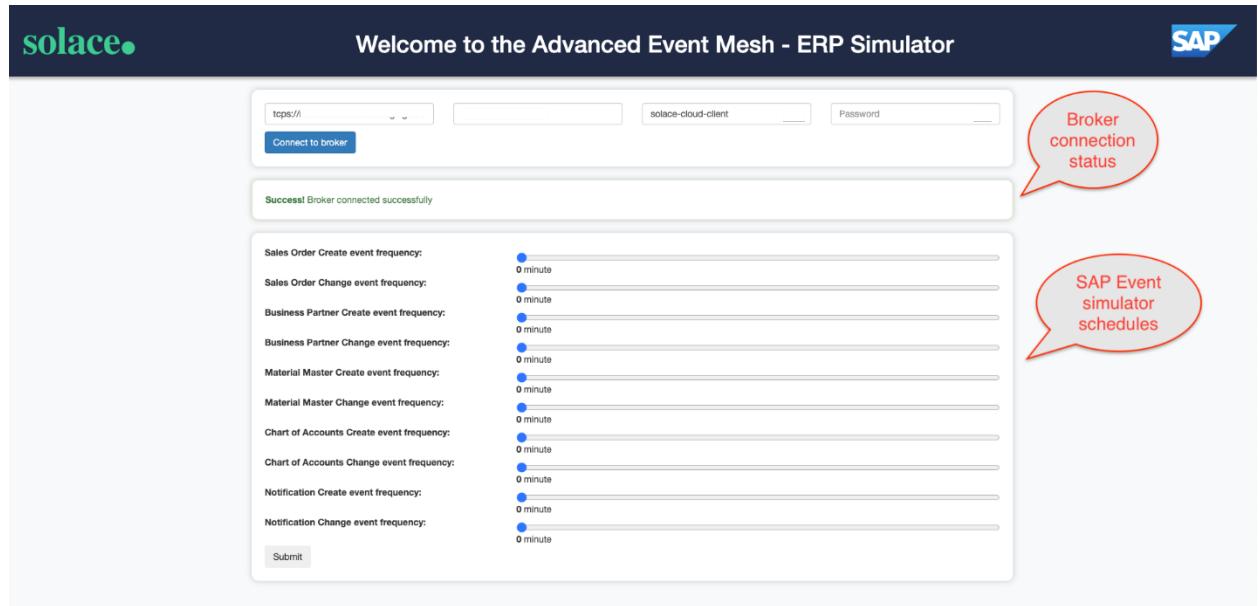
Chart of Accounts Create event frequency: 0 minute

Chart of Accounts Change event frequency: 0 minute

Notification Create event frequency: 0 minute

Notification Change event frequency: 0 minute

Submit



- 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.

Welcome to the Advanced Event Mesh - ERP Simulator

SAP

tcp://montrealbroker.messaging.solace.cloud

montréalbroker-10-1

solace-cloud-client

Password

Connect to broker

Success! Broker connected successfully

Sales Order Create event frequency: 10 minute

Sales Order Change event frequency: 6 minute

Business Partner Create event frequency: 15 minute

Business Partner Change event frequency: 4 minute

Material Master Create event frequency: 1 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

Submit

- 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.

Welcome to the Advanced Event Mesh - ERP Simulator

SAP

tcp://montrealbroker.messaging.solace.cloud

montréalbroker-10-1

solace-cloud-client

Password

Connect to broker

Success! Broker connected successfully

Sales Order Create event frequency: 0 minute

Sales Order Change event frequency: 6 minute

Business Partner Create event frequency: 15 minute

Business Partner Change event frequency: 4 minute

Material Master Create event frequency: 1 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

Submit

3 : Test the incoming events

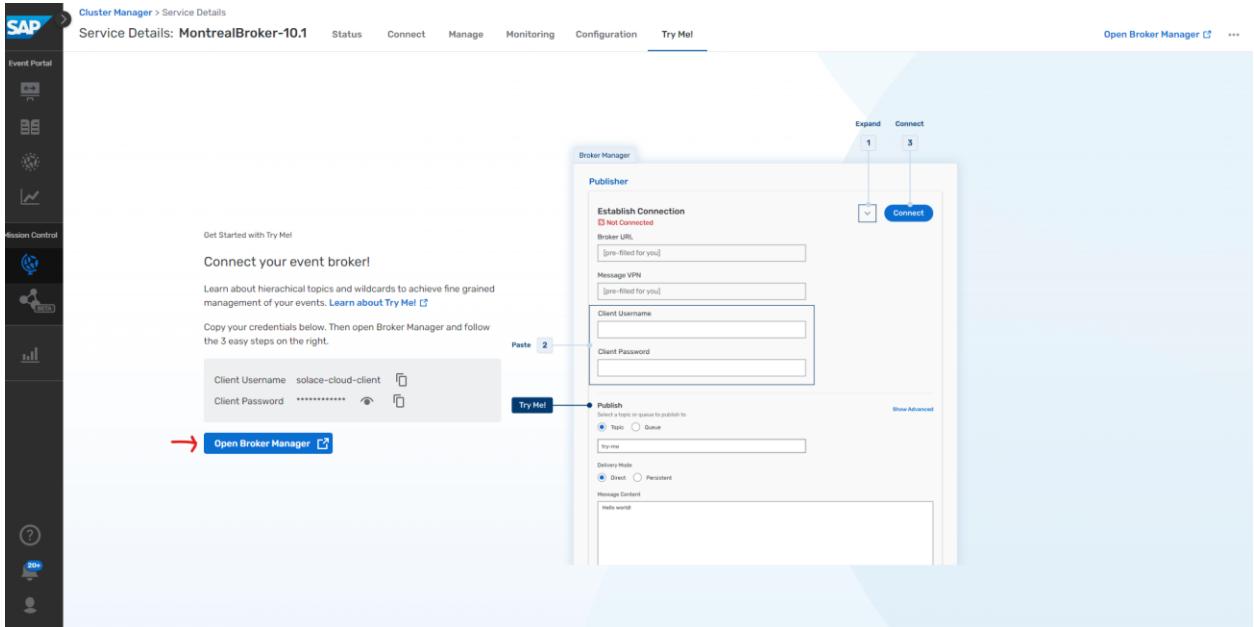
You can easily test the simulator by using the **Cluster Manager - Try-Me** as below:

- > 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.

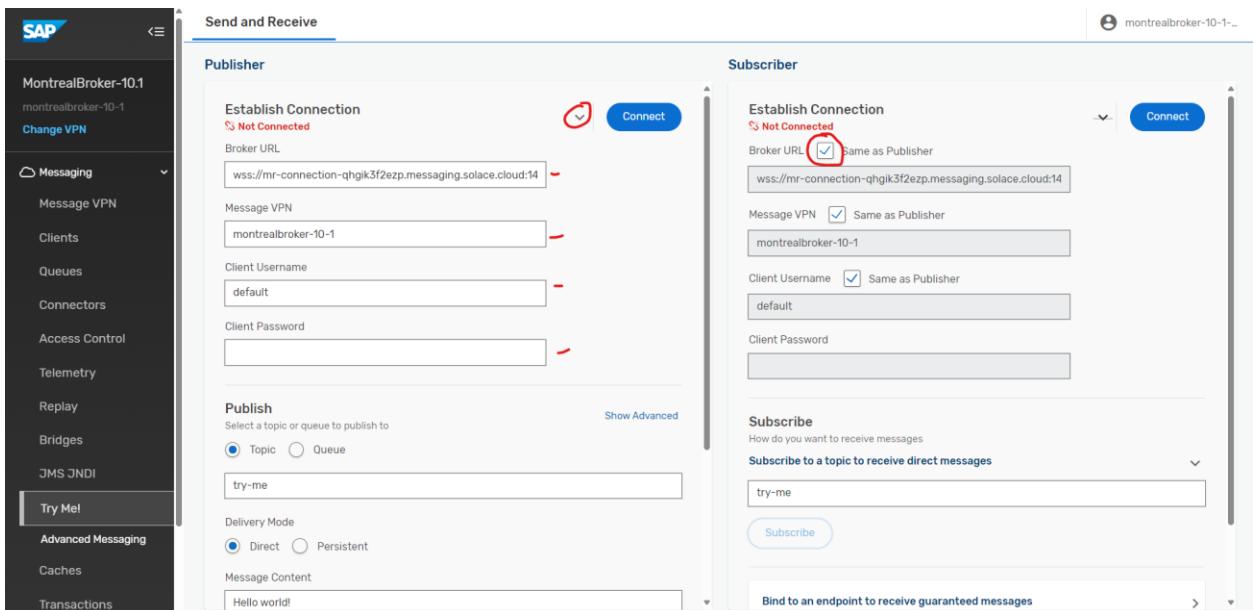
Rapid Pilot Part 2



- Click on the Open Broker Manager button as below :



- Select the drop arrow on the **Publisher** side and enter the credentials of your Brokers. On the **Subscriber** select the "Same as Publisher" box and click **Connect**:



Publisher

Endpoint Connectivity: Public Endpoint (Public Internet)

1 Establish connection Connect Disconnect show advanced settings

2 Publish

Topic: try-me

Message:

```
Hello world!
```

binary text

Publish

Subscriber

Endpoint Connectivity: Public Endpoint (Public Internet)

1 Establish connection Connect Disconnect show advanced settings

2 Subscribe

Add Topic: sap.com/salesorder/create Subscribe

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

Messages:

- 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/>

Rapid Pilot Part 2

- Change : sap.com/material/change/>
- **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

