

Implementation Guide

– *Sales Order Fulfillment* Template App

Contents

0.Prerequisites	2
1.Environment Configuration in Logistics Business Network, Global Track and Trace Option	2
1.1 Enable Cloud Foundry and Create Space	2
1.2 Maintain Roles and Users	4
1.3 Maintain Technical User	5
2.Implement <i>Sales Order Fulfillment</i> Template App.....	8
2.1 Download Template Code.....	8
2.2 Download and Setup Tools	8
2.3 Deploy Model.....	9
2.4 Package and Deploy	10
2.4.1 Login to SAP Cloud Foundry	10
2.4.2 Package and Deploy <i>Sales Order Fulfillment</i> Frontend App	10
2.4.3 Package and Deploy SAP Fiori Launchpad for <i>Sales Order Fulfillment</i>	12
2.4.4 Package and Deploy <i>Sales Order Fulfillment</i> Backend Service	14
2.4.5 Check Deployed Apps on SAP Cloud Foundry Cockpit.....	15
2.4.6 Maintain Destination	15
2.4.7 Maintain Roles and Users	17
2.4.8 Open <i>Sales Order Fulfillment</i> Template App from SAP Fiori Launchpad	18
3. Implement ERP Extractor	19
4. Other Configuration	19
4.1 Obtain Here Map License.....	19
4.2 Maintain Location Master	19
4.3 Maintain Data Contributors	21

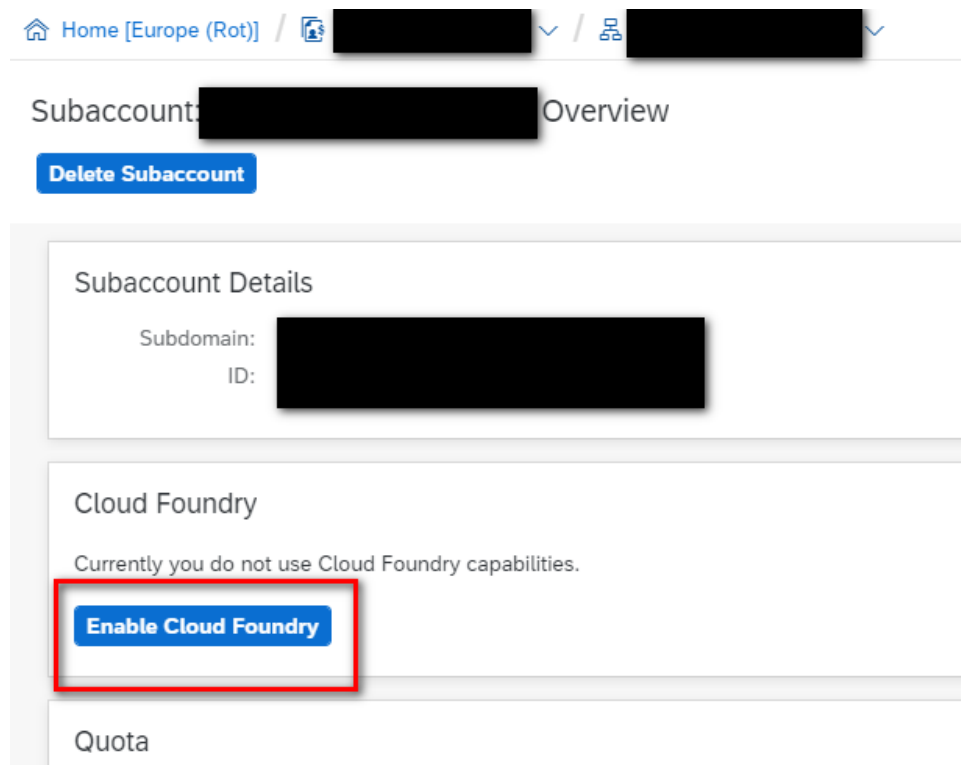
0.Prerequisites

- An SAP Cloud Platform global account with entitlement to the global track and trace option for SAP Logistics Business Network, 1 portal service quota and 10 GB Application Runtime quota
- To integrate with ERP, an SAP ERP or SAP ECC system running on Netweaver 7.31 or higher with SAP NOTE 2937175 being implemented
- To integrate with visibility provider, log your incident in SAP BCP system with component “SCM-LBN-GTT-COR”

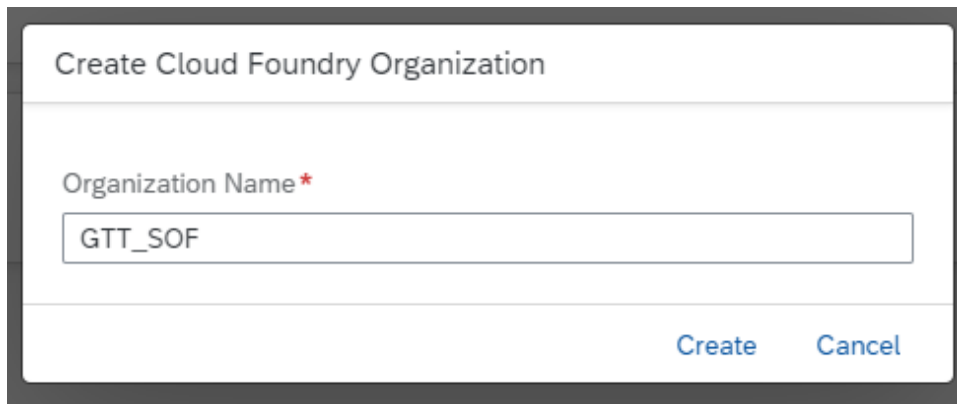
1.Environment Configuration in Logistics Business Network, Global Track and Trace Option

1.1 Enable Cloud Foundry and Create Space

1. Click “Enable Cloud Foundry” to enable the development landscape

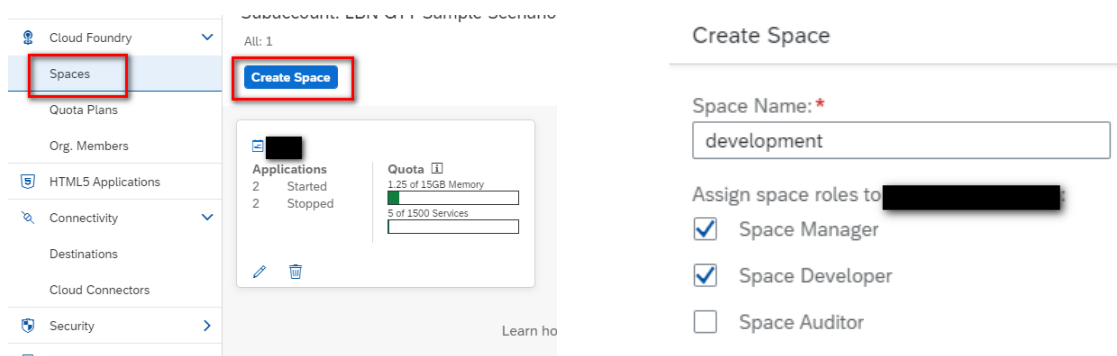


2. Fill in the Organization Name and click "Create"



A dialog box titled "Create Cloud Foundry Organization". It contains a text input field for "Organization Name" with a red asterisk indicating it is required. The field contains the text "GTT_SOF". At the bottom right, there are two buttons: "Create" and "Cancel".

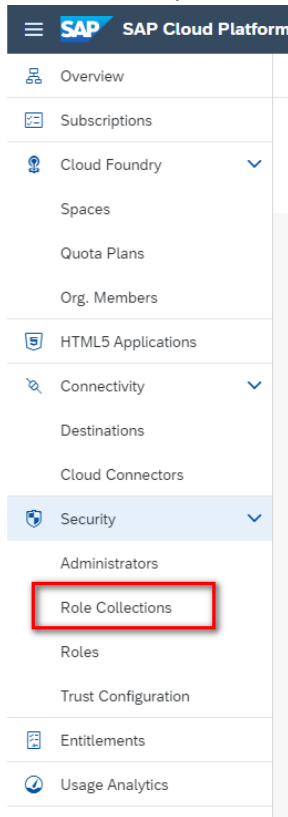
3. Go to "Cloud Foundry" -> "Spaces"
Click "Create Space"
Fill in the Space Name and click "Create"



The image shows the Cloud Foundry Spaces interface and the "Create Space" dialog. On the left, the "Spaces" menu item is highlighted in the sidebar. In the main area, a "Create Space" button is highlighted. Below it, a summary card shows "Applications" (2 Started, 2 Stopped) and "Quota" (1.25 of 15GB Memory, 5 of 1500 Services). On the right, the "Create Space" dialog is open, showing the "Space Name" field with the value "development". Below it, the "Assign space roles to" section shows three roles: "Space Manager" (checked), "Space Developer" (checked), and "Space Auditor" (unchecked).

1.2 Maintain Roles and Users

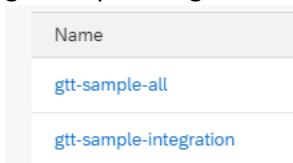
1. Go to “Security” -> “Role Collections”



2. Create 2 role collections (You can have user-defined names):

gtt-sample-all

gtt-sample-integration



- Role template “gtt-sample-all” is granted to users who have access to global track and trace option and *Sales Order Fulfillment* template app.
- Role template “gtt-sample-integration” is granted to technical users only who have access to the global track and trace APIs for development purpose.

3. Go to the role of “gtt-sample-all”, and add the following roles:

Role	Application
MessageLogAdminTemplate	lbn_gtt_core
DataModelAdminTemplate	lbn_gtt_core
LocationProcessTemplate	lbnplatform-loc-broker
BPEditTemplate	lbnplatform-bprepo
BusinessPartnerDiscoveryTemplate	lbnplatform-bprepo

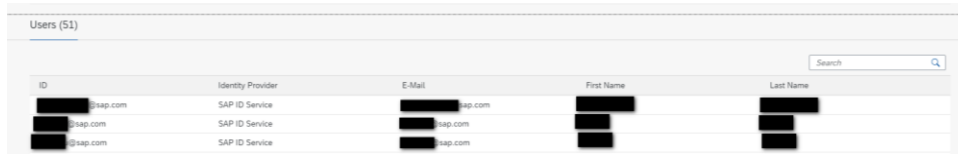
InviteBPPProcessTemplate	lbnplatform-bprepo
ManageInvitationsProcessTemplate	lbnplatform-bprepo
BlockStoreProcessTemplate	lbn-retentionmanagement-sb
ReminderTemplateProcessTemplate	lbn-retentionmanagement-sb

4. Grant "gtt-sample-all" to your **application** users:

- Application users are those who have access to global track and trace option and *Sales Order Fulfillment* template app.

IT Administrator needs to maintain application users in identity provider.

For example, user1@sap.com



The screenshot shows a table titled 'Users (51)' with columns: ID, Identity Provider, E-Mail, First Name, and Last Name. There is a search bar on the right. The table contains three rows of data, all with 'SAP ID Service' as the identity provider and email addresses ending in '@sap.com'. The first names and last names are partially visible as '...', '...', and '...'.

ID	Identity Provider	E-Mail	First Name	Last Name
...	SAP ID Service	...@sap.com
...	SAP ID Service	...@sap.com
...	SAP ID Service	...@sap.com

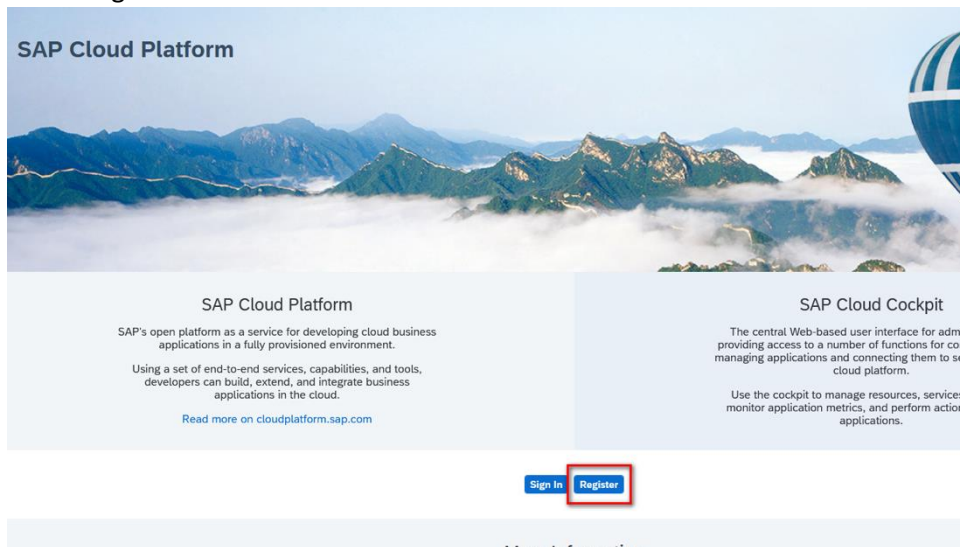
Note: The application user in this step should be different from the technical user created in Step 1.3.

1.3 Maintain Technical User

1. Go to the SAP Cloud Platform to create the technical user account:

<https://account.hana.ondemand.com/#/home/welcome>

Click "Register"



2. Fill in the mandatory fields:

Tell us about yourself

E-mail address *

First name *

Last name *

Company *

Country/Region
United States

Phone
+1 XXX-XXX-XXXX

Department *

Relationship to SAP *

Create password *

Confirm password *

Must have at least 8 characters.


☐ I have read and understood the Terms and Conditions of [SAP.com](#)

SAP will use any of the data provided hereunder in accordance with the [Privacy Statement](#).

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

Submit

3. Go to the mail inbox to activate the account

 CLOUD IDENTITY

Hello [redacted]

An account has been created for you. To activate your account for [Administration Console](#) of SAP Cloud Identity, click the link below. You will be taken to a page where you will also set a password for your account.

Click here to activate

If the link above is not displayed or does not work, copy and paste the link below to the address bar of your browser.

[redacted link]

For more information, see [SAP Cloud Identity service documentation](#)

In case of any issues, create a ticket with BC-IAM-IDS component in [SAP Support Portal](#).

4. As shown in Step 1.2, go to the role of “gtt-sample-integration”, and add the following roles:

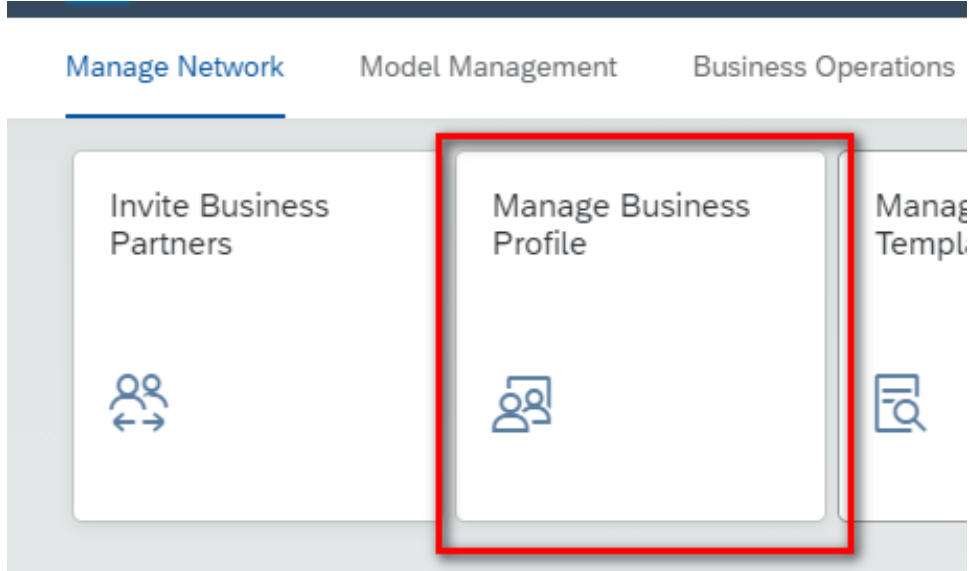
Role	Application
EMIDocIntegrationTemplate	lbn_gtt_core
MetadataServiceIntegrationTemplate	lbn_gtt_core
ReadServiceIntegrationTemplate	lbn_gtt_core
WriteServiceIntegrationTemplate	lbn_gtt_core

5. Grant "gtt-sample-integration" to your **technical** user:
For example, techuser1@techcompany.com

Users (1)

ID	Identity Provider	E-Mail	First Name	Last Name
[REDACTED]	[REDACTED].com	[REDACTED].com	[REDACTED]	[REDACTED]

6. Go to SAP Fiori Launchpad for global track and trace option
Open *Manage Business Profile* app



7. Add one Alternate Identifier:
Identifier Type: Technical User
ID: the email address of technical user

Email	Phone Number
[REDACTED]	[REDACTED]

Alternate Identifiers (1)	
Identifier Type	ID
Technical User	[REDACTED].com

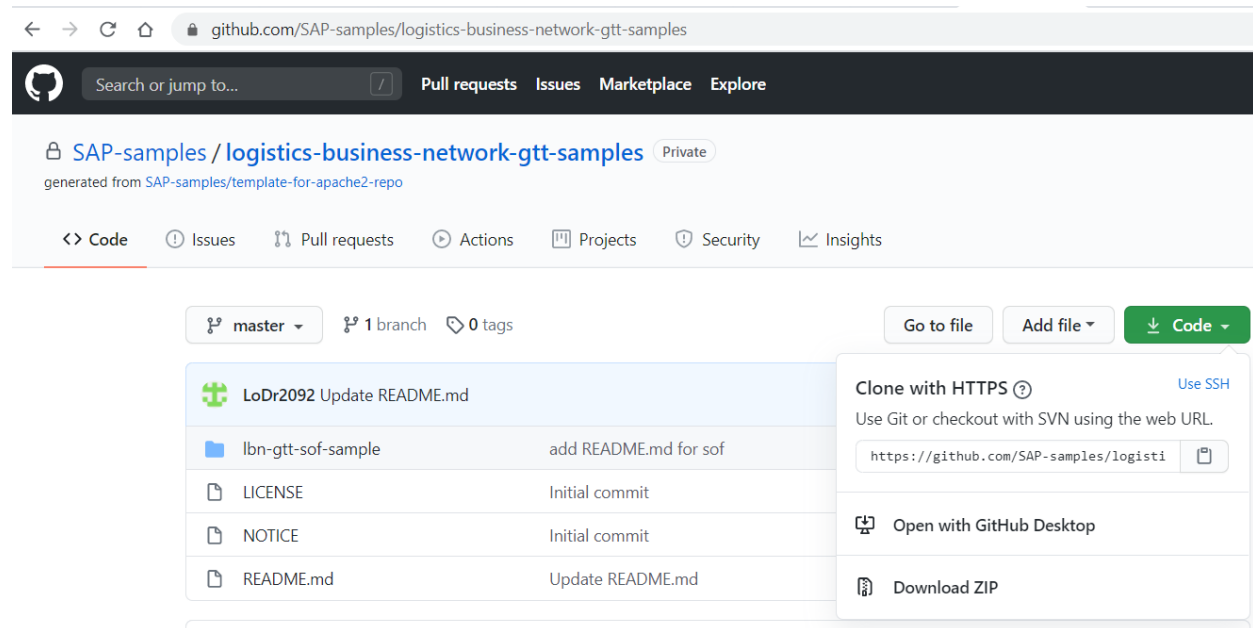
Note: The technical user in this step should be different from the application user in Step 1.2.4.

2. Implement *Sales Order Fulfillment* Template App

2.1 Download Template Code

Go to <https://github.com/SAP-samples/logistics-business-network-gtt-samples>.

Download and extract, or git clone the template code from SAP-samples on GitHub.

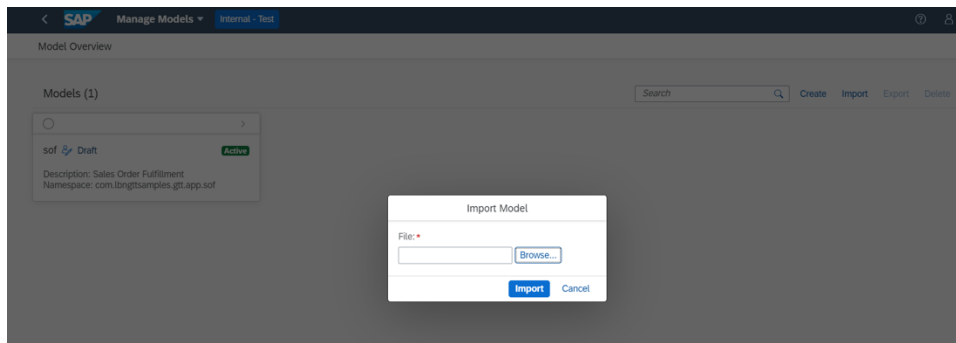


2.2 Download and Setup Tools

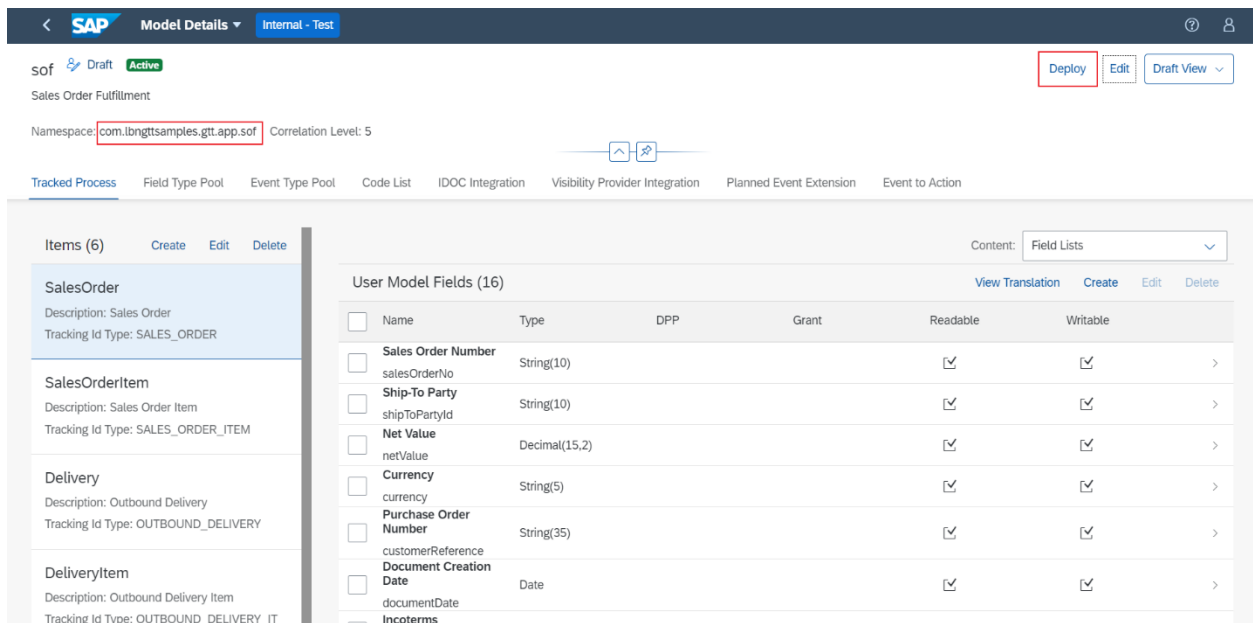
1. Install JDK 8 and Maven
2. Install NPM, Node JS, and grunt-cli
<https://www.saplearners.com/how-to-install-node-js-and-npm/>
Run the following command after NPM is installed:
`npm config set @sap:registry "https://npm.sap.com" -g`
3. Install Cloud Foundry Command Line Interface
<https://developers.sap.com/tutorials/cp-cf-download-cli.html>
4. Install MultiApps CF CLI Plugin
<https://github.com/cloudfoundry-incubator/multiapps-cli-plugin>
5. Install Cloud MTA Build Tool (MBT)
<https://sap.github.io/cloud-mta-build-tool/download/>
6. Install MAKE
<https://sap.github.io/cloud-mta-build-tool/makefile/>

2.3 Deploy Model

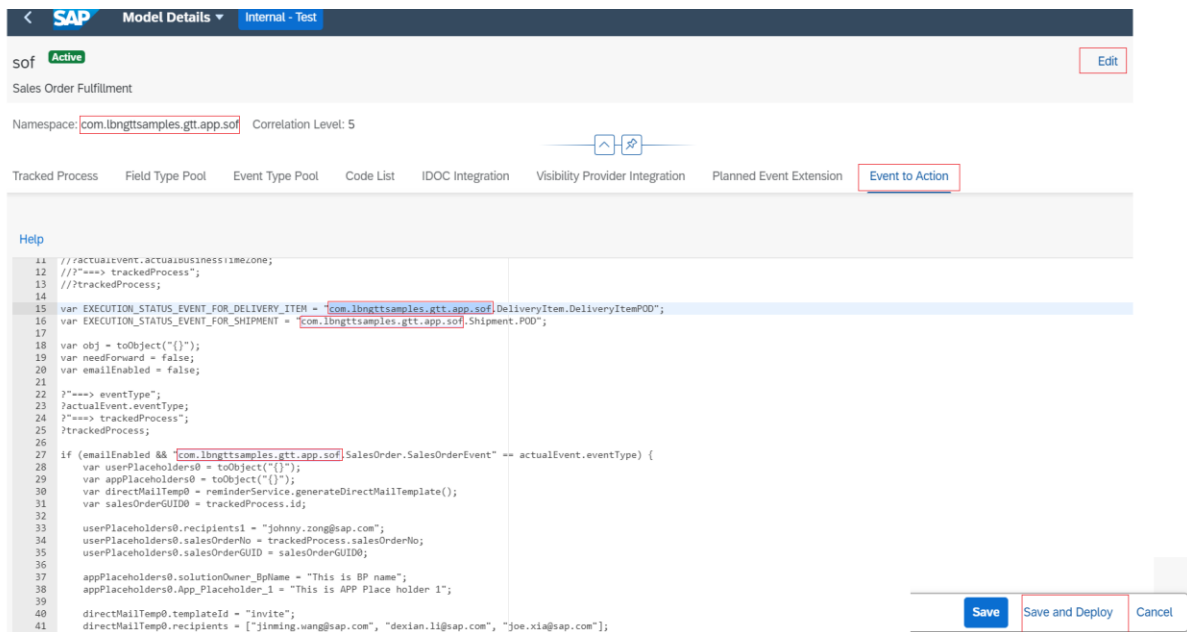
- Go to the folder *model* with downloaded source code. Make sure you have a JSON file *sof-model-YYYYMMDDxx.json* (e.g. *sof-model-2020091001.json*).
- Open *Manage Models* app on SAP Fiori Launchpad for global track and trace option
- Click “Import”
- Click “Browse” to select the *sof-model-YYYYMMDDxx.json* file
- Click “Import” to import the model



- Click “Deploy”
- Write down the model namespace for later use, in this case, “com.lbngttsamples.gtt.app.sof” in the picture below



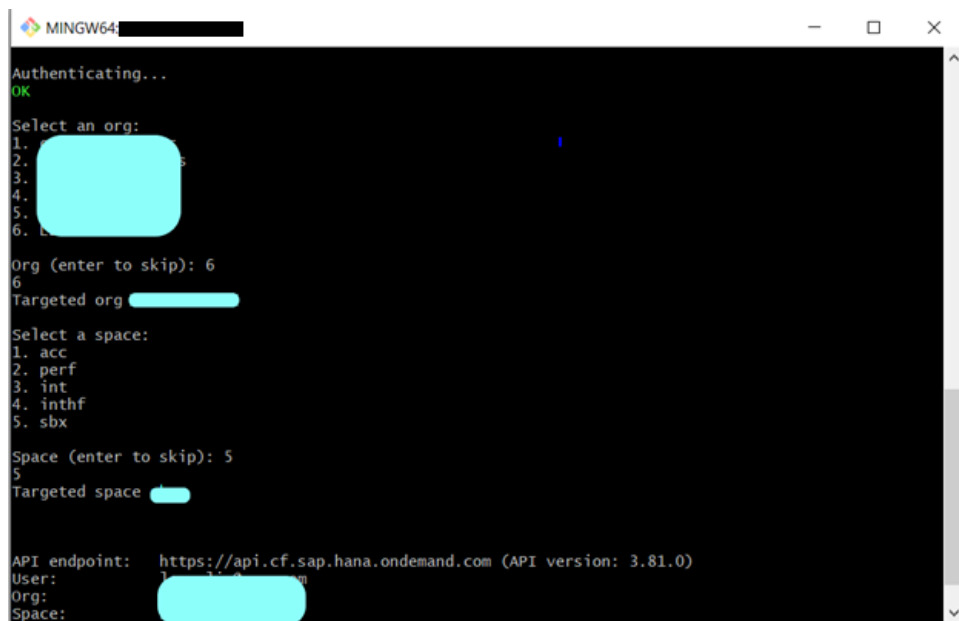
- Click “Edit”
- Go to *Event to Action* tab
- Replace all the namespace strings, in this case, “com.lbngttsamples.gtt.app.sof” in *Event to Action* script with the model namespace you wrote down in the previous step
- Click “Save and Deploy” to deploy the model



2.4 Package and Deploy

2.4.1 Login to SAP Cloud Foundry

- Open a command window, *cf login* to SAP Cloud Foundry with your username and password
- Select the organization and space in which you want to deploy the template apps

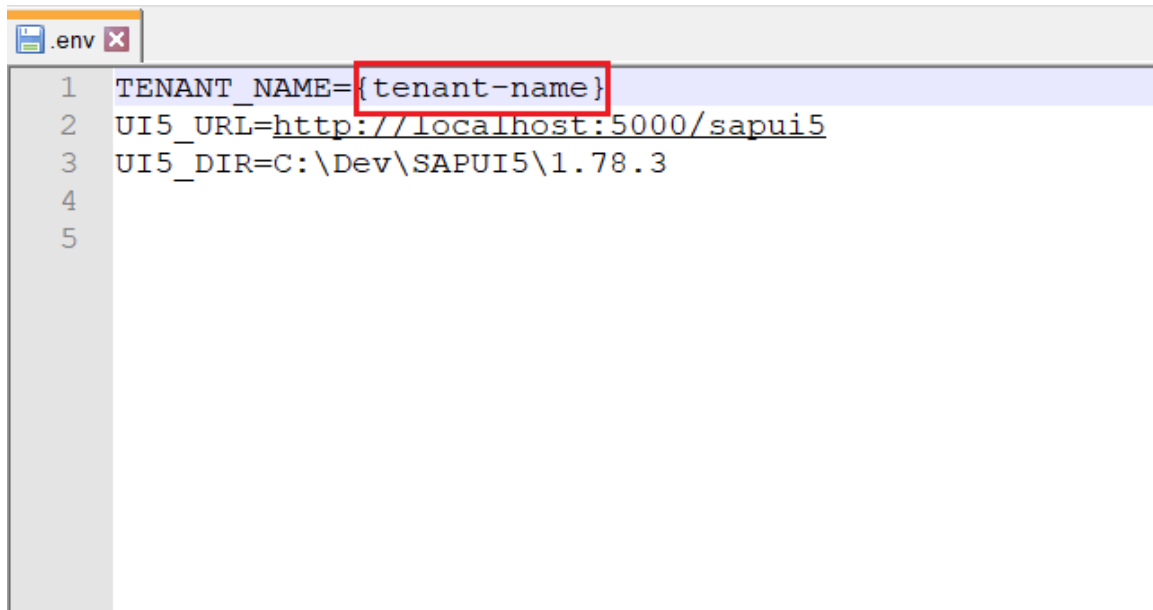


2.4.2 Package and Deploy *Sales Order Fulfillment* Frontend App

Go to GTT-V2-Sample-TrackSalesOrders-UI directory.

- Open the `./UI/.env` file

- Replace {tenant-name} with part of the namespace you got from Step 2.3.
For example, replace it with “lbngttsamples” in “com.lbngttsamples.gtt.app.sof” shown in Step 2.3.



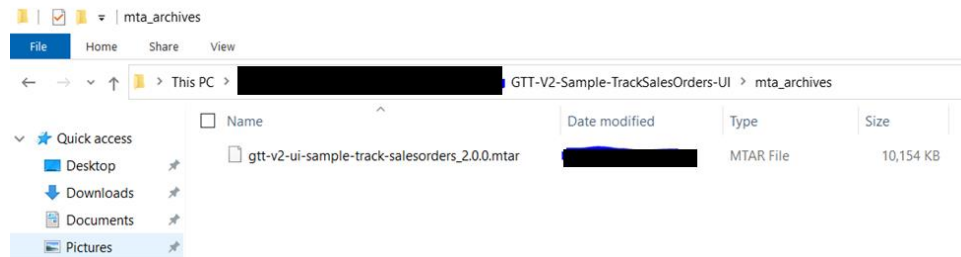
```
.env
1 TENANT_NAME={tenant-name}
2 UI5_URL=http://localhost:5000/sapui5
3 UI5_DIR=C:\Dev\SAPUI5\1.78.3
4
5
```

In the command line tool,

- Run the following command to build the MTAR package:

mbt build

A MTAR file `./mta_archives/gtt-v2-ui-sample-track-salesorders_xxx.mtar` is then generated.



- To deploy the MTAR package in Cloud Foundry, run the following command:

cf deploy ./mta_archives/gtt-v2-ui-sample-track-salesorders-service_xxx.mta

The deployment time depends on your network conditions.

2.4.3 Package and Deploy SAP Fiori Launchpad for *Sales Order Fulfillment*

Go to `GTT-V2-Sample-App-FLP` directory.

- Open the *mtad.yaml* file
- Replace {unique-identifier} with unique information such as your company name or cloud foundry global account name

```

10 - name: lbn-gtt-sample-app-appRouter
11   type:approuter.nodejs
12   path: flp_appRouter
13   parameters:
14     disk-quota: 256M
15     memory: 256M
16     routes:
17       - route: lbn-gtt-samples-app-flp-{unique-identifier}.${domain}
18   properties:
19     SAP_JWT_TRUST_ACL:
20       - clientid: "*"
21       identityzone: "*"
22     destinations:
23       - name: lbn-gtt-sof-api
24         url: "{lbn-gtt-sof-api/url}"
25         forwardAuthToken: true
26   requires:
27     - name: lbn-gtt-sample-app-uaa
28     - name: lbn-gtt-sample-app-runtime
29     - name: lbn-gtt-sample-portal-resources
30     - name: lbn-gtt-sof-api
31
32 - name: flp
33   type: com.sap.portal.content
34   path: flp
35   parameters:
36     stack: cflinuxfs3
37     memory: 128M
38     buildpack: 'https://github.com/cloudfoundry/nodejs-buildpack/releases/download/v1.6.39/nodejs-buildpack-cflinuxfs3-v1.6.39.zip'
39   requires:
40     - name: lbn-gtt-sample-app-uaa
41     - name: lbn-gtt-sample-portal-resources
42     - name: gtt-ui-sample-track-salesorders-app-host
43 resources:
44   - name: lbn-gtt-sample-app-runtime
45     parameters:
46       service-plan: app-runtime
47       service: html5-apps-repo
48     type: org.cloudfoundry.managed-service
49   - name: lbn-gtt-sample-portal-resources
50     parameters:
51       service-plan: standard
52       service: portal
53     type: org.cloudfoundry.managed-service
54   - name: lbn-gtt-sof-api
55     properties:
56       url: https://gtt-sample-track-salesorders-service-{unique-identifier}.${default-domain}
57   - name: lbn-gtt-sample-app-uaa

```

In the command line tool,

- Run the following command to build the MTAR package:

```
mbt build
```

A MTAR file *./mta_archives/lbn-gtt-sample-app-flp_xxx.mtar* is then generated.

- Deploy the MTAR package in Cloud Foundry by running the following command:

```
cf deploy ./mta_archives/lbn-gtt-sample-app-flp_xxx.mtar
```

2.4.4 Package and Deploy Sales Order Fulfillment Backend Service

Replace User/Password, Here Map License Key and Host Configuration

- Obtain Here Map license from Step 4.1
- Open the file *mtad.yaml* under the folder “GTT-V2-Sample-TrackSalesOrders-Service”:

```
mtad.yaml
1  ---
2  _schema-version: '3.1'
3  ID: gtt-v2-sample-track-salesorders-service
4  version: 2.3.0
5  modules:
6  - name: gtt-sample-track-salesorders-service
7    type: java
8    path: srv/
9    properties:
10   SPRING_PROFILES_ACTIVE: cloud
11   DESTINATION_GTT: gtt_core_engine_flp_base_url
12   HERE_MAP_KEY: {here-map-license-key}
13   parameters:
14     buildpack: java_buildpack
15     health-check-timeout: 180
16     host: gtt-sample-track-salesorders-service-{unique-identifier}
17   build-parameters:
18     ignore:
19       - "*.md"
20     build-result: target/*.jar
21     timeout: 10m
22   requires:
23     - name: lbn-gtt-sample-app-uaa
24   resources:
25     - name: lbn-gtt-sample-app-uaa
26       type: org.cloudfoundry.existing-service
27
28
29
```

- Replace {here-map-license-key} with the license key you got
- Replace {unique-identifier} with unique information such as your company name or cloud foundry global account name
- Open the file *./srv/src/main/java/com/sap/gtt/v2/sample/sof/constant/Constants.java*
 - Replace “com.lbngttsamples.gtt.app.sof” with the model namespace you got from Step 2.3

```
Constants.java
1  package com.sap.gtt.v2.sample.sof.constant;
2
3  public class Constants {
4    public static final String GTT_MODEL_NAMESPACE = "com.lbngttsamples.gtt.app.sof";
5    public static final String GTT_MODEL_NAMESPACE_WRITE_SERVICE = GTT_MODEL_NAMESPACE + ".sofWriteService";
6    public static final String MODEL_NAMESPACE = GTT_MODEL_NAMESPACE + ".sofService";
7
8    public static final String URL_SPLITTER = "/";
9    public static final String ENTITY_CONTAINER_NAME = "EntityContainer";
10   public static final String SAP_ICON_LEAD = "sap-icon://lead";
11   public static final String ID = "id";
12
13   public static final String EVENT_STATUS_PLANNED = "PLANNED";
14
15   public static final String PROCESS_STATUS_OVERRIDE = "OVERRIDE";
16
```

- Run the following command to build the MTAR package:

mbt build

A MTAR file *./mta_archives/gtt-v2-sample-track-salesorders-service_xxx.mtar* is then generated.

- Deploy the MTAR package in Cloud Foundry by running the following command:

```
cf deploy ./mta_archives/gtt-v2-sample-track-salesorders-service_xxx.mtar
```

2.4.5 Check Deployed Apps on SAP Cloud Foundry Cockpit

After operations from Step 2.4.1 to Step 2.4.3, check the deployed apps on SAP Cloud Platform Cockpit.

The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar contains navigation links: Applications, Services, SAP HANA Cloud, SAP HANA, Database Systems, Tenant Databases, Service Requests, Portal, Routes, Security Groups, Events, and Members. The main area displays a list of applications in the 'Space: [redacted] Applications' view, filtered to 3 of 40. A 'Deploy Application' button is visible. The table lists the following applications:

Requested State	Name	Instances	Disk Quota	Memory
Started	gtt-sample-track-salesorders-service	1/1	1024 MB	1024 MB
Stopped	gtt-ui-sample-track-salesorders-ui_deployer	0/1	1024 MB	256 MB
Started	lbn-gtt-sample-app-appRouter	1/1	256 MB	256 MB

2.4.6 Maintain Destination

1. Obtain the UAA Token

- Go to the *Sales Order Fulfillment* backend application that you have deployed

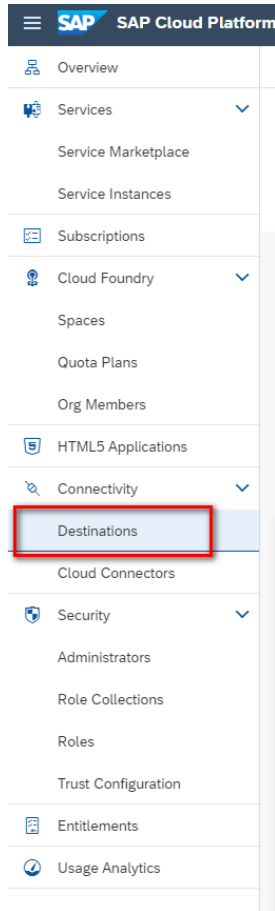
The screenshot shows the application status for 'gtt-sample-track-salesorders-service-green' with 1/1 instances. The status is 'Started'.

- Find the environment of UAA
- Copy the value of **clientid** and **clientsecret**

The screenshot shows the 'Service Bindings' page for the application 'gtt-sample-track-salesorders-service-green'. It displays a binding for 'lbn-gtt-sample-app-uaa' to the 'xsuaa' service. Below the binding, the sensitive data is expanded, showing the following configuration:

```
{
  "tenantmode": "dedicated",
  "sburl": "https://api.authentication.eu10.hana.ondemand.com",
  "subaccountid": "[redacted]",
  "clientid": "[redacted]",
  "xsappname": "[redacted]",
  "clientsecret": "[redacted]",
  "url": "https://api.authentication.eu10.hana.ondemand.com",
  "uaadomain": "authentication.eu10.hana.ondemand.com",
  "verificationkey": "-----BEGIN PUBLIC KEY-----",
  "apiurl": "https://api.authentication.eu10.hana.ondemand.com",
  "identityzone": "lbn-gtt-samples",
  "identityzoneid": "[redacted]",
  "tenantid": "[redacted]",
  "zoneid": "[redacted]"
}
```

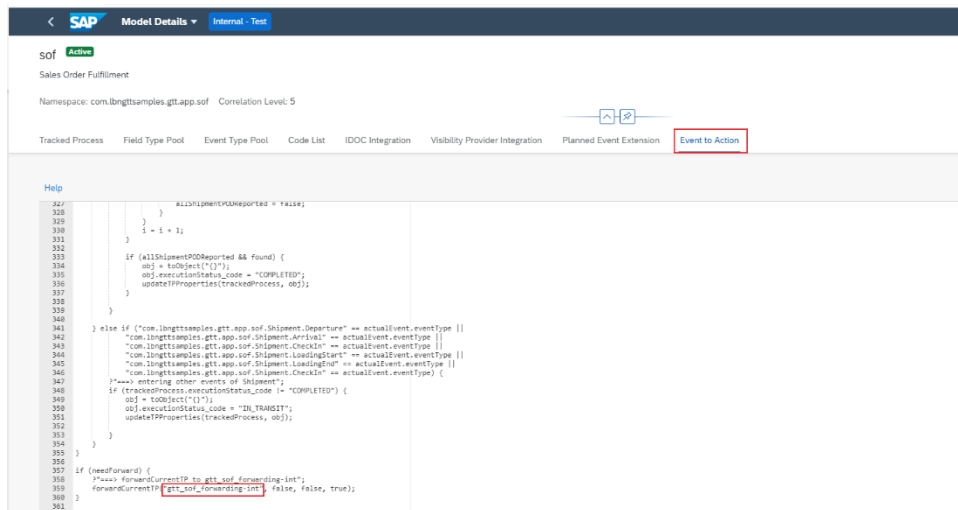
2. Go to “Destinations”



3. Configure Destination of Event to Action Forwarding

Name	e.g. gtt_sof_forwarding_int
Type	HTTP
Description	<optional>
URL	e.g. https://gtt-sample-track-salesorders-service.cfapps.eu10.hana.ondemand.com/sap/logistics/gtt/sample/sof/rest/v1/forward
Proxy Type	Internet
Authentication	OAuth2ClientCredentials
Client ID	Copied in the previous step
Client Secret	Copied in the previous step
Token Service URL	e.g. https://lbn-gtt-samples.authentication.eu10.hana.ondemand.com/oauth/token

Note: You need to update the destination name in *Event to Action* script of *Sales Order Fulfillment* model if you change the name from “gtt_sof_forwarding_int” to others.



4. Configure Destination of GTT Core Engine Base URL

Name	gtt_core_engine_flp_base_url
Type	HTTP
Description	<optional>
URL	e.g. https://lbn-gtt-samples.gtt-flp-lbnplatform-int.cfapps.sap.hana.ondemand.com/api
Proxy Type	Internet
Authentication	BasicAuthentication
User	User from Step 1.3
Password	Password from Step 1.3

Quota Plans
Org Members
HTML5 Applications
Connectivity
Destinations
Cloud Connectors
Entitlements
Usage Analytics
Favorites
Useful Links
Legal Information

Destination Configuration

Name: * gtt_core_engine_flp_base_url

Type: HTTP

Description:

URL: * https://lbn-gtt-samples.gtt-flp-lbnplatform-int.cfapps.sap.hana.ondemand.com/api

Proxy Type: Internet

Authentication: BasicAuthentication

User: *

Password: *****

Additional Properties

☒ Use default JDK truststore

Save Cancel

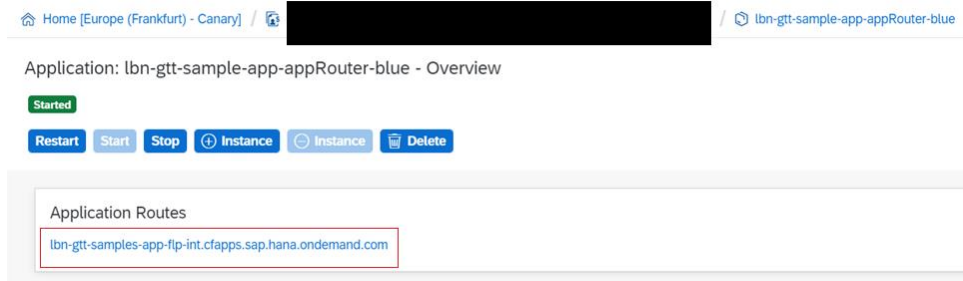
2.4.7 Maintain Roles and Users

- As shown in Step 1.2, go to the role of "gtt-sample-all", and add the following role:

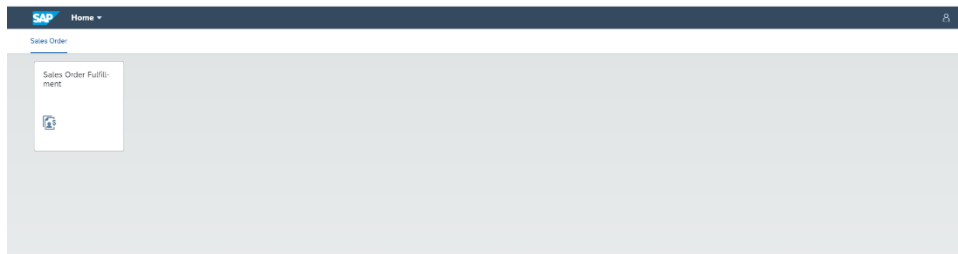
Role	Application
SalesOrderFulfillmentAppUser	tt_sof_{space}

2.4.8 Open *Sales Order Fulfillment* Template App from SAP Fiori Launchpad

- Obtain the Application Routes from SAP Cloud Foundry Cockpit



- Click the link of Application Routes
The system browser then navigates to the SAP Fiori Launchpad of *Sales Order Fulfillment*



- Open the *Sales Order Fulfillment* template app

Sales Order Number	Ship-To Party	Completion Rate	Overall Delay Impact (By Value)	Net Value	Is Delayed
9900023663	0000010105	265.00 EUR		265.00 EUR	No
9900063395	0000010105			265.00 EUR	No
9900015038					No
9900095285	0000010105			265.00 EUR	No

3. Implement ERP Extractor

For ERP extractor implementation, please refer to the following document:

02_Extractor_Creation_Guide-SOF.pdf

4. Other Configuration

4.1 Obtain Here Map License

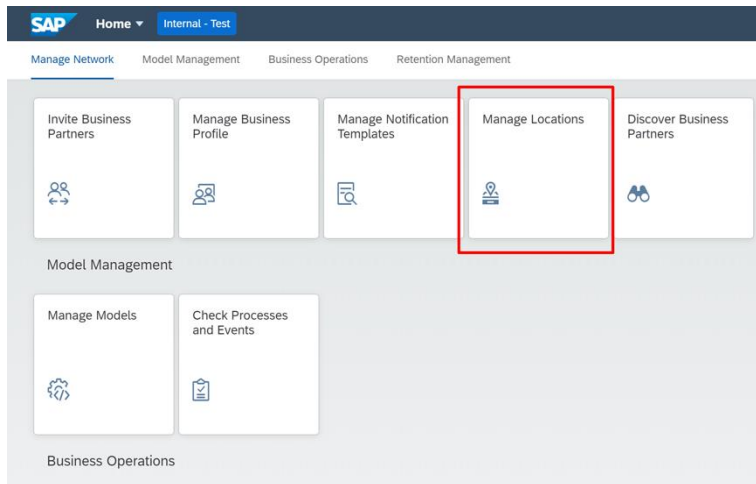
To have the map view, log the incident with the URL of Fiori Launchpad of *Sales Order Fulfillment* template app in SAP BCP system under component “SCM-LBN-GTT-COR”. The Here Map tile service is enabled manually.

For quick validation, you can also use the tile service of other map services.

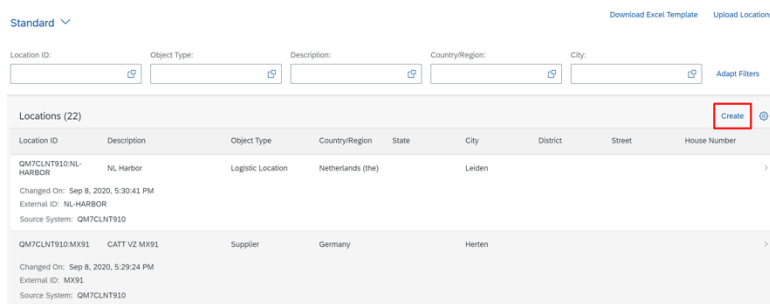
4.2 Maintain Location Master

To show stops and planned events in the map, you need to maintain location master data with valid geocoordinates in *Manage Locations* app.

1. Open *Manage Locations* app in your SAP Fiori Launchpad for global track and trace option



2. Click “Create” to create a new location

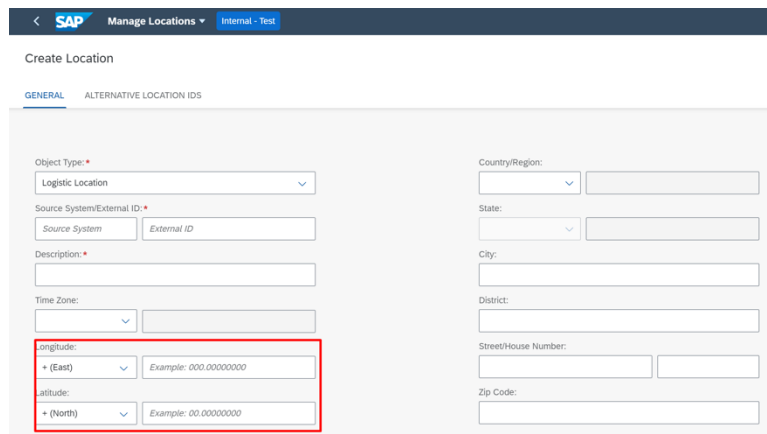


3. Maintain location information with valid geocoordinates

To integrate with ERP, you need to keep the Object Type, Source System, and External ID location consistent with ERP location information.

To know the details, please refer to the following document:

02_Extractor_Creation_Guide-SOF.pdf



4.3 Maintain Data Contributors

To receive actual events from other business partners, you need to invite them to join your business network. You can find more details in [Chapter 6.3 of *Administration Guide for Version 2*](#).