Implementation Guide – *Track Shipments* Template App

Contents

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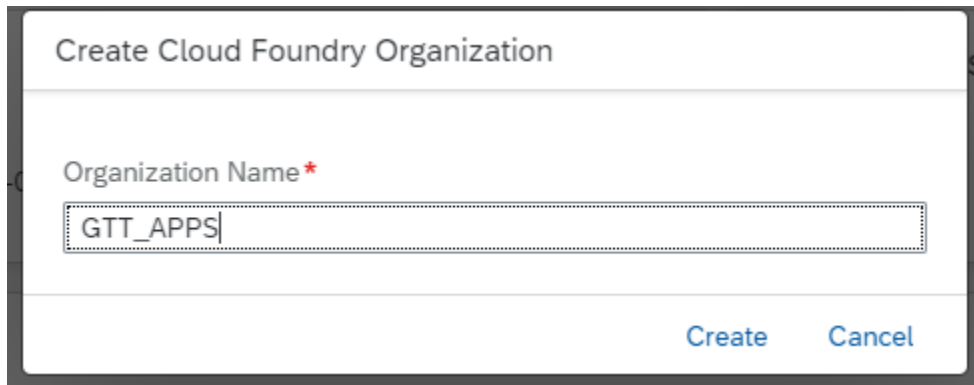
1.Environment Configuration in Logistics Business Network, Global Track and Trace Option

1.1 Enable Cloud Foundry and Create Space

1. Click "Enable Cloud Foundry" to enable the development landscape
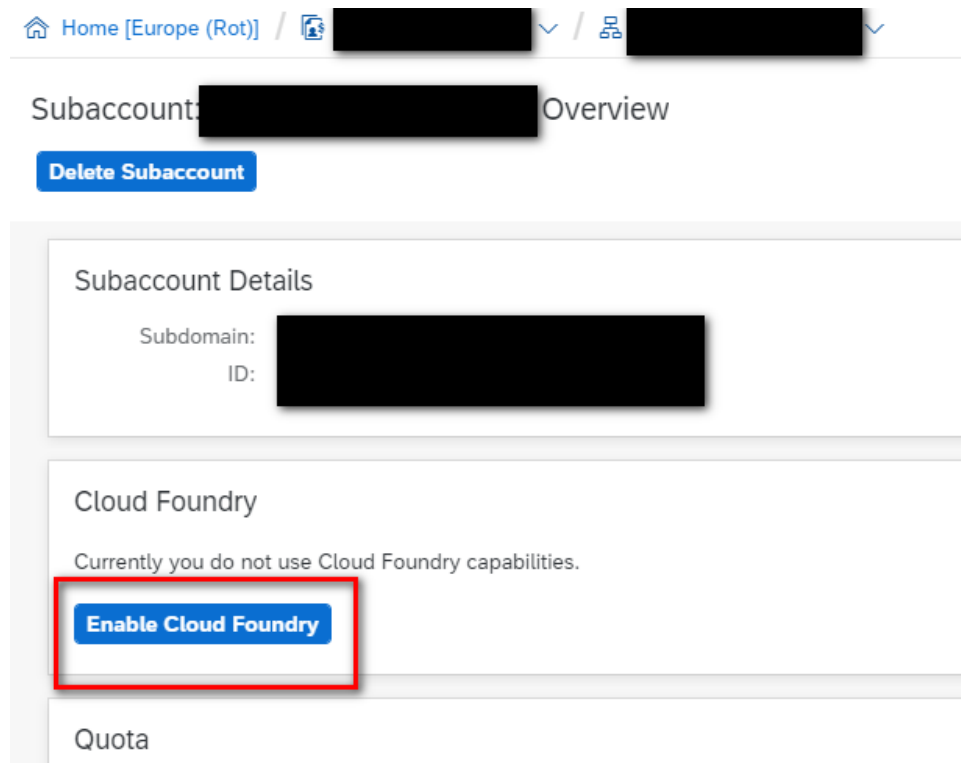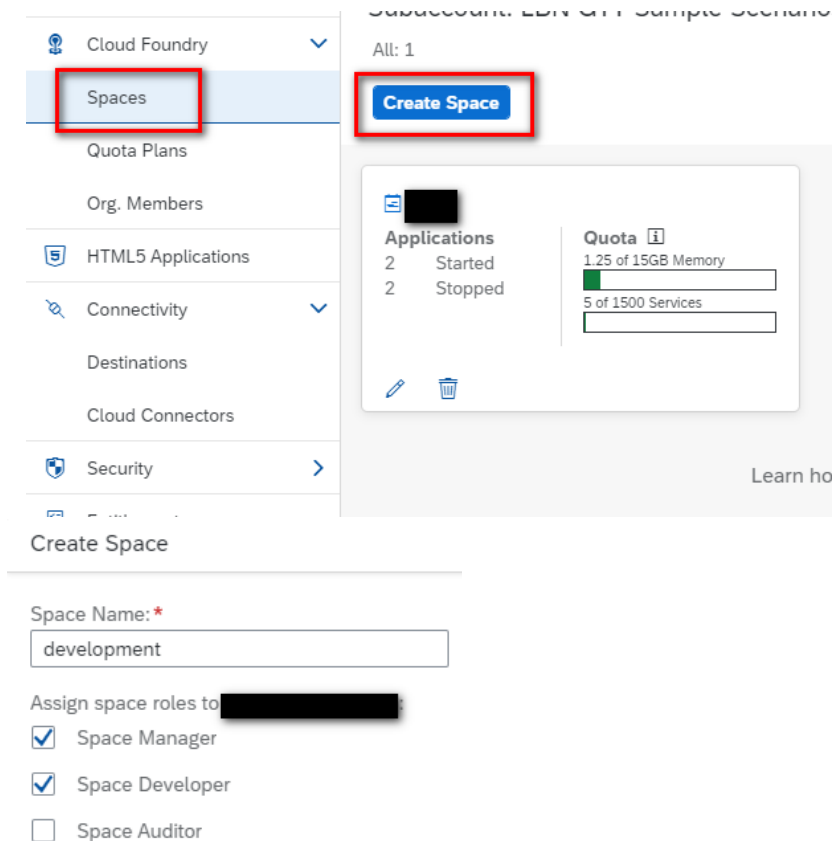
2. Fill in the Organization Name and click "Create"

Create Cloud Foundry Organization

Organization Name*

GTT_APPS

Create   Cancel

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

Create Space

Space Name:*

development

Assign space roles to

☑ Space Manager

☑ Space Developer

☐ Space Auditor

1.2 Maintain Roles and Users

1. Go to "Security" -> "Role Collections"
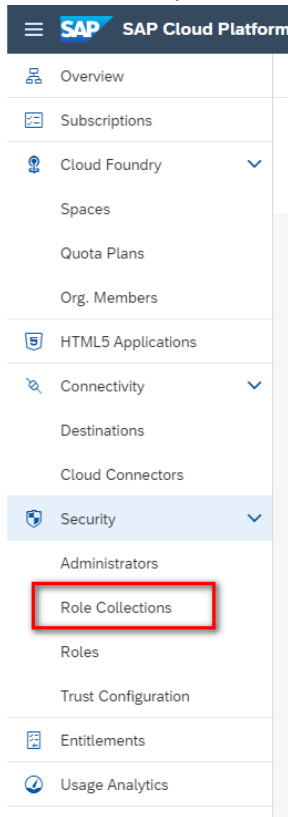
2. Create 2 role collections (You can have user-defined names):
   gtt-sample-all
   gtt-sample-integration
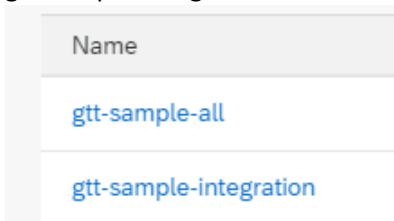
- Role template "gtt-sample-all" is granted to users who have access to global track and trace option and *Track Shipments* template app.
- Role template "gtt-sample-integration" is granted to technical users only who have access to the global track and trace APIs for development purpose.

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
InviteBPProcessTemplate	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 *Track Shipments* template app.

IT Administrator needs to maintain application users in identity provider.
For example, user1@sap.com

Users (51)

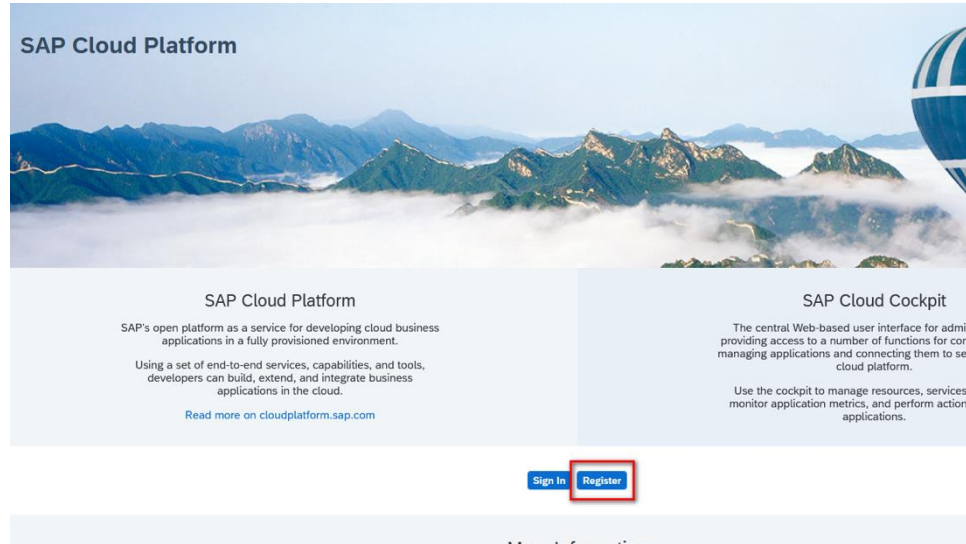| ID	Identity Provider	E-Mail	First Name	Last Name
████@sap.com	SAP ID Service	████sap.com	████	████
████@sap.com	SAP ID Service	████@sap.com	████	████
████@sap.com	SAP ID Service	████@sap.com	████	████

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:

3. Go to the mail inbox to activate the account
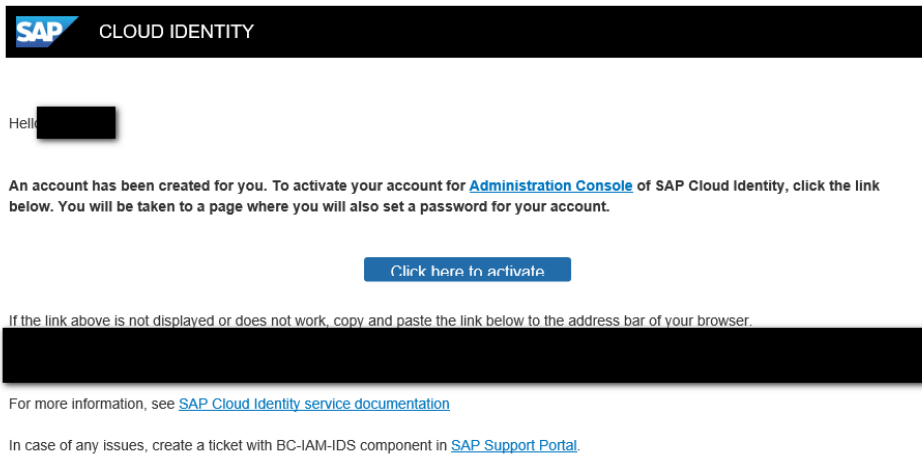
4. As shown in Step 1.2, go to the role of "gtt-sample-integration", and add the following roles:

Role	Application
EMIDocIntegrationTemplate	lbn_gtt_core
MetadataServiceIntegrationTemplate	lbn_gtt_core
ReadServiceIntegrationTemplate	lbn_gtt_core
WriteServiceIntegrationTemplate	lbn_gtt_core
LocationProcessTemplate	lbnplatform-loc-broker

5. Grant "gtt-sample-integration" to your **technical** user:
   For example, techuser1@techcompany.com
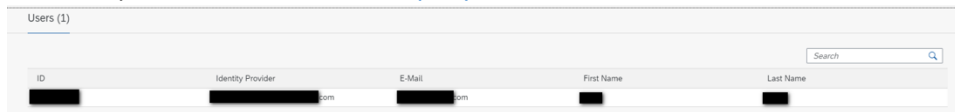
6. Go to SAP Fiori Launchpad for global track and trace option
   Open *Manage Business Profile* app
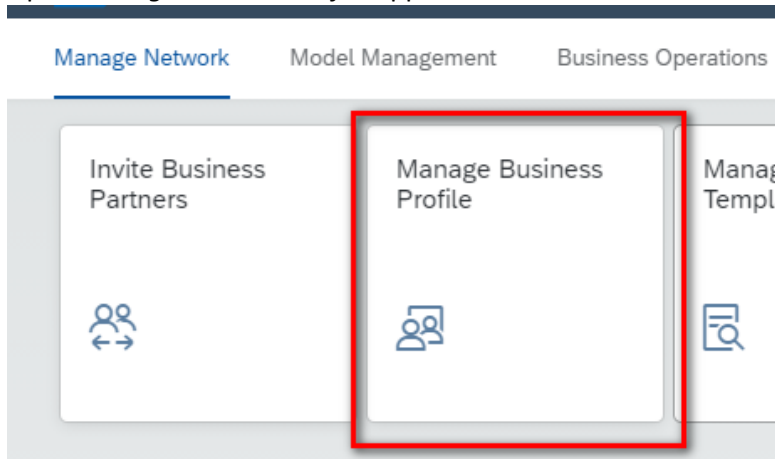
7. Add one Alternate Identifier:
   Identifier Type: Technical User
   ID: the email address of technical user

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

1.4 Maintain Destination of GTT Core Engine Base URL

- Go to "Destinations"
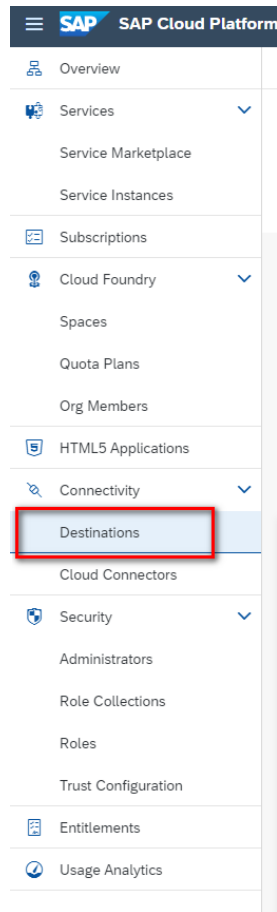
- Configure Destination of GTT Core Engine Base URL

Name	gtt_core_engine_flp_base_url
Type	HTTP
Description	<optional>
URL	e.g. https://lbn-gtt-samples.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com/api
Proxy Type	Internet
Authentication	BasicAuthentication
User	User from Step 1.3
Password	Password from Step 1.3

2.Implement *Track Shipments* 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.
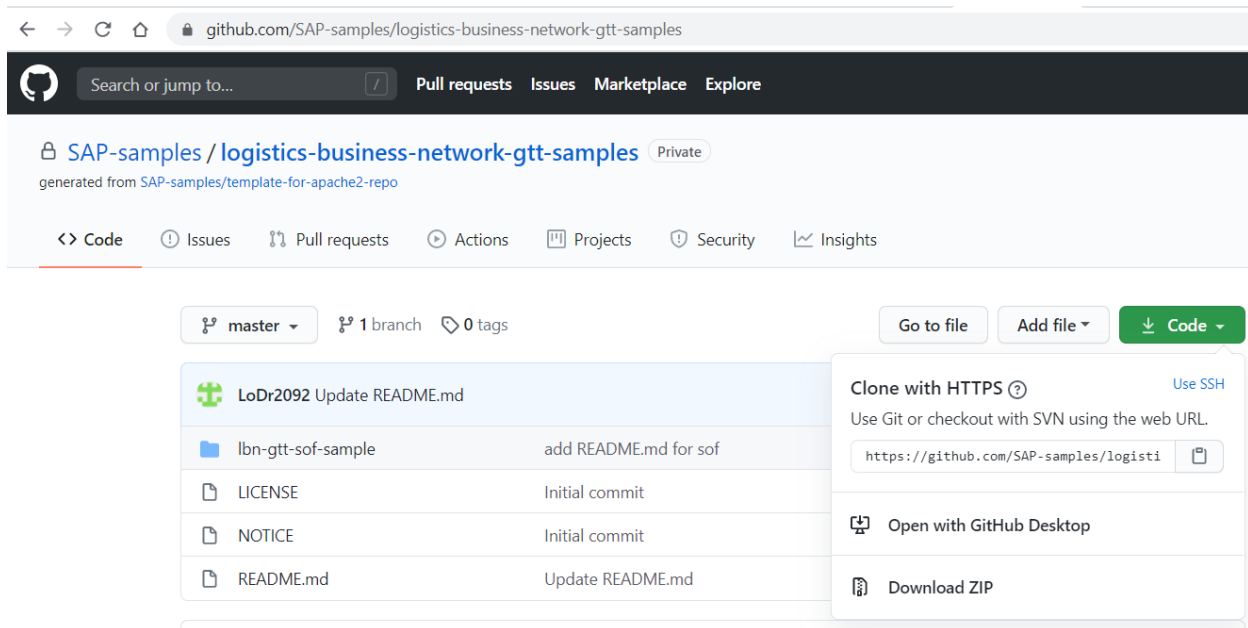
2.2 Download and Setup Tools

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
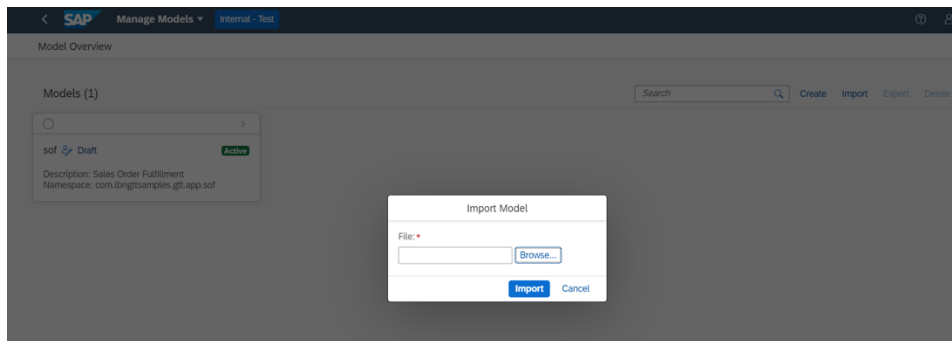
- Click "Deploy"
- Write down the model namespace for later use, in this case, "com.lbngttsamples.gtt.app.sof" in the picture below
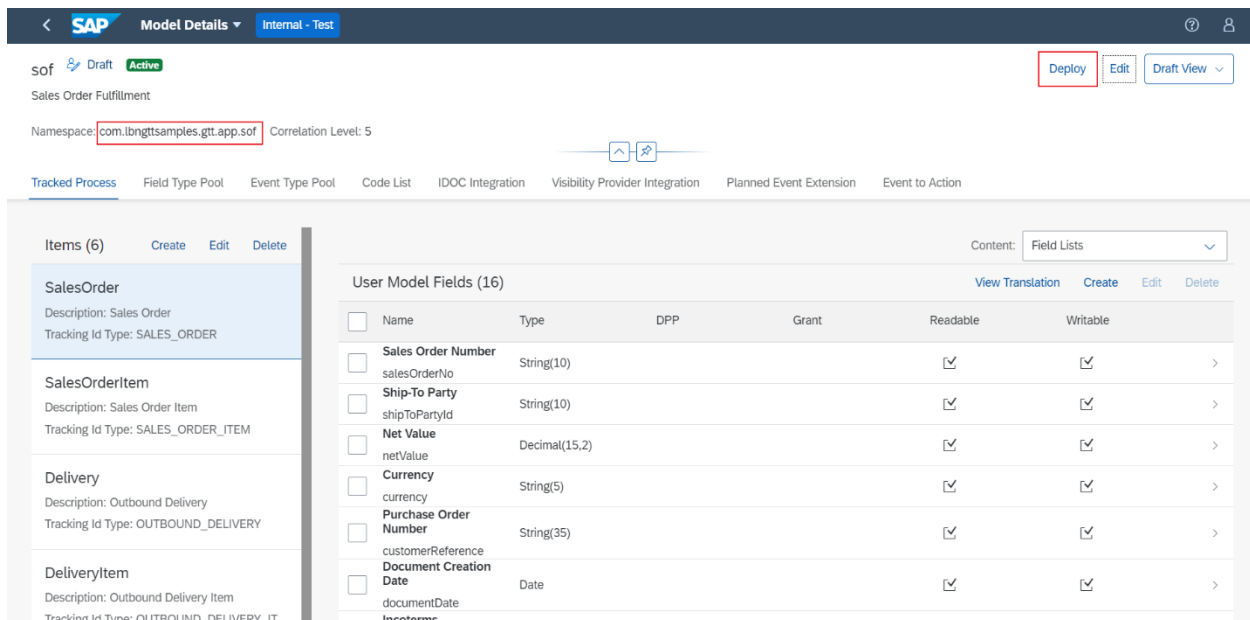
- Click "Edit"
- Go to *Event to Action* tab
- Replace all the namespace strings, in this case, "com.lbngttsamples.gtt.app.sof" in *Event to Action* script with the model namespace you wrote down in the previous step
- Click "Save and Deploy" to deploy the model
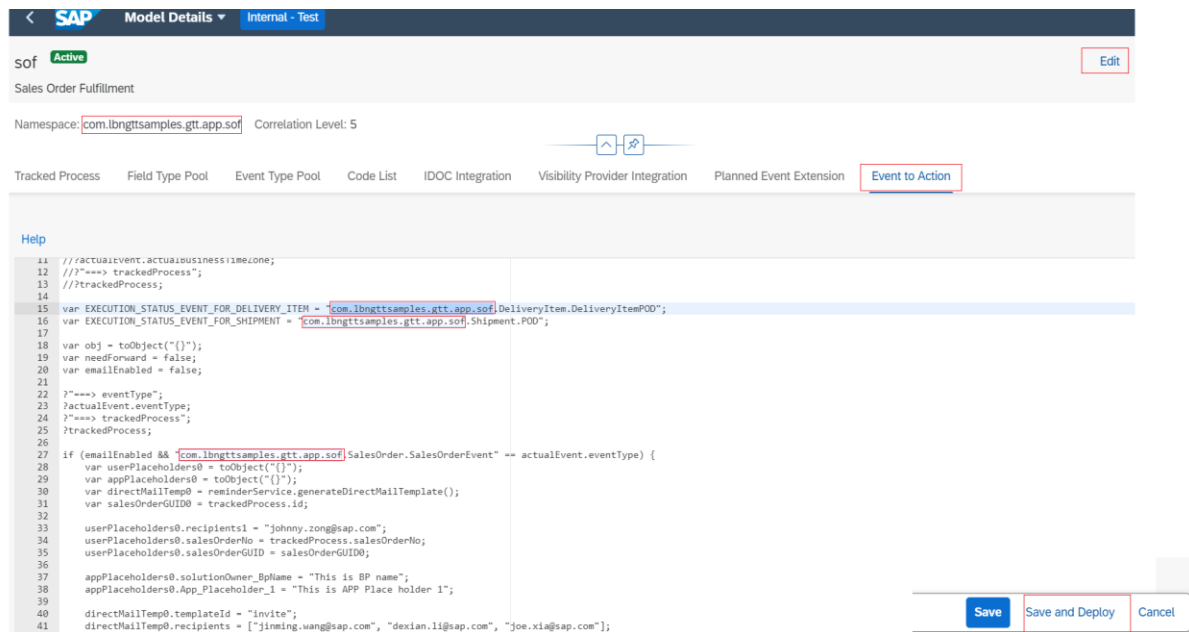
2.4 Package and Deploy

2.4.1 Login to SAP Cloud Foundry
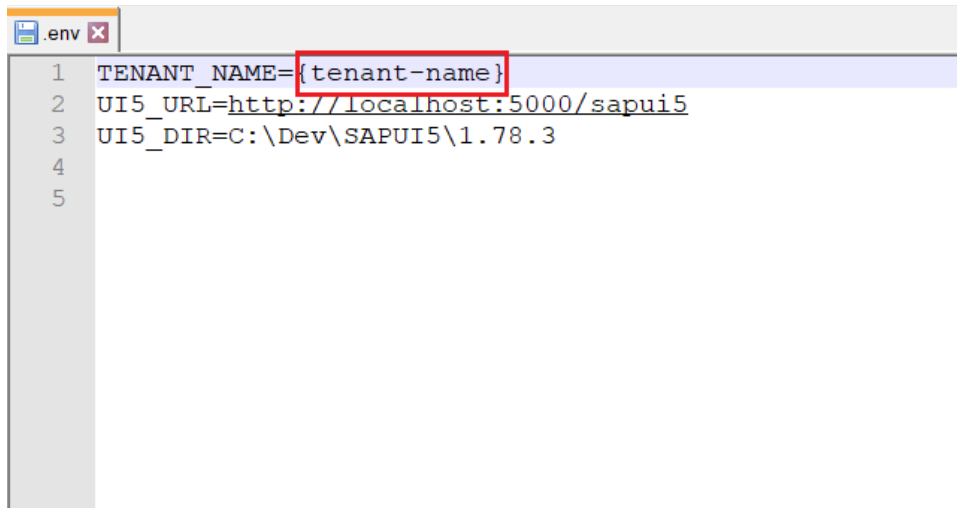
- Open a command window, *cf login* to SAP Cloud Foundry with your username and password
- Select the organization and space in which you want to deploy the template apps
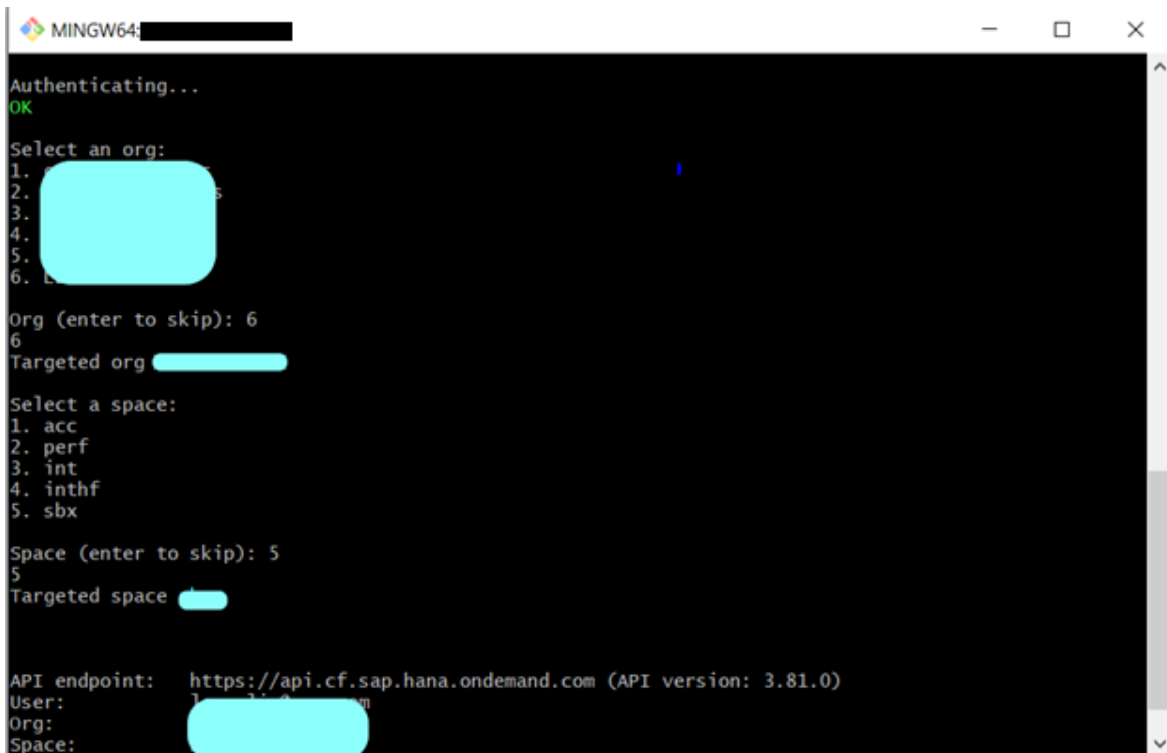
2.4.2 Package and Deploy *Track Shipments* Frontend App

Go to GTT-V2-Sample-TrackShipments-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-shipments_xxx.mtar is then generated.
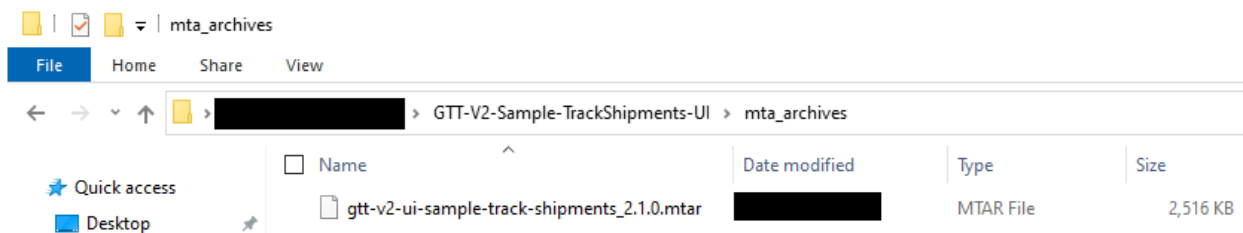
- To deploy the MTAR package in Cloud Foundry, run the following command:

```
cf deploy ./mta_archives/gtt-v2-ui-sample-track-shipments_xxx.mtar
```

The deployment time depends on your network conditions.

2.4.3 Package and Deploy SAP Fiori Launchpad for *Track Shipments*
Go to GTT-V2-Sample-App-FLP directory.

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

```yaml
1    ---
2    ID: lbn-gtt-sample-app-flp
3    _schema-version: '3.1'
4    parameters:
5        deploy_mode: html5-repo
6        enable-parallel-deployments: true
7        append: "${space}"
8        xsappname: tt_sof_${append}
9    version: 2.2.0
10   modules:
11   - name: lbn-gtt-sample-app-appRouter
12     type: approuter.nodejs
13     path: flp_appRouter
14     parameters:
15       disk-quota: 256M
16       memory: 256M
17       routes:
18       - route: lbn-gtt-samples-app-flp-{unique-identifier}.${domain}
19     properties:
20       SAP_JWT_TRUST_ACL:
21       - clientid: "*"
22         identityzone: "*"
23       destinations:
24       - name: lbn-gtt-sof-api
25         url: "~{lbn-gtt-sof-api/url}"
26         forwardAuthToken: true
27       - name: lbn-gtt-sst-api
28         url: "~{lbn-gtt-sst-api/url}"
29         forwardAuthToken: true
30       - name: lbn-gtt-pof-api
31         url: "~{lbn-gtt-pof-api/url}"
32         forwardAuthToken: true
```

```
 mta.yaml
40              - name: lbn-gtt-sample-app-uaa
41              - name: lbn-gtt-sample-portal-resources
42              - name: gtt-ui-sample-track-shipments-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-sst-api
55        properties:
56          url: https://gtt-sample-track-shipments-service-{unique-identifier}.${default-domain}
57      - name: lbn-gtt-sample-app-uaa
58        type: com.sap.xs.uaa-application
59        parameters:
60          config:
61            xsappname: "${xsappname}"
62            tenant-mode: dedicated
63            oauth2-configuration:
64              system-attributes:
65                - rolecollections
66            description: Security profile of track shipments application
```

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:

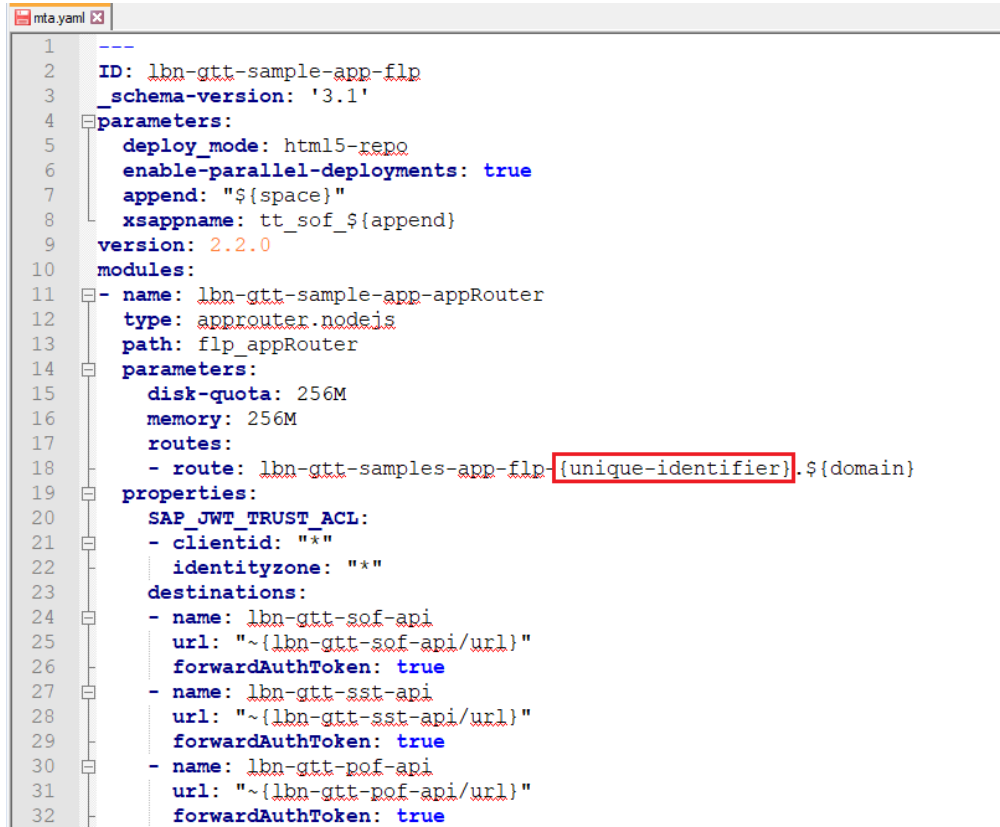 cf deploy ./mta_archives/lbn-gtt-sample-app-flp_xxx.mtar

2.4.4 Package and Deploy *Track Shipments* Backend Service

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

- Obtain Here Map license from Step 4.1
- Open the file *mta.yaml* under the folder "GTT-V2-Sample-TrackShipments-Service":

```
mta.yaml
1    ---
2    _schema-version: '3.1'
3    ID: gtt-v2-sample-track-shipments-service
4    version: 2.3.0
5    modules:
6    - name: gtt-sample-track-shipments-service
7        type: java
8        path: srv/
9        properties:
10          SPRING_PROFILES_ACTIVE: cloud
11          DESTINATION_GTT: gtt_core_engine_flp_base_url-${space}
12          HERE_MAP_KEY: {here-map-license-key}
13       parameters:
14          buildpack: java_buildpack
15          health-check-timeout: 180
16          host: gtt-sample-track-shipments-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       - name: lbn-gtt-sample-destination
25    resources:
26    - name: lbn-gtt-sample-app-uaa
27       type: org.cloudfoundry.existing-service
28    - name: lbn-gtt-sample-destination
29       type: org.cloudfoundry.existing-service
30
```

- o Replace {here-map-license-key} with the license key you got
  - o 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/sst/common/constant/Constants.java
  - o 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.sst.common.constant;
2
3    import java.math.BigDecimal;
4
5    public class Constants {
6
7        public static final String GTT_MODEL_NAMESPACE = "com.lbngttsamples.gtt.app.sof";
8        public static final String GTT_MODEL_NAMESPACE_WRITE_SERVICE = GTT_MODEL_NAMESPACE + ".sofWriteService";
9        public static final String MODEL_NAMESPACE = GTT_MODEL_NAMESPACE + ".sofService";
10       public static final String ENTITY_CONTAINER_NAME = "EntityContainer";
11       public static final String COMBINED_MODEL_PATH_SEGMENT = "combined-model";
12       public static final String SHIPMENT_MODEL_NAME = "Shipment";
13       public static final String ELEMENTS_PATH_SEGMENT = "elements";
14       public static final String REST_ROOT_URL = "/sap/logistics/gtt/sample/sst/rest/v1";
15       public static final String ODATA_ROOT_URL = "/sap/logistics/gtt/sample/sst/odata/v1";
16       public static final String DATE_TIME_PATTERN = "yyyy-MM-dd'T'HH:mm:ss'Z'";
17       public static final BigDecimal MAX_LONGITUDE = BigDecimal.valueOf(180);
18       public static final BigDecimal MAX_LATITUDE = BigDecimal.valueOf(90);
```

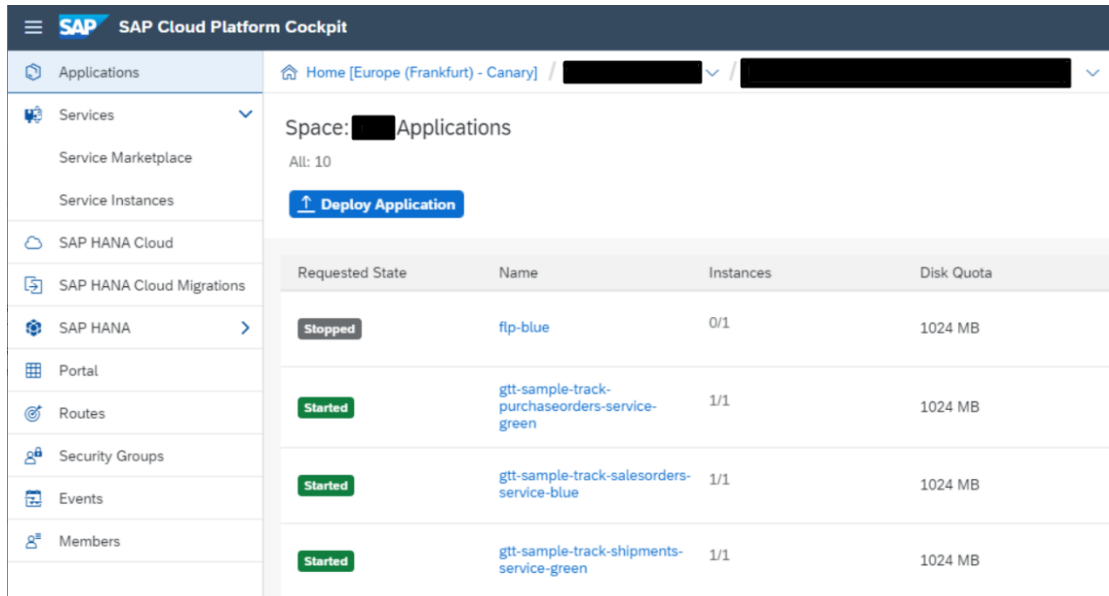- Run the following command to build the MTAR package:

```
mbt build
```

A MTAR file *./mta_archives/gtt-v2-sample-track-shipments-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-shipments-service_xxx.mtar
```

2.4.5 Check Deployed Apps on SAP Cloud Foundry Cockpit

After operations from Step 2.4.1 to Step 2.4.4, check the deployed apps on SAP Cloud Platform Cockpit.
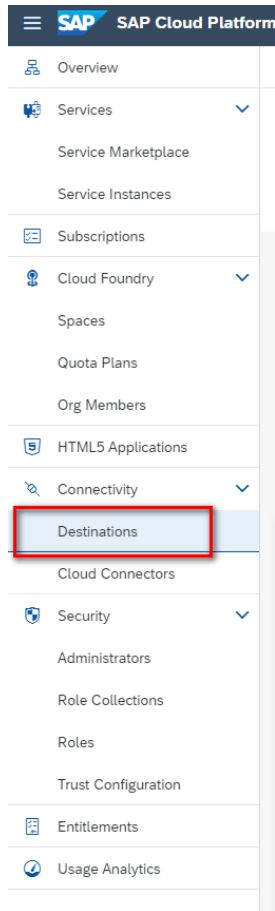
2.4.6 Maintain Destination of Forwarding

1. Go to "Destinations"

2. Configure Destination of Event to Action

Refer to Chapter 3 of the document 03_Synchronize_Actual_Events_To_TM-TS.pdf

Note:
If you only deploy the *Track Shipments* application, the part starting from "===> sync back event to TM system" in the *Event to Action* script is enough. The rest of the script is used for *Track Sales Orders* application. Replace "TO_QW9_170_STANDARD" with the destination name that you maintained in Step 2 and replace "QW9CLNT170" with the logical system of shipment tracked process.

```
341    ?"===> forwardCurrentTP to gtt_sof_forwarding-act";
342    forwardCurrentTP("gtt_sof_forwarding-act", false, false, true);
343  }
344
345    ?"===> sync back event to TM system";
346    ?"===> begin forwardEventToTM";
347
348    var eventTypeOfActualEvent = actualEvent.eventType;
349
350    if ("com.lbngttsamples.gtt.app.sof.Shipment.Departure" == eventTypeOfActualEvent ||
351         "com.lbngttsamples.gtt.app.sof.Shipment.Arrival" == eventTypeOfActualEvent ||
352         "com.lbngttsamples.gtt.app.sof.Shipment.Delay" == eventTypeOfActualEvent ||
353         "com.lbngttsamples.gtt.app.sof.Shipment.LoadingStart" == eventTypeOfActualEvent ||
354         "com.lbngttsamples.gtt.app.sof.Shipment.LoadingEnd" == eventTypeOfActualEvent ||
355         "com.lbngttsamples.gtt.app.sof.Shipment.POD" == eventTypeOfActualEvent ||
356         "com.lbngttsamples.gtt.app.sof.Shipment.POPU" == eventTypeOfActualEvent ||
357         "com.lbngttsamples.gtt.app.sof.Shipment.Coupling" == eventTypeOfActualEvent ||
358         "com.lbngttsamples.gtt.app.sof.Shipment.Decoupling" == eventTypeOfActualEvent ||
359         "com.lbngttsamples.gtt.app.sof.Shipment.UnloadingStart" == eventTypeOfActualEvent ||
360         "com.lbngttsamples.gtt.app.sof.Shipment.UnloadingEnd" == eventTypeOfActualEvent) {
361
362        var destinationName = "TO_QW9_170_STANDARD";
363        var reportedBy = null;
364        var senderPartyId = actualEvent.senderPartyId;
365        var trackingIdType = actualEvent.trackingIdType;
366        var partyId = trackedProcess.partyId;
367        var logicalSystemOfTp = trackedProcess.logicalSystem;
368
369        if ("QW9CLNT170" == logicalSystemOfTp && "SHIPMENT_ORDER" == trackingIdType) {
370            if (actualEvent.containsProperty("reportedBy")) {
371                reportedBy = actualEvent.reportedBy;
372            }
373            if (senderPartyId != partyId || reportedBy != null) {
374                forwardEventToTM(toObject("{}"), destinationName);
375            }
376        }
```

19

2.4.7 Maintain Roles and Users

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

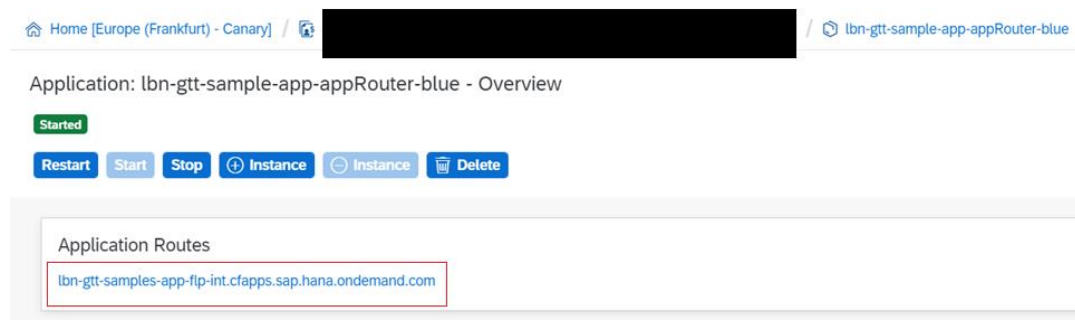| Role	Application
SalesOrderFulfillmentAppUser	tt_sof_{space}

Note that the role of "SalesOrderFulfillmentAppUser" is used to access the *Track Sales Orders* application.

If you need to access other template applications such as *Track Shipments* or *Track Purchase Orders*, add the following roles:
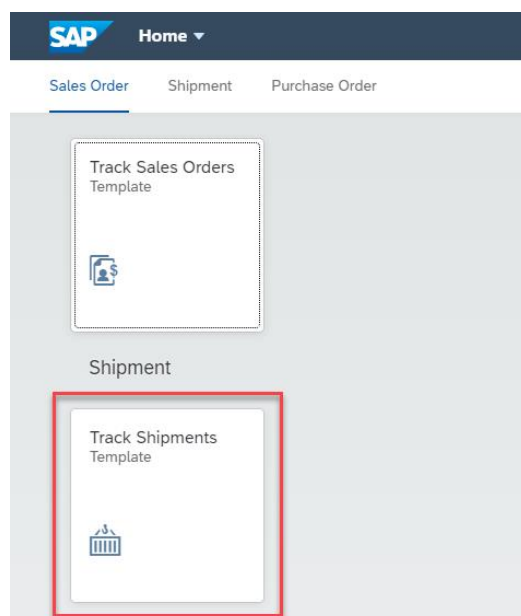
Role	Application
StandaloneShipmentTrackingAppUser	tt_sof_{space}
PurchaseOrderFulfillmentAppUser	tt_sof_{space}

2.4.8 Open *Track Shipments* 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 *Track Shipments*

- Open the *Track Shipments* template app

3. Implement ERP Extractor

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

04_Extractor_Creation_Guide-TS.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 *Track Shipments* 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.
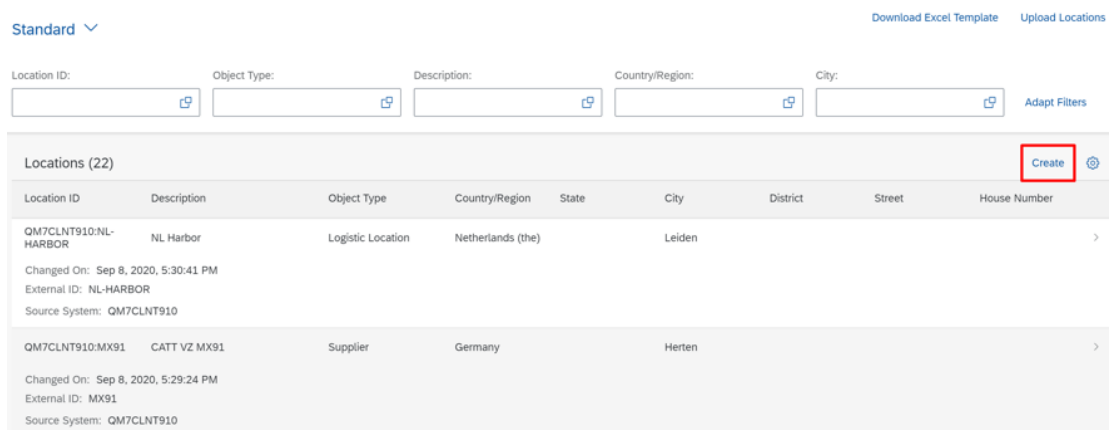
4.2 Maintain Location Master

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
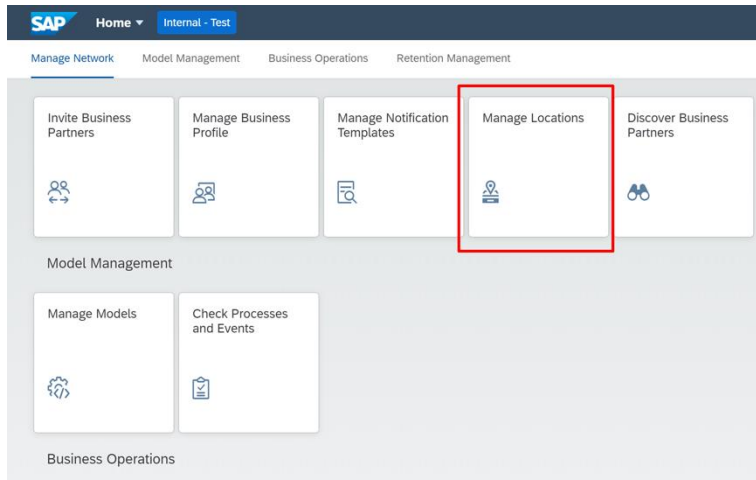
2. Click "Create" to create a new location

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:

04_Extractor_Creation_Guide-TS.pdf

4.3 Maintain Data Contributors

To receive actual events from other business partners, you need to invite them to join your business network. For more, search for *Administration Guide for Version 2* at help.sap.com/gtt and see the chapter "Onboarding and User Management (Data Contributors)".