Implementation Guide

- Track Sales Orders Template App

Contents

O.Prerequisites	2
1.Environment Configuration in SAP Business Network Global Track and Trace	2
1.1 Enable Cloud Foundry and Create Space	2
1.2 Maintain Roles and Users	4
1.3 Maintain Technical User	5
1.4 Maintain Destination of GTT Core Engine Base URL	8
2.Implement <i>Track Sales Orders</i> Template App	9
2.1 Download Template Code	9
2.2 Download and Setup Tools	9
2.3 Deploy Model	10
2.4 Package and Deploy	12
2.4.1 Login to SAP Cloud Foundry	12
2.4.2 Package and Deploy Track Sales Orders Frontend App	12
2.4.3 Package and Deploy SAP Fiori Launchpad for Template Applications	13
2.4.4 Package and Deploy Track Sales Orders Backend Service	20
2.4.5 Check Deployed Apps on SAP Cloud Foundry Cockpit	21
2.4.6 Maintain Destination of Forwarding	21
2.4.7 Maintain Roles and Users	24
2.4.8 Open Track Sales Orders Template App from SAP Fiori Launchpad	24
3. Implement ERP Extractor	25
4. Other Configuration	25
4.1 Obtain Here Map License	25
4.2 Maintain Location Master	26
4.3 Maintain Data Contributors	27
4.4 Integrate with Visibility Provider	27

0.Prerequisites

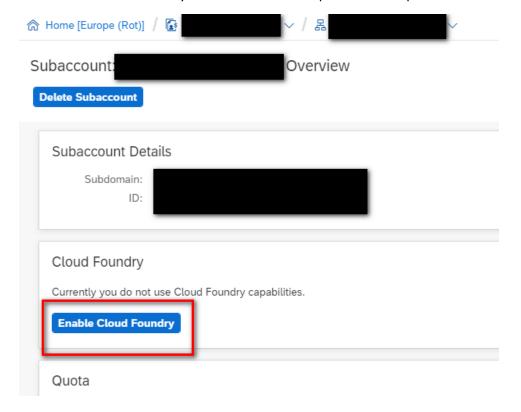
• You have an SAP Cloud Platform global account with the entitlement to SAP Business Network Global Track and Trace, and the account is assigned 1 portal service quota and 2 GB Application Runtime quota.

You can refer to the 1st Q&A in "FAQs for Template Code Implementation" for guidance on how to add quotas.

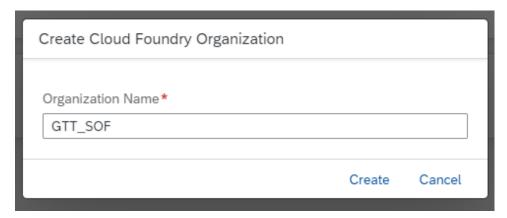
1.Environment Configuration in SAP Business Network Global Track and Trace

1.1 Enable Cloud Foundry and Create Space

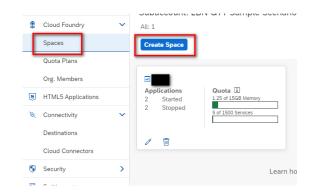
a. Click "Enable Cloud Foundry" to enable the development landscape.

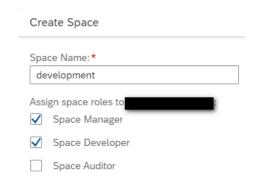


b. Fill in the Organization Name and click "Create"



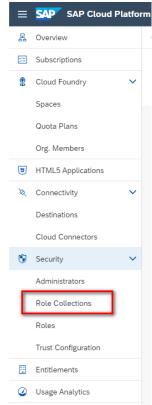
c. Go to "Cloud Foundry" -> "Spaces"Click "Create Space"Fill in the Space Name and click "Create"





1.2 Maintain Roles and Users

a. Go to "Security" -> "Role Collections"



 b. 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 SAP Business
 Network Global Track and Trace and Track Sales Orders 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 purposes.
- c. 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 Ibn-retentionmanagement-sb	
ReminderTemplateProcessTemplate	lbn-retentionmanagement-sb

- d. Grant "gtt-sample-all" to your **application** users:
 - Application users are those who have access to SAP Business Network Global Track and Trace and *Track Sales Orders* template app.

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

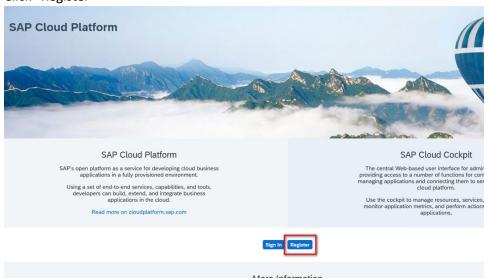


Note: The application user in this step should be different from the technical user created in Step 1.3. A user should **never** be assigned to **both** "gtt-sample-all" and "gtt-sample-integration".

1.3 Maintain Technical User

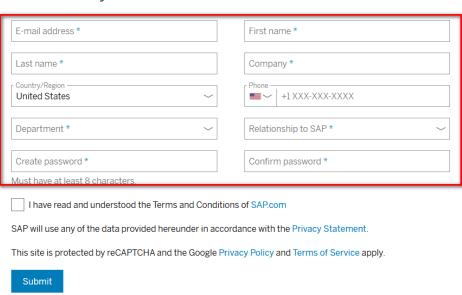
a. Go to the SAP Cloud Platform to create the technical user account: https://account.hana.ondemand.com/#/home/welcome



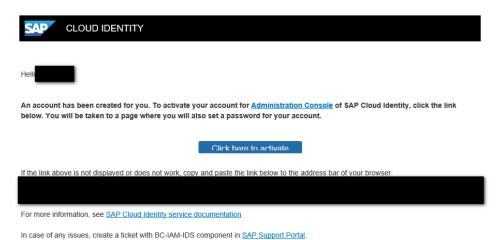


b. Fill in the mandatory fields:

Tell us about yourself



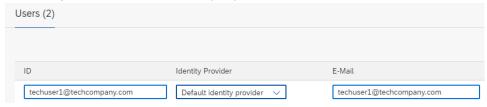
c. Go to the mail inbox to activate the account



d. 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

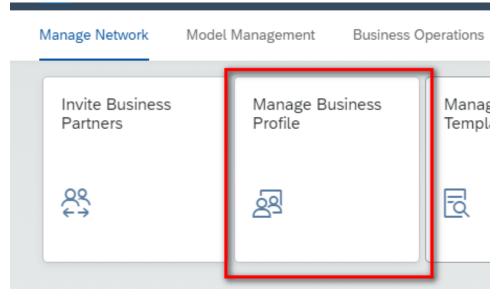
e. Grant "gtt-sample-integration" to your **technical** user: For example, <u>techuser1@techcompany.com</u>



Note:

In the "Identity Provider" field, select "Default identity provider".

f. Go to SAP Fiori Launchpad for SAP Business Network Global Track and Trace Open *Manage Business Profile* app



g. Add one Alternate Identifier:Identifier Type: Technical User

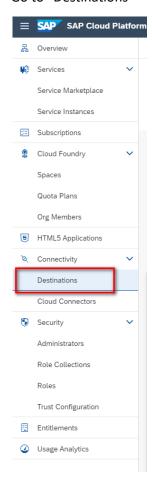
ID: the email address of technical user



Note: The technical user in this step should be different from the application user in Step 1.2.4. A user should **never** be assigned to **both** "gtt-sample-all" and "gtt-sample-integration".

1.4 Maintain Destination of GTT Core Engine Base URL

a. Go to "Destinations"



b. Configure Destination of GTT Core Engine Base URL

Name	gtt_core_engine_flp_base_url
Туре	HTTP
Description	<optional></optional>
URL	Enter the GTT Fiori Launchpad URL and add "/api" at the end of it.
	For example:
	https://xxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com <mark>/api</mark>
	For instructions on how to get the GTT Fiori Launchpad URL,
	see the note under this table.
Proxy Type	Internet
Authentication	BasicAuthentication
User	Email of Technical User from Step 1.3
Password	Password of Technical User from Step 1.3

Note:

- On your subaccount page, click "Instances and Subscriptions" in the "Service tab" and click "Global Track and Trace" application. Click "Go to Application".
- Copy the GTT Fiori Launchpad URL as shown in the picture:

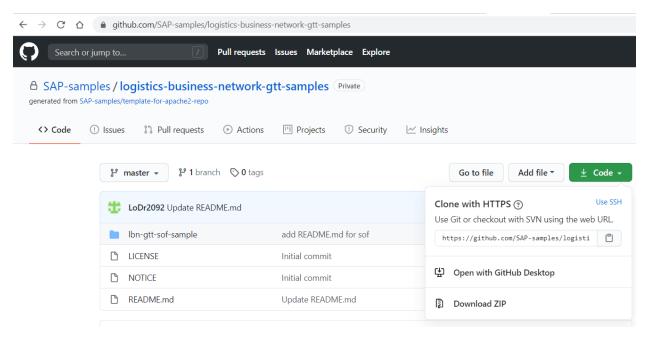


2.Implement Track Sales Orders Template App

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

a. Install JDK 8 and Maven

JDK 8: https://www.oracle.com/java/technologies/javase-jdk8-downloads.html Maven: https://maven.apache.org/download.cgi

Install NPM, Node JS 12, and grunt-cli
 Node JS 12: https://nodejs.org/dist/latest-v12.x/

For the instructions on how to install Node JS and NPM, see https://www.saplearners.com/how-to-install-node-js-and-npm/

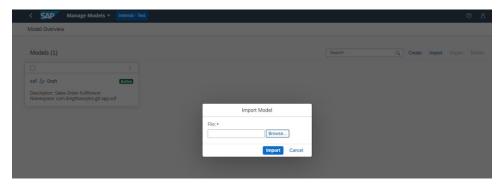
Run the following command after NPM is installed:

npm config set @sap:registry "https://npm.sap.com" -g

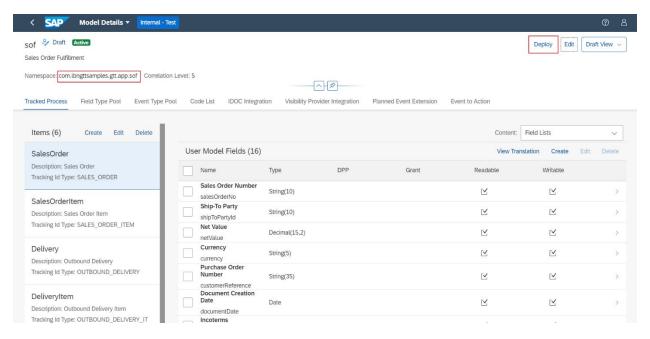
- Install Cloud Foundry Command Line Interface
 https://developers.sap.com/tutorials/cp-cf-download-cli.html
- d. Install MultiApps CF CLI Plugin https://github.com/cloudfoundry-incubator/multiapps-cli-plugin
- e. Install Cloud MTA Build Tool (MBT)
 https://sap.github.io/cloud-mta-build-tool/download/
- f. Install MAKE https://sap.github.io/cloud-mta-build-tool/makefile/

2.3 Deploy Model

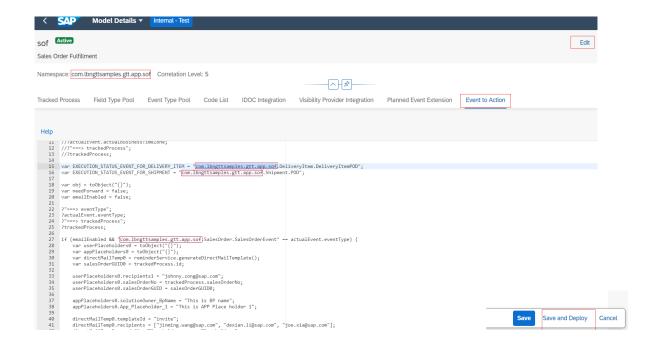
- Go to the folder *model* with downloaded source code. Make sure you have a JSON file *sof-model-YYYYMMDDxx.json* (e.g. sof-model-2020091001.json).
- Open Manage Models app on SAP Fiori Launchpad for SAP Business Network Global Track and Trace
- Click "Import"
- Click "Browse" to select the sof-model-YYYYMMDDxx.json 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

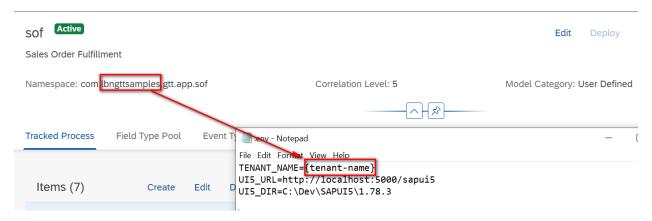
- Open a command window, cf login to SAP Cloud Foundry with your username and password.
- Select the organization and space in which you want to deploy the template apps.



2.4.2 Package and Deploy Track Sales Orders Frontend App

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

- Open the ./UI/.env file
- Replace {tenant-name} with the middle part of the namespace you got from Step 2.3.



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 *Template Applications* Go to GTT-V2-Sample-App-FLP directory.

- First ensure that at least 1 portal quota is assigned to the subaccount in which apps are to be deployed.
 - Please refer to the 1st Q&A in *FAQs for Template Code Implementation* for guidance on how to assign quotas.
- Open the *mta.yaml* file
- Replace {unique-identifier} with unique information such as your company name or cloud foundry global account name

Note:

- {unique-identifier} must only contain lowercase letters, numbers and hyphens. For example, "company-s01".
- o For the length of "route", the maximum limit is 63 characters.

```
1
        2
            ID: lbn-gtt-sample-app-flp
            _schema-version: "3.1"
            parameters:
        5
             deploy_mode: html5-repo
        6
               enable-parallel-deployments: true
        7
              append: "${space}"
              xsappname: tt_sof_${append}
        9
            version: 2.3.0
       10
            modules:
       11
              - name: lbn-gtt-sample-app-appRouter
                 type: approuter.nodejs
                path: flp_appRouter
                parameters:
       14
       15
                   disk-quota: 256M
                  memory: 256M
                  routes:
                     - route: lbn-gtt-samples-app-flp-{unique-identifier} ${domain}
                properties:
       20
                  SAP_JWT_TRUST_ACL:
                     - clientid: "*"
                       identityzone: "*"
       22
                   destinations:
       2.4
                     - name: 1hn-att-sof-ani
resources:
 - name: lbn-gtt-sample-app-runtime
   parameters:
    service-plan: app-runtime
     service: html5-apps-repo
   type: org.cloudfoundry.managed-service
  - name: lbn-gtt-sample-portal-resources
   parameters:
    service-plan: standard
    service: portal
   type: org.cloudfoundry.managed-service
 - name: lbn-gtt-sof-api
   properties:
    url: https://gtt-sample-track-salesorders-service-{unique-identifier},${default-domain}
 - name: lbn-gtt-sst-api
   properties:
     url: https://gtt-sample-track-shipments-service-{unique-identifier} ${default-domain}
 - name: 1bn-gtt-pof-api
   properties:
     url: https://gtt-sample-track-purchaseorders-service-{unique-identifier},${default-domain}
 - name: lbn-gtt-sample-app-uaa
   type: com.sap.xs.uaa-application
   parameters:
```

This FLP project includes 3 template applications which are Track SO Fulfillment, Track Shipments and Track Purchase Orders. Before you build and deploy it, please deploy all the UI projects you need in advance, otherwise the deployment will fail.

If you only need Track SO Fulfillment app, do the following changes:

- Open the *mta.yaml* file
- Delete *lbn-gtt-sst-api* and *lbn-gtt-pof-api* in *lbn-gtt-sample-app-appRouter* and *resources* modules
- Delete gtt-ui-sample-track-shipments-app-host and gtt-ui-sample-track-purchaseorders-app-host in flp and resources modules

```
ID: lbn-gtt-sample-app-flp
_schema-version: "3.1"
parameters:
     enable-parallel-deployments: true
       append: "${space}"
     xsappname: tt_sof_${append}
version: 2.3.0
modules:
     - name: lbn-gtt-sample-app-appRouter
            path: flp_appRouter
           parameters:
                disk-quota: 256M
                memory: 256M
                       - route: lbn-gtt-samples-app-flp-{unique-identifier}.${domain}
            properties:
                 SAP_JWT_TRUST_ACL:
- clientid: "*"
                            identityzone: "*"
                 destinations:
                       url: "~{lbn-gtt-sof-api/url}"
forwardAuthToken: true
- name: lbn-gtt-sst-api
                          url: "~{lbn-gtt-sst-api/url}"
                            forwardAuthToken: true
                    - name: lbn-gtt-pof-api
                         url: "~{lbn-gtt-pof-api/url}"
                            forwardAuthToken: true
                - name: 1bn-gtt-sample-app-uaa
                - name: 1bn-gtt-sample-app-runtime
                  - name: lbn-gtt-sample-portal-resources
                - name: lbn-gtt-sof-api
- name: lbn-gtt-sst-api
         - name: lbn-gtt-pof-api
            type: com.sap.portal.content
            path: flp
                stack: cflinuxfs3
                  build pack: \ \texttt{https://github.com/cloudfoundry/nodejs-build pack/releases/download/v1.6.39/nodejs-build pack-cflinux fs3-v1.6.39.zip the state of the state of
            requires:
              - name: lbn-gtt-sample-app-uaa
                - name: 1bn-gtt-sample-portal-resources
              - name: gtt-ui-sample-track-salesorders-app-host
- name: gtt-ui-sample-track-shipments-app-host
                     - name: gtt-ui-sample-track-purchaseorders-app-host
```

```
- name: 1bn-gtt-sample-app-runtime
 parameters:
   service-plan: app-runtime
service: html5-apps-repo
  {\sf type: org.cloudfoundry.managed-service}
- name: 1bn-gtt-sample-portal-resources
   service-plan: standard
    service: portal
  {\tt type: org.cloudfoundry.managed-service}
 name: lbn-gtt-sof-api
  properties:
   url: https://gtt-sample-track-salesorders-service-{unique-identifier}.${default-domain}
  properties:
   url: https://gtt-sample-track-shipments-service-{unique-identifier}.${default-domain}
  name: lbn-gtt-pof-api
  properties:
 url: https://gtt-sample-track-purchaseorders-service-{unique-identifier}.${default-domain}
name: lbn:gtt-sample-app-uaa
  type: com.sap.xs.uaa-application
  parameters:
config:
      xsappname: "${xsappname}"
      tenant-mode: dedicated
      oauth2-configuration:
       system-attributes:
          - rolecollections
      description: Security profile of sales order fulfillment application
      scopes:
       - name: "$XSAPPNAME.sof.r"
         description: Allow to access sales order fulfillment application
       - name: "$XSAPPNAME.pof.r"
       description: Allow to access purchase order fulfillment application - name: "$XSAPPNAME.sst.r"
       description: Allow to access standalone shipment tracking application - name: "$XSAPPNAME.sst.w"
         description: Allow to report events in standalone shipment tracking application
        - name: uaa.user
          description: UAA
      authorities:
- "$XSAPPNAME.sof.r"
        - "$XSAPPNAME.pof.r"
        - "$XSAPPNAME.sst.r"
        - "$XSAPPNAME.sst.w"
        - uaa.user
      role-templates:
       - name: SalesOrderFulfillmentAppUserDisplayTemplate
          description: Track & Trace Sales Order Fulfillment Template App User Read Access
           - "$XSAPPNAME.sof.r"
- waa.wser
```

```
- name: "$XSAPPNAME.sst.r"
                   description: Allow to access standalone shipment tracking application
              - name: "$XSAPPNAME.sst.w"
                  description: Allow to report events in standalone shipment tracking application
           authorities:
                - "$XSAPPNAME.sof.r"
               - "$XSAPPNAME.pof.r"
- "$XSAPPNAME.sst.r"
               - "$XSAPPNAME.sst.w"
100
           role-templates:

    - name: SalesOrderFulfillmentAppUserDisplayTemplate
    description: Track & Trace Sales Order Fulfillment Template App User Read Access

104
                     - uaa.user

    - uear.user
    - name: PurchaseOrderFulfillmentAppUserDisplayTemplate
    description: Track & Trace Purchase Order Fulfillment Template App User Read Access
    scope-references:

                  - "$XSAPPNAME.pof.r"
- uaa.user
          - uaa.user

- name: StandaloneShipmentTrackingAppUserDisplayTemplate

description: Track & Trace Standalone Shipment Tracking Template App User Read Access

scope-references:
                    - "$XSAPPNAME.sst.r
                     - uaa.user
           - name: StandaloneShipmentTrackingAppUserProcessTemplate
                  description: Track & Trace Standalone Shipment Tracking Template App User Write Access
                  scope-references:
                     - "$XSAPPNAME.sst.w'
                     - uaa.user
       - name: gtt-ui-sample-track-salesorders-app-host
           type: org.cloudfoundry.existing-service
         - name: gtt-ui-sample-track-shipments-app-host
         type: org.cloudfoundry.existing-service
        - name: gtt-ui-sample-track-purchaseorders-app-host
          type: org.cloudfoundry.existing-service
          name: lbn-gtt-sample-destination
          type: org.cloudfoundry.managed-service
          parameters:
            service-plan: lite
         service: destination
```

- Open the flp/portal-site/CommonDataModel.json file
- Delete viz items com.sap.gtt.app.sample.sst and com.sap.gtt.app.sample.pof in payload/catalogs/payload/viz array
- Delete groups shipments and purchaseOrder in payload/groups array
- Delete orders shipments and purchaseOrder in payload/sites/payload/groupsOrder array

```
"_version": "3.0.0",
"identification": {
  "id": "ee7d30df-c86c-48b1-8f02-c2abc4e18f54-1578021444073",
   "entityType": "bundle"
"payload": {
  "catalogs": [
   {
    "_version": "3.0.0",
    "^*=+ion": {
      "identification": {
    "id": "defaultCatalogId",
         "title": "{{title}}",
         "entityType": "catalog",
"i18n": "i18n/defaultCatalogId.properties"
       "payload": {
         "viz": [
              "id": "com.sap.gtt.app.sample.sof",
              "vizId": "SalesOrder-track"
              "id": "com.sap.gtt.app.sample.sst",
"vizId": "Shipment-track"
              "id": "com.sap.gtt.app.sample.pof",
              "vizId": "PurchaseOrder-track"
    }
  "groups": [
       "_version": "3.0.0",
       "identification": {
         "id": "salesOrder",
         "title": "{{title}}",
"entityType": "group",
         "i18n": "i18n/salesOrder.properties"
         "viz": [
              "id": "com.sap.gtt.app.sample.sof",
              "appId": "com.sap.gtt.app.sample.sof",
"vizId": "SalesOrder-track"
        1
```

```
"_version": "3.0.0",
    "identification": {
      "id": "shipment",
      "title": "{{title}}",
      "entityType": "group",
     "i18n": "i18n/shipment.properties"
   },
    "payload": {
      "viz": [
          "id": "com.sap.gtt.app.sample.sst",
          "appId": "com.sap.gtt.app.sample.sst",
          "vizId": "Shipment-track"
 },
   "identification": {
     "id": "purchaseOrder",
"title": "{{title}}",
"entityType": "group",
     "i18n": "i18n/purchaseOrder.properties"
   "payload": {
     "viz": [
          "id": "com.sap.gtt.app.sample.pof",
          "appId": "com.sap.gtt.app.sample.pof",
          "vizId": "PurchaseOrder-track"
     ]
"sites": [
   "identification": {
     "id": "a3eee102-5181-4222-9924-8b8e90985f07-1578021444073",
     "entityType": "site",
"title": "SAP Fiori launchpad site on Cloud Foundry",
     "description": "SAP Fiori launchpad site on Cloud Foundry, deployed from SAP Web IDE"
   "payload": {
      "config": {
        "ushellConfig": {
```

```
"sites": [
91
             "_version": "3.0.0",
           "identification": {
94
              "id": "a3eee102-5181-4222-9924-8b8e90985f07-1578021444073",
              "entityType": "site",
             "title": "SAP Fiori launchpad site on Cloud Foundry",
              "description": "SAP Fiori launchpad site on Cloud Foundry, deployed from SAP Web IDE"
           },
100
            "payload": {
L01
              "config": {
102
               "ushellConfig": {
                 "renderers": {
104
                    "fiori2": {
                      "componentData": {
                        "config": {
                          "applications": {
                            "Shell-home": {}
L09
                          },
110
                          "enableHelp": false,
                          "enableSearch": false,
                          "sessionTimeoutIntervalInMinutes": 19,
                          "sessionTimeoutReminderInMinutes": 3
                    }
                  }
              }
              }.
120
              "groupsOrder": [
                "shipment",
               "purchaseOrder"
124
```

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 Sales Orders Backend Service

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

• Open the file *mta.yaml* under the folder "GTT-V2-Sample-TrackSalesOrders-Service":

```
mta.yaml
       schema-version: '3.1'
     ID: gtt-v2-sample-track-salesorders-service
     version: 2.3.0
     modules:
  6 F- name: gtt-sample-track-salesorders-service
       type: java
       path: srv/
    properties:
         SPRING_PROFILES_ACTIVE: cloud
         DESTINATION_GTT: gtt_core_engine_flp_base_url
 11
         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:
         ignore:
 19
          - "*.md"
 20
         build-result: target/*.jar
         timeout: 10m
       requires:
 23
        - name: lbn-gtt-sample-app-uaa
 24
 25
     resources:
 26 = name: lbn-gtt-sample-app-uaa
 27
        type: org.cloudfoundry.existing-service
 28
 29
```

- Replace {here-map-license-key} with the license key that you can get from step 4.1.
 Note: This step is optional, you could ignore this to proceed to verify if the application could be built and pushed successfully. After the actual license key is obtained, you could come back and replace it and execute this step again to build and push the application to cloud so that the map could be shown.
- Replace {unique-identifier} with the unique information such as your company name or cloud foundry global account name.

Note:

- {unique-identifier} must only contain lowercase letters, numbers and hyphens. For example, "company-s01".
- o For the length of "host", the maximum limit is 63 characters.
- 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 🗵
      package com.sap.gtt.v2.sample.sof.constant;
    □public class Constants {
          public static final String GTT MODEL NAMESPACE = "com.lbngttsamples.gtt.app.sof
          public static final String GTT MODEL NAMESPACE WRITE SERVICE = GTT MODEL NAMESPACE + ".sofWriteServic
          public static final String MODEL_NAMESPACE = GTT_MODEL_NAMESPACE + ".sofService";
  6
          public static final String URL SPLITTER = "/";
  8
          public static final String ENTITY CONTAINER NAME = "EntityContainer";
  9
          public static final String SAP_ICON_LEAD = "sap-icon://lead";
 11
          public static final String ID = "id";
 12
          public static final String EVENT_STATUS_PLANNED = "PLANNED";
 14
```

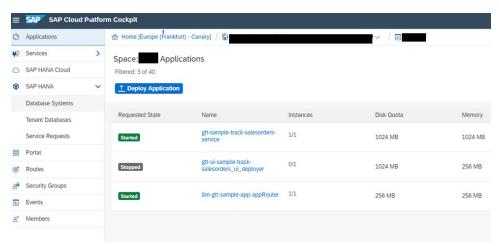
Run the following command to build the MTAR package:
 mbt build

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.

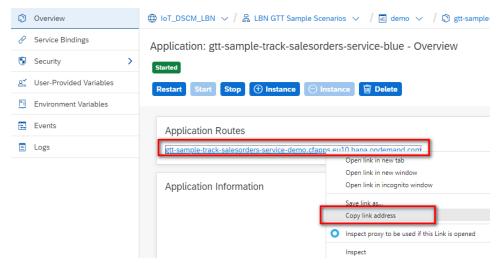


2.4.6 Maintain Destination of Forwarding

- a. Obtain the UAA Token
- Go to the Track Sales Orders backend application that you have deployed

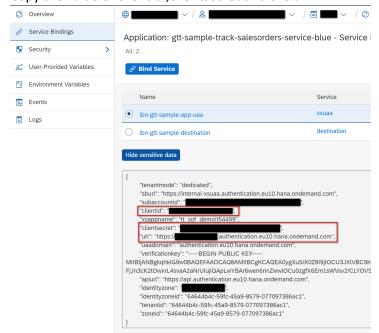


Copy the link address of application routes

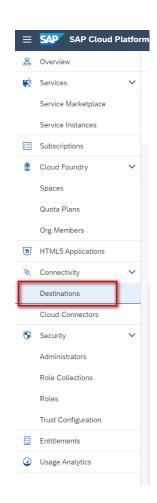


Find the environment of UAA

Copy the value of clientid, clientsecret and the url



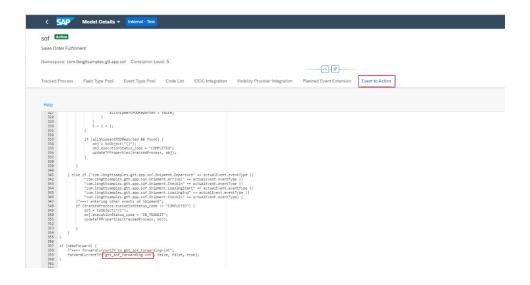
b. Go to "Destinations"



c. Configure Destination of Event to Action Forwarding

Name	e.g. gtt_sof_forwarding-int	
Туре	HTTP	
Description	<optional></optional>	
URL	Enter the link address of application routes copied in the previous step and	
	add "/sap/logistics/gtt/sample/sof/rest/v1/forward" at the end of it.	
	For example:	
	https://gtt-sample-track-salesorders-	
	service.cfapps.eu10.hana.ondemand.com/sap/logistics/gtt/sample/sof/rest/v1/	
	forward	
Proxy Type	Internet	
Authenticati	OAuth2ClientCredentials	
on		
Client ID	Copied in the previous step	
Client Secret	Copied in the previous step	
Token	Enter the URL copied in previous step and add "/oauth/token" at the end of the	
Service URL	it.	
	For example:	
	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 *Track Sales Orders* model if you change the name from "gtt_sof_forwarding-int" to others.



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
SalesOrderFulfillmentAppUserDisplayTemplate	tt_sof_{space}

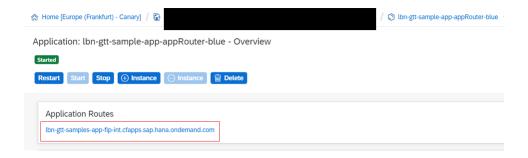
Note: the role of "SalesOrderFulfillmentAppDisplayTemplate" is used to access the *Track SO Fulfillment* application.

If you need to access other template applications such as *Track Shipments* or *Track PO Fulfillment*, add the following roles:

Role	Application
Standal one Shipment Tracking App User Display Template	tt_sof_{space}
Standal one Shipment Tracking App User Process Template	tt_sof_{space}
PurchaseOrderFulfillmentAppUserDisplayTemplate	tt_sof_{space}

2.4.8 Open Track Sales Orders Template App from SAP Fiori Launchpad

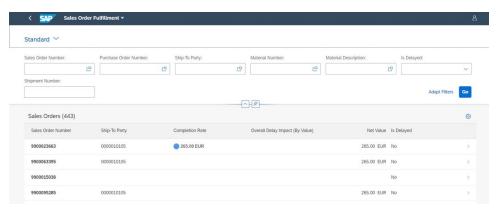
• Obtain the Application Routes from SAP Cloud Foundry Cockpit



Click the link of Application Routes
 The system browser then navigates to the SAP Fiori Launchpad of *Track Sales Orders*



• Open the Track Sales Orders template app



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 template apps in SAP support system under component "SBN-LBN-GTT-APP". And follow the 2 steps below when you create an incident:

- Add logon credentials in the incident to the secure area so that we can provide you the key in a safe way. You can refer to SAP Note 1773689 for reference: https://launchpad.support.sap.com/#/notes/0001773689
- 2) Provide us with the application routes which you get in step 2.4.8 "Obtain the Application Routes from SAP Cloud Foundry Cockpit"

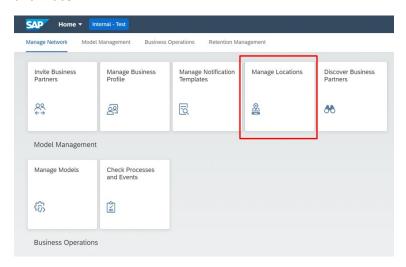
After you get the key, you can use it in Step 2.4.4 to re-deploy the backend service.

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

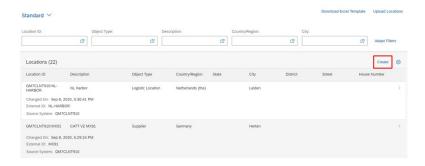
4.2 Maintain Location Master

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

• Open *Manage Locations* app in your SAP Fiori Launchpad for SAP Business Network Global Track and Trace



a. Click "Create" to create a new location

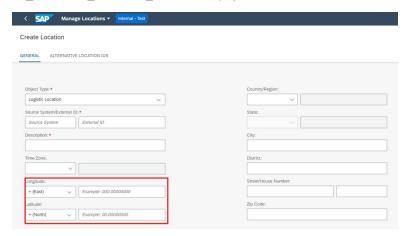


b. Maintain location information with valid geocoordinates

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. For more, search for *Administration Guide for Data Contributors* at help.sap.com/gtt and see the chapters "Onboarding" and "User Management'.

4.4 Integrate with Visibility Provider

To integrate with visibility provider, log your incident in SAP Support system with the component "SBN-LBN-GTT-APP".