Implementation Guide – *Track Shipments* Template App

Contents

Note

The SAP Cloud Platform brand is retired. For more information, see A Robust Platform for RISE with SAP . Parts of this content were created before our branding changes related to SAP technology were announced in January 2021.

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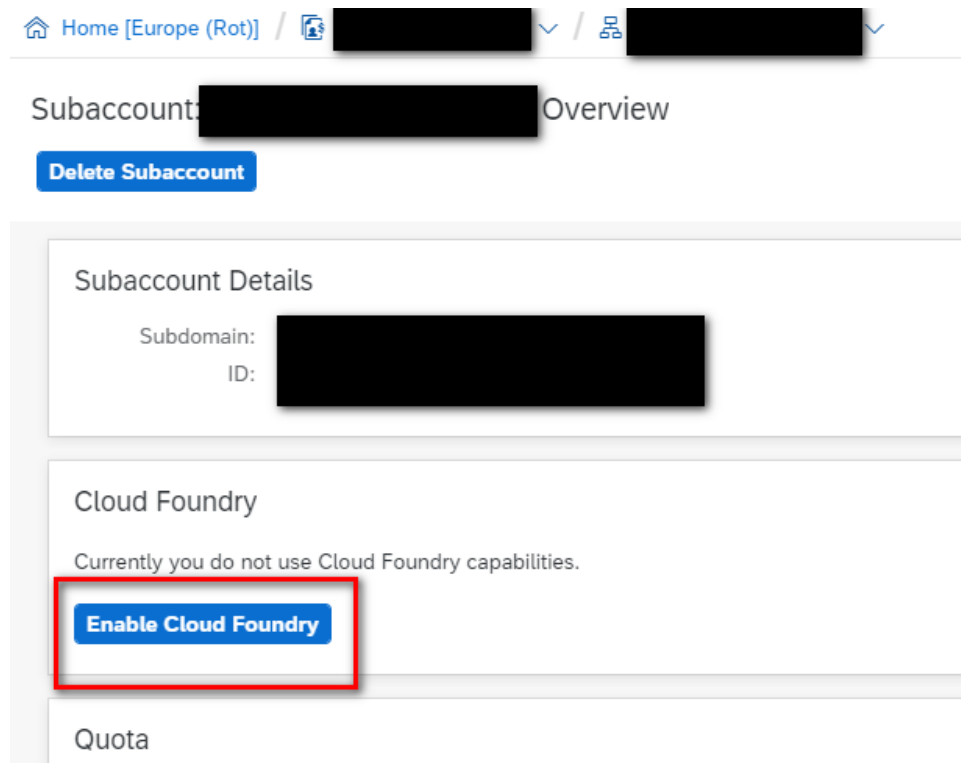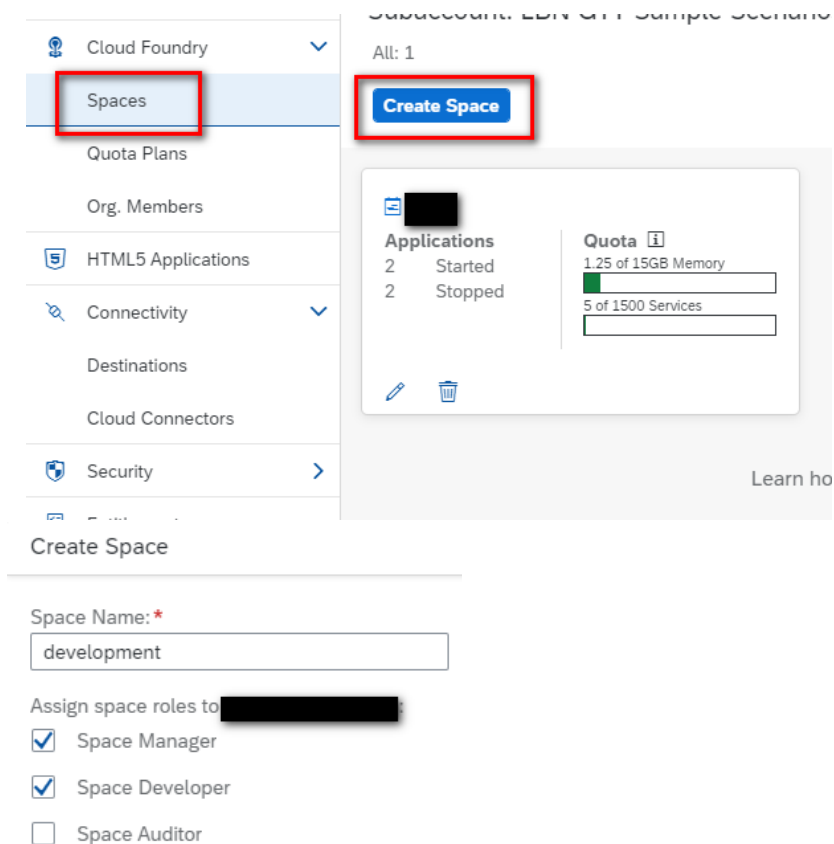
2. Fill in the Organization Name and click "Create"

Create Cloud Foundry Organization

Organization Name*

GTT_APPS

Create   Cancel

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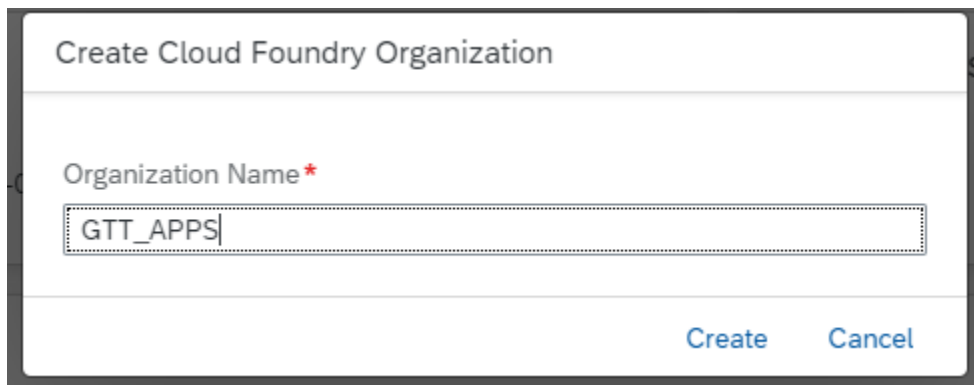   Fill in the Space Name and click "Create"

Cloud Foundry	∨
Spaces	
Quota Plans	
Org. Members	
HTML5 Applications	
Connectivity	∨
Destinations	
Cloud Connectors	
Security	>

All: 1

Create Space

Applications
2   Started
2   Stopped

Quota ⓘ
1.25 of 15GB Memory

5 of 1500 Services

Learn ho

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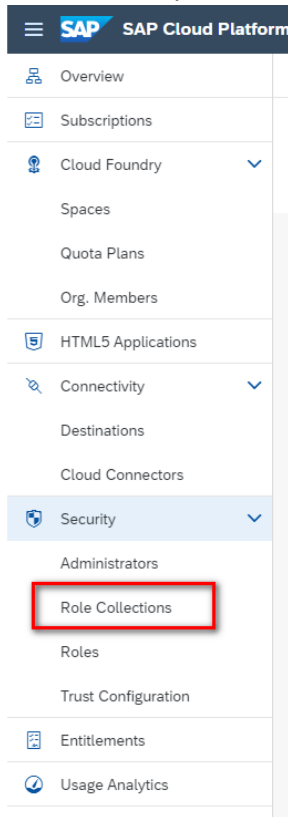
2. Create 2 role collections (You can have user-defined names):
    gtt-sample-all
    gtt-sample-integration
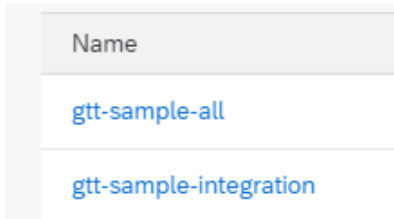
-   Role template "gtt-sample-all" is granted to users who have access to the global track and trace option and the *Track Shipments* template app.
-   Role template "gtt-sample-integration" is granted to technical users who only have access to the global track and trace APIs for development purpose.
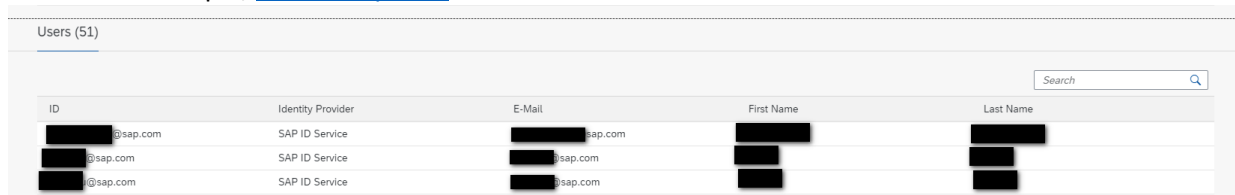
3. Go to the role of "gtt-sample-all", and add the following roles:

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 the global track and trace option and the *Track Shipments* template app.

   IT Administrator needs to maintain application users in the 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.

1.3 Maintain Technical User

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

   Click "Register"
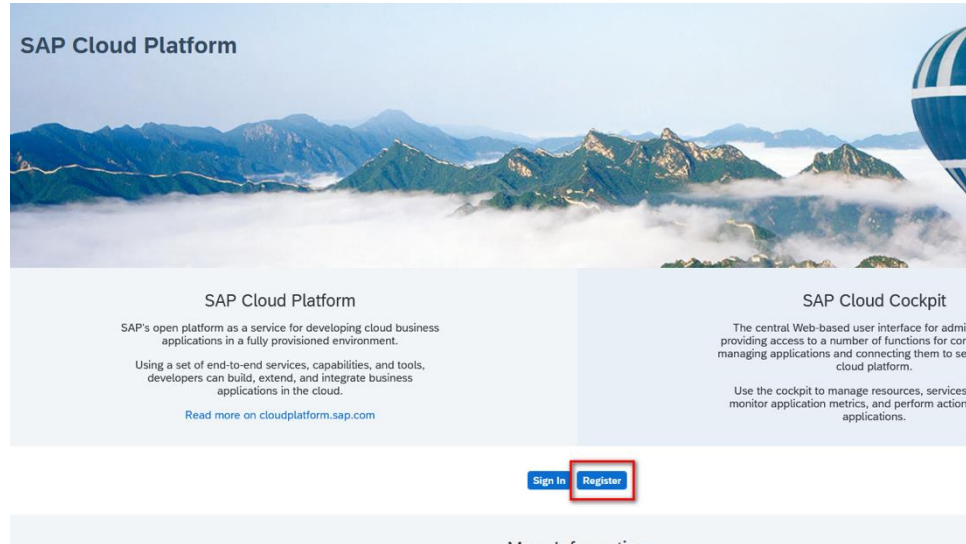
2. Fill in the mandatory fields:

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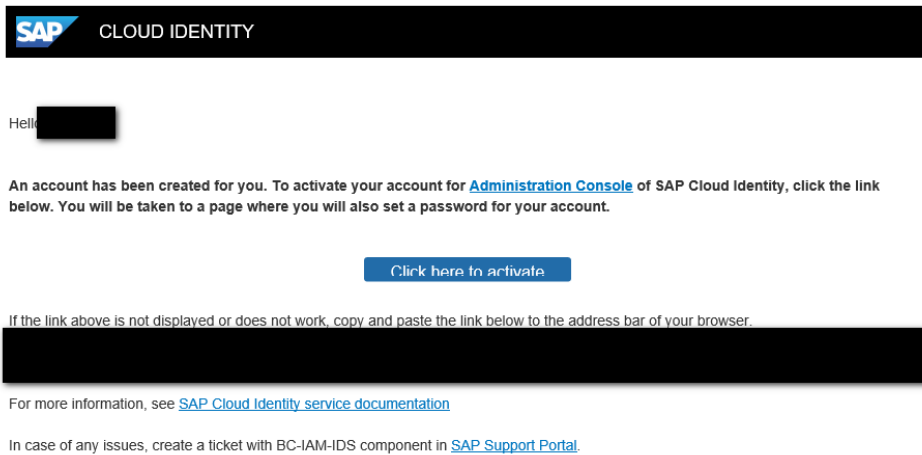
4. As shown in Step 1.2, go to the role of "gtt-sample-integration", and add the following roles:

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
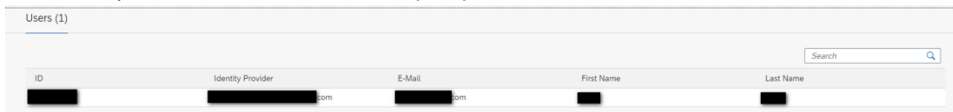
6. Go to SAP Fiori Launchpad for the global track and trace option.
   Open the *Manage Business Profile* app.
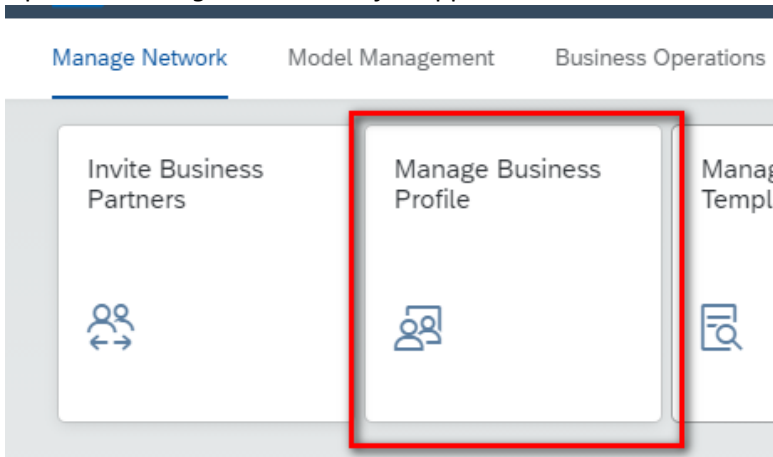
7. Add one Alternate Identifier:
   Identifier Type: Technical User
   ID: the email address of technical user

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"
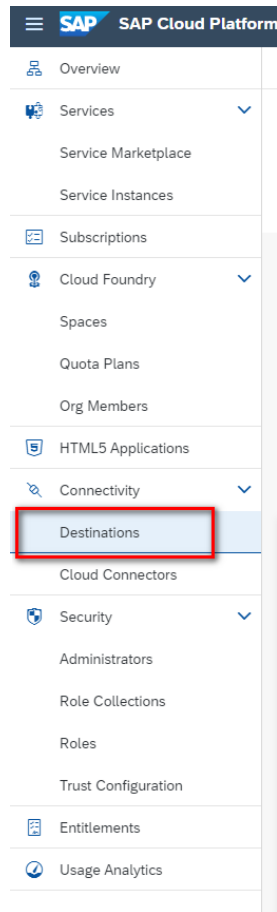
- Configure Destination of GTT Core Engine Base URL

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.
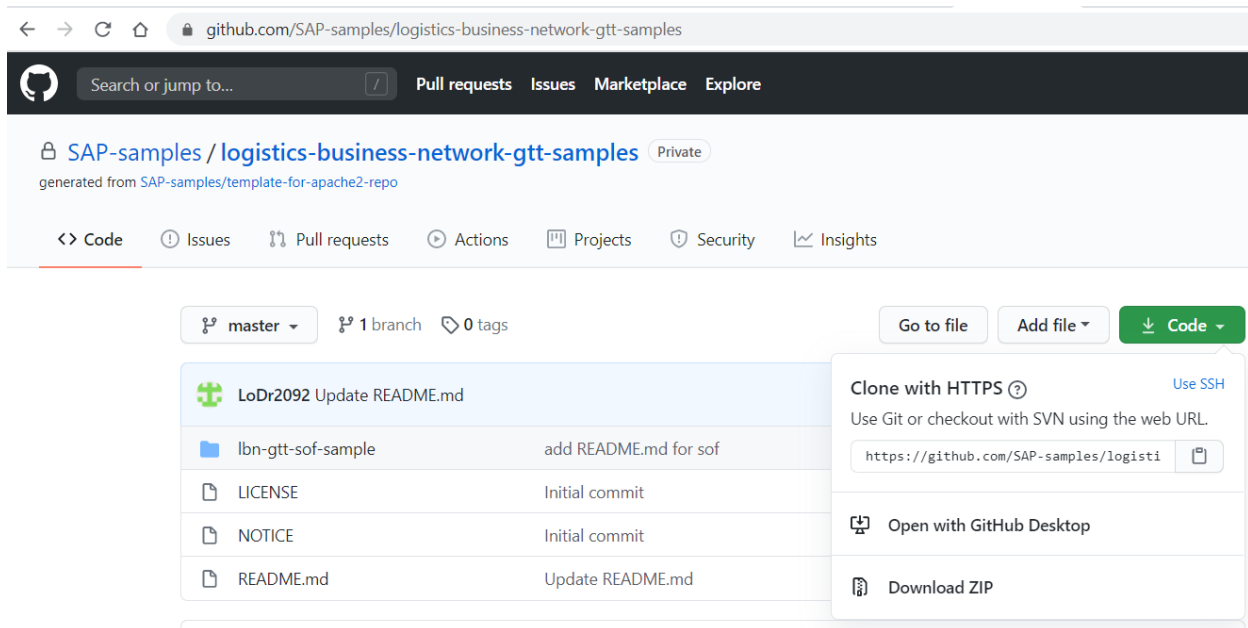
2.2 Download and Setup Tools

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 the *Manage Models* app on SAP Fiori Launchpad for the global track and trace option
- Click "Import"
- Click "Browse" to select the *sof-model-YYYYMMDDxx.json* file
- Click "Import" to import the model
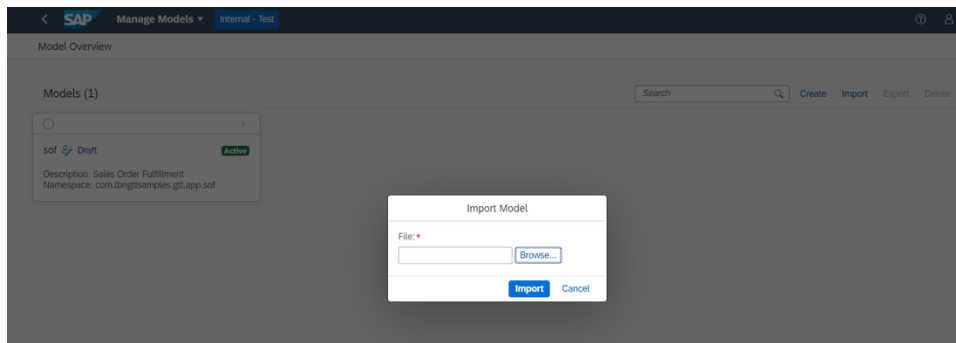
- Click "Deploy"
- Note down the model namespace for later use, in this case, "com.lbngttsamples.gtt.app.sof" as shown below
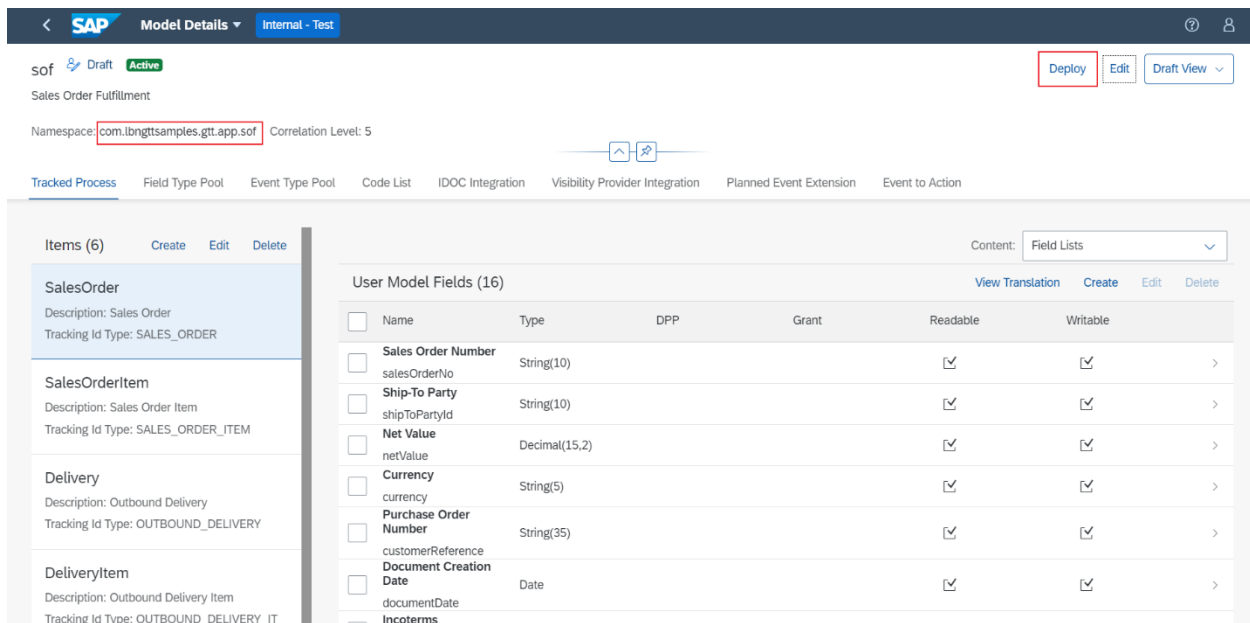
- Click "Edit"
- Go to *Event to Action* tab
- Replace all the namespace strings, in this case, "com.lbngttsamples.gtt.app.sof" in the *Event to Action* script with the model namespace you noted down in the previous step
- Click "Save and Deploy" to deploy the model
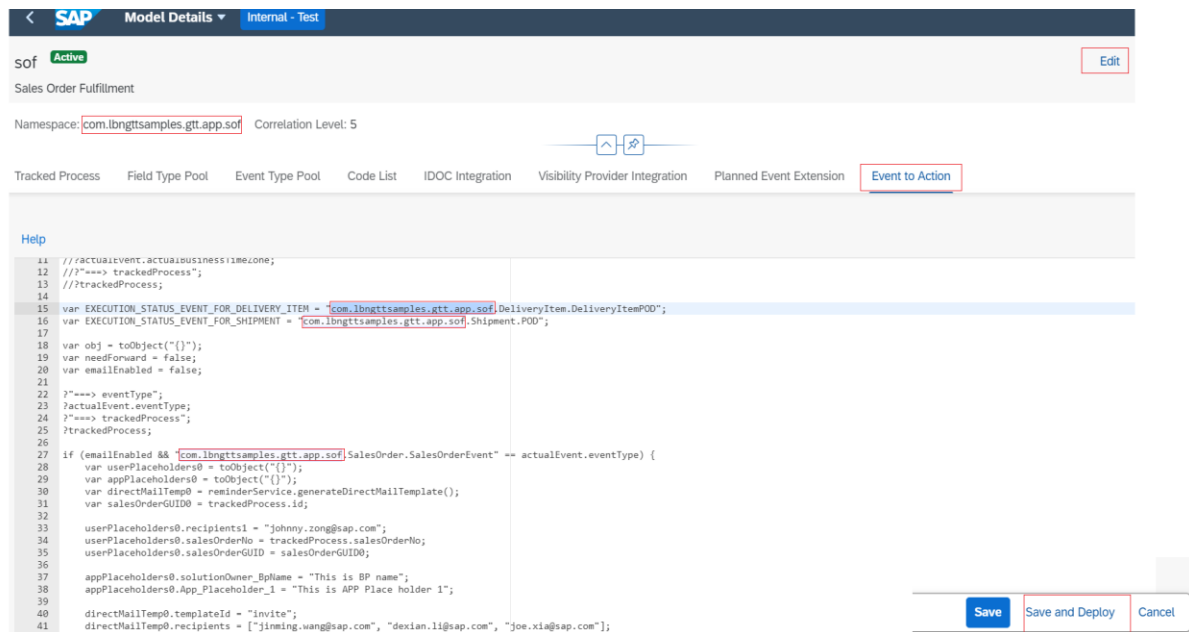
2.4 Package and Deploy

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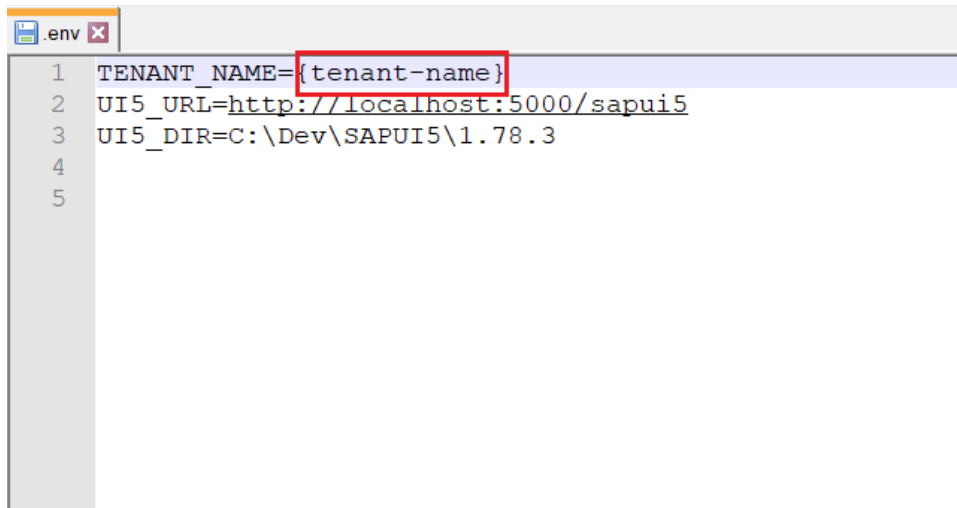
- Open a command window, *cf login* to SAP Cloud Foundry with your username and password
- Select the organization and space in which you want to deploy the template apps
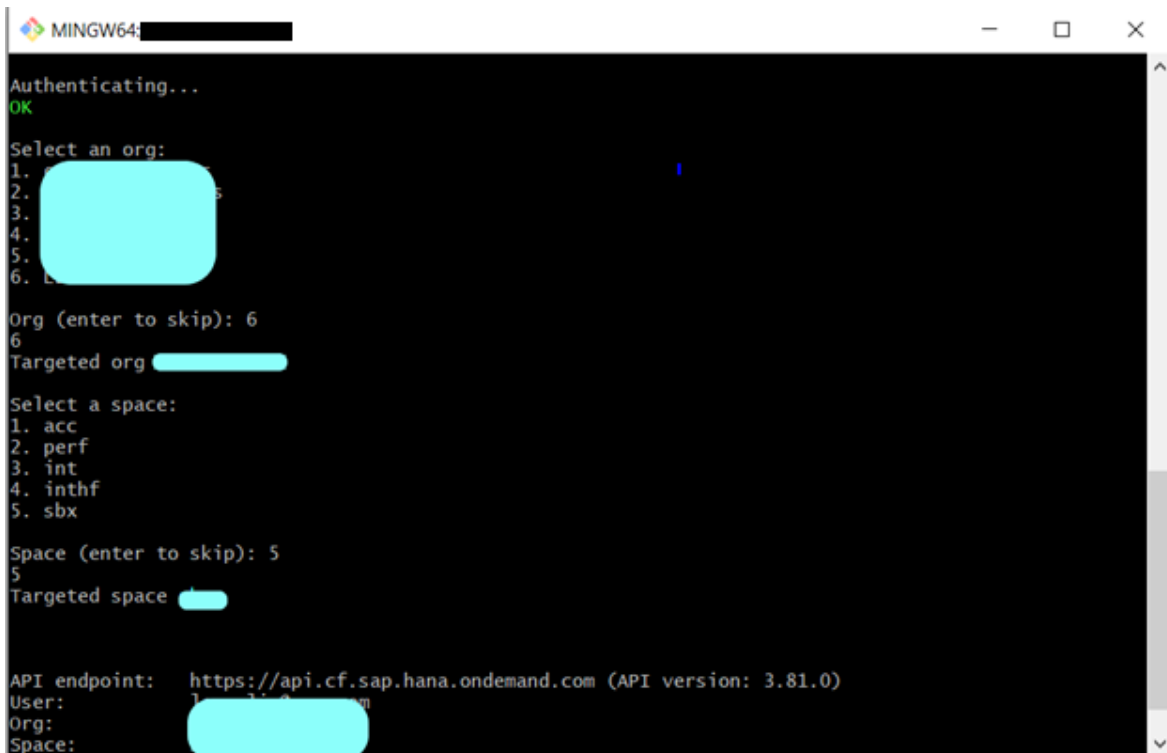
2.4.2 Package and Deploy *Track Shipments* Frontend App

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

- Open the *./UI/.env* file
- Replace {tenant-name} with part of the namespace you got from <u>Step 2.3</u>.
  For example, replace it with "lbngttsamples" in "com.lbngttsamples.gtt.app.sof" as 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.
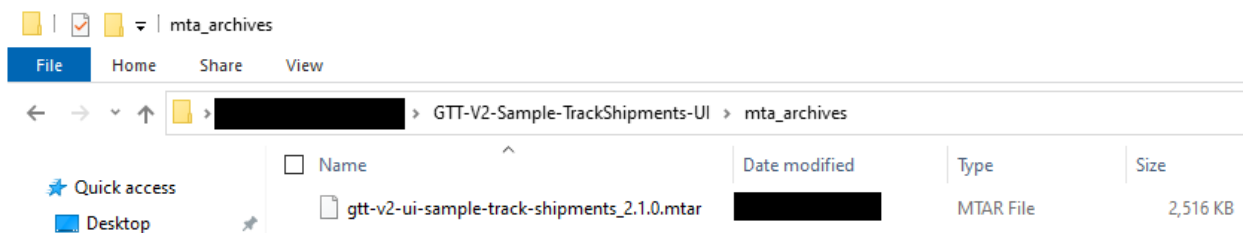
- To deploy the MTAR package in Cloud Foundry, run the following command:

```
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 *Template Apps*

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

```
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: 2.3.0
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
```

```
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: 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:
```

This FLP project includes 3 template apps: Track SO Fulfillment, Track Shipments and Track PO Fulfillment. Before you build and deploy it, deploy all the UI projects you need in advance. Otherwise, the deployment of this project will fail.

If you only need the *Track Shipments* template app, make the following changes:

- Open the *mta.yaml* file
- Delete *lbn-gtt-sof-api* and *lbn-gtt-pof-api* in l*bn-gtt-sample-app-appRouter* and *resources* modules
- Delete *gtt-ui-sample-track-salesorders-app-host* and *gtt-ui-sample-track-purchaseorders-app-host* in *flp* and *resources* module

```yaml
---
ID: lbn-gtt-sample-app-flp
_schema-version: "3.1"
parameters:
  deploy_mode: html5-repo
  enable-parallel-deployments: true
  append: "${space}"
  xsappname: tt_sof_${append}
version: 2.3.0
modules:
  - name: lbn-gtt-sample-app-appRouter
    type: approuter.nodejs
    path: flp_appRouter
    parameters:
      disk-quota: 256M
      memory: 256M
      routes:
        - route: lbn-gtt-samples-app-flp-{unique-identifier}.${domain}
    properties:
      SAP_JWT_TRUST_ACL:
        - clientid: "*"
          identityzone: "*"
      destinations:
        - name: lbn-gtt-sof-api
          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
    requires:
      - name: lbn-gtt-sample-app-uaa
      - name: lbn-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
  - name: flp
    type: com.sap.portal.content
    path: flp
    parameters:
      stack: cflinuxfs3
      memory: 128M
      buildpack: https://github.com/cloudfoundry/nodejs-buildpack/releases/download/v1.6.39/nodejs-buildpack-cflinuxfs3-v1.6.39.zip
    requires:
      - name: lbn-gtt-sample-app-uaa
      - name: lbn-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
```

```
53  resources:
54    - name: lbn-gtt-sample-app-runtime
55      parameters:
56        service-plan: app-runtime
57        service: html5-apps-repo
58      type: org.cloudfoundry.managed-service
59    - name: lbn-gtt-sample-portal-resources
60      parameters:
61        service-plan: standard
62        service: portal
63      type: org.cloudfoundry.managed-service
64    - name: lbn-gtt-sof-api
65      properties:
66        url: https://gtt-sample-track-salesorders-service-{unique-identifier}.${default-domain}
67    - name: lbn-gtt-sst-api
68      properties:
69        url: https://gtt-sample-track-shipments-service-{unique-identifier}.${default-domain}
70    - name: lbn-gtt-pof-api
71      properties:
72        url: https://gtt-sample-track-purchaseorders-service-{unique-identifier}.${default-domain}
73    - name: lbn-gtt-sample-app-uaa
74      type: com.sap.xs.uaa-application
75      parameters:
76        config:
77          xsappname: "${xsappname}"
78          tenant-mode: dedicated
79          oauth2-configuration:
80            system-attributes:
81              - rolecollections
82          description: Security profile of sales order fulfillment application
83          scopes:
84            - name: "$XSAPPNAME.sof.r"
85              description: Allow to access sales order fulfillment application
86            - name: "$XSAPPNAME.pof.r"
87              description: Allow to access purchase order fulfillment application
88            - name: "$XSAPPNAME.sst.r"
89              description: Allow to access standalone shipment tracking application
90            - name: "$XSAPPNAME.sst.w"
91              description: Allow to report events in standalone shipment tracking application
92            - name: uaa.user
93              description: UAA
94          authorities:
95            - "$XSAPPNAME.sof.r"
96            - "$XSAPPNAME.pof.r"
97            - "$XSAPPNAME.sst.r"
98            - "$XSAPPNAME.sst.w"
99            - uaa.user
100         role-templates:
101           - name: SalesOrderFulfillmentAppUserDisplayTemplate
102             description: Track & Trace Sales Order Fulfillment Template App User Read Access
103             scope-references:
104               - "$XSAPPNAME.sof.r"
105               - uaa.user
```

```
 88              - name: "$XSAPPNAME.sst.r"
 89                description: Allow to access standalone shipment tracking application
 90              - name: "$XSAPPNAME.sst.w"
 91                description: Allow to report events in standalone shipment tracking application
 92              - name: uaa.user
 93                description: UAA
 94           authorities:
 95             - "$XSAPPNAME.sof.r"
 96             - "$XSAPPNAME.pof.r"
 97             - "$XSAPPNAME.sst.r"
 98             - "$XSAPPNAME.sst.w"
 99             - uaa.user
100           role-templates:
101             - name: SalesOrderFulfillmentAppUserDisplayTemplate
102               description: Track & Trace Sales Order Fulfillment Template App User Read Access
103               scope-references:
104                 - "$XSAPPNAME.sof.r"
105                 - uaa.user
106             - name: PurchaseOrderFulfillmentAppUserDisplayTemplate
107               description: Track & Trace Purchase Order Fulfillment Template App User Read Access
108               scope-references:
109                 - "$XSAPPNAME.pof.r"
110                 - uaa.user
111             - name: StandaloneShipmentTrackingAppUserDisplayTemplate
112               description: Track & Trace Standalone Shipment Tracking Template App User Read Access
113               scope-references:
114                 - "$XSAPPNAME.sst.r"
115                 - uaa.user
116             - name: StandaloneShipmentTrackingAppUserProcessTemplate
117               description: Track & Trace Standalone Shipment Tracking Template App User Write Access
118               scope-references:
119                 - "$XSAPPNAME.sst.w"
120                 - uaa.user
121    - name: gtt-ui-sample-track-salesorders-app-host
122      type: org.cloudfoundry.existing-service
123    - name: gtt-ui-sample-track-shipments-app-host
124      type: org.cloudfoundry.existing-service
125    - name: gtt-ui-sample-track-purchaseorders-app-host
126      type: org.cloudfoundry.existing-service
127    - name: lbn-gtt-sample-destination
128      type: org.cloudfoundry.managed-service
129      parameters:
130        service-plan: lite
131        service: destination
```

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

```json
{
  "_version": "3.0.0",
  "identification": {
    "id": "ee7d30df-c86c-48b1-8f02-c2abc4e18f54-1578021444073",
    "entityType": "bundle"
  },
  "payload": {
    "catalogs": [
      {
        "_version": "3.0.0",
        "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"
        },
        "payload": {
          "viz": [
            {
              "id": "com.sap.gtt.app.sample.sof",
              "appId": "com.sap.gtt.app.sample.sof",
              "vizId": "SalesOrder-track"
            }
          ]
        }
      }
```

```
{
  "_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"
      }
    ]
  }
},
{
  "_version": "3.0.0",
  "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": [
  {
    "_version": "3.0.0",
    "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": {
```

```
 90        ],
 91        "sites": [
 92          {
 93            "_version": "3.0.0",
 94            "identification": {
 95              "id": "a3eee102-5181-4222-9924-8b8e90985f07-1578021444073",
 96              "entityType": "site",
 97              "title": "SAP Fiori launchpad site on Cloud Foundry",
 98              "description": "SAP Fiori launchpad site on Cloud Foundry, deployed from SAP Web IDE"
 99            },
100            "payload": {
101              "config": {
102                "ushellConfig": {
103                  "renderers": {
104                    "fiori2": {
105                      "componentData": {
106                        "config": {
107                          "applications": {
108                            "Shell-home": {}
109                          },
110                          "enableHelp": false,
111                          "enableSearch": false,
112                          "sessionTimeoutIntervalInMinutes": 19,
113                          "sessionTimeoutReminderInMinutes": 3
114                        }
115                      }
116                    }
117                  }
118                }
119              },
120              "groupsOrder": [
121                "salesOrder",
122                "shipment",
123                "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 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:
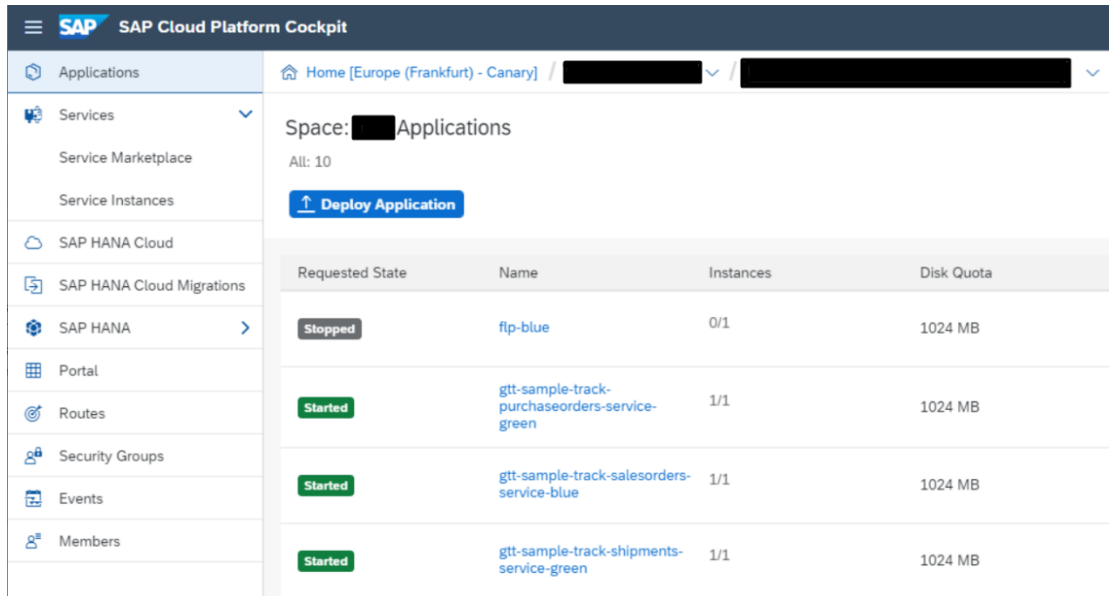
```
mbt build
```

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

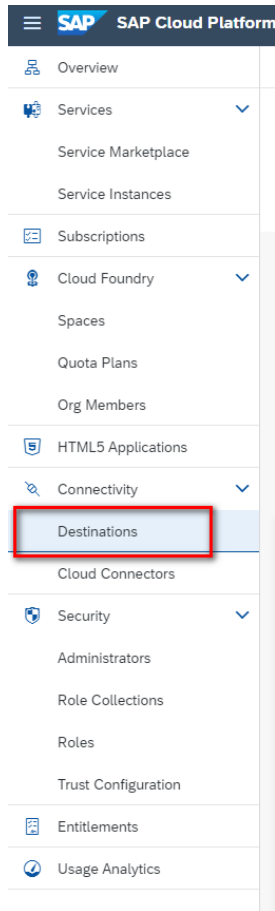After operations from Step 2.4.1 to Step 2.4.4, check the deployed apps on SAP Cloud Platform cockpit.

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:
Replace "TO_QW9_170_STANDARD" with the destination name that you maintained here and replace "QW9CLNT170" with the logical system of shipment tracked process.

```
Help
368  /*-----------------------------------Forward to TM system begin-------------------------------*/
369  var destinationTMName = "TO_QW9_170_STANDARD";
370  var msgObj = toObject("{}");
371  var eventObj = toObject("{}");
372
373  var estimatedArrivalTime = null;
374  var refPlannedEventMatchKey = null;
375  var estimatedArrival = null;
376
377  eventObj.eventReasonCode = "";
378
379  if (trackedProcess != null &&
380      "QW9CLNT170" == trackedProcess.logicalSystem &&
381      "SHIPMENT_ORDER" == actualEvent.trackingIdType &&
382      ("com.lbngttsamples.gtt.app.sof.Shipment.Departure" == actualEvent.eventType ||
383       "com.lbngttsamples.gtt.app.sof.Shipment.Arrival" == actualEvent.eventType ||
384       "com.lbngttsamples.gtt.app.sof.Shipment.Delay" == actualEvent.eventType ||
385       "com.lbngttsamples.gtt.app.sof.Shipment.LoadingStart" == actualEvent.eventType ||
386       "com.lbngttsamples.gtt.app.sof.Shipment.LoadingEnd" == actualEvent.eventType ||
387       "com.lbngttsamples.gtt.app.sof.Shipment.POD" == actualEvent.eventType ||
388       "com.lbngttsamples.gtt.app.sof.Shipment.POPU" == actualEvent.eventType ||
389       "com.lbngttsamples.gtt.app.sof.Shipment.Coupling" == actualEvent.eventType ||
390       "com.lbngttsamples.gtt.app.sof.Shipment.Decoupling" == actualEvent.eventType ||
391       "com.lbngttsamples.gtt.app.sof.Shipment.UnloadingStart" == actualEvent.eventType ||
392       "com.lbngttsamples.gtt.app.sof.Shipment.UnloadingEnd" == actualEvent.eventType)) {
393      if (actualEvent.containsProperty("estimatedArrival") && actualEvent.containsProperty("refPlannedEventMatchKey")) {
394          refPlannedEventMatchKey = actualEvent.refPlannedEventMatchKey;
395          estimatedArrival = actualEvent.estimatedArrival;
396          var i = 0;
397          // Get the lastest matched estimatedArrivalTime
398          while (i < estimatedArrival.size()) {
399              if (refPlannedEventMatchKey == estimatedArrival[i].stopId) {
400                  estimatedArrivalTime = estimatedArrival[i].estimatedArrivalTime;
401              }
402              i = i + 1;
403          }
```

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
StandaloneShipmentTrackingAppUserDisplayTemplate	tt_sof_{space}
StandaloneShipmentTrackingAppUserProcessTemplate	tt_sof_{space}

Note that the role "StandaloneShipmentTrackingAppUserDisplayTemplate" can only enable the user to view the *Track Shipments* template app.
The role "StandaloneShipmentTrackingAppUserProcessTemplate" can only enable the user to report events in the Track Shipments template app.
If the user needs access to both view the Track Shipments template app and report events in the app, then assign both roles to the role template.

If the user needs access to other template apps such as *Track SO Fulfillment* or *Track PO Fulfillment*, add the following roles:

Role	Application
SalesOrderFulfillmentAppUserDisplayTemplate	tt_sof_{space}
PurchaseOrderFulfillmentAppUserDisplayTemplate	tt_sof_{space}

2.4.8 Open *Track Shipments* Template App from SAP Fiori Launchpad
- Obtain the Application Routes from SAP Cloud Foundry cockpit
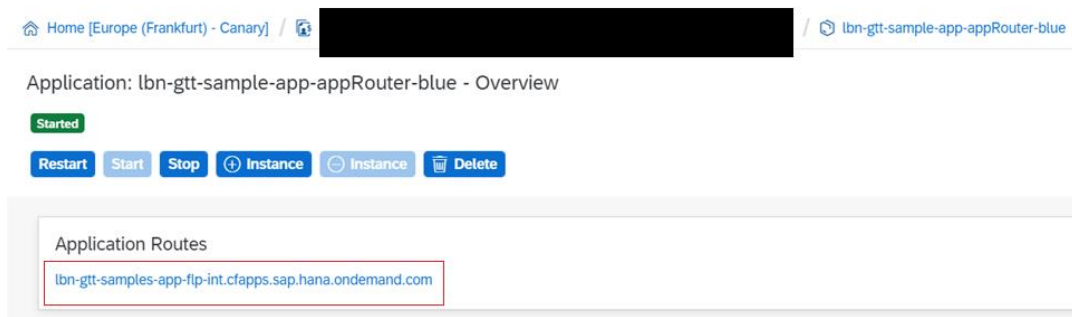
- Click the link of Application Routes
The system browser then navigates to the SAP Fiori Launchpad of *Track Shipments*

- Open the *Track Shipments* template app

3. Implement ERP Extractor

For ERP extractor implementation, 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 template apps in SAP BCP system under component "SCM-LBN-GTT-APP". Then follow the SAP Note 1773689: https://launchpad.support.sap.com/#/notes/0001773689 to add logon credentials to the secure area so that we can provide you with the key in a safe way. After you get the key, you can use it in Step 2.4.4 when deploying 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 the *Manage Locations* app.

1. Open the *Manage Locations* app in your SAP Fiori Launchpad for the global track and trace option
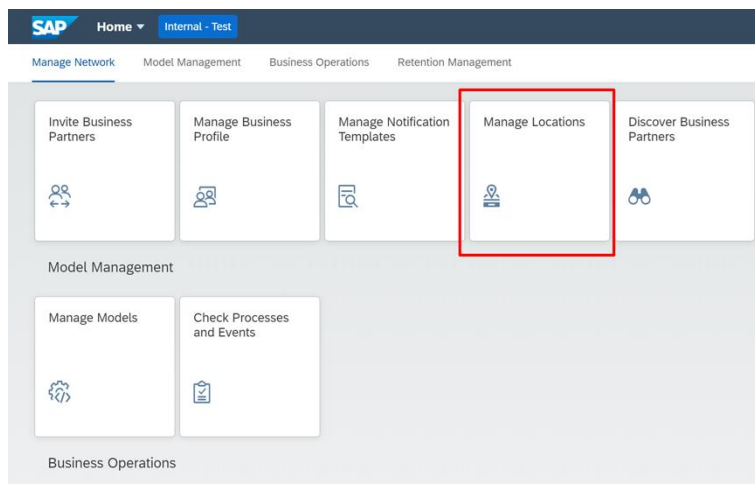
2. Click "Create" to create a new location
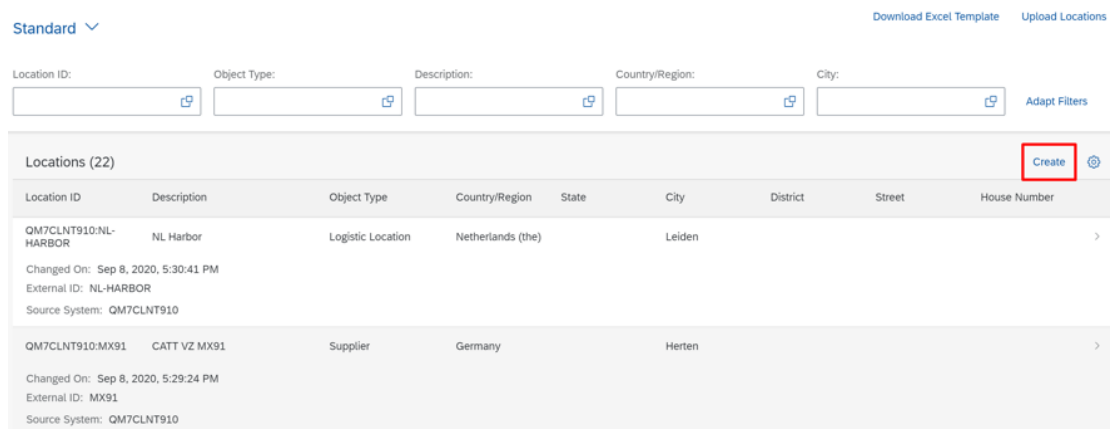
3. Maintain location information with valid geocoordinates

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

For more, 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)".