# Implementation Guide
# – *Track PO Fulfillment* 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 Business Technology Platform global account with entitlement to SAP Business Network Global Track and Trace, 1 portal service quota and 2 GB Application Runtime quota.
- To integrate SAP Business Network Global Track and Trace with ERP, your SAP ERP system should be running on Netweaver 7.31 or higher with SAP NOTE 2937175 being implemented. Besides, to implement the template apps' extractor codes, your SAP ERP system version should be S4 1909 SP03 on premise or higher.
- To integrate with visibility providers, log your incident in SAP BCP system with component 'SBN-LBN-GTT-APP'.
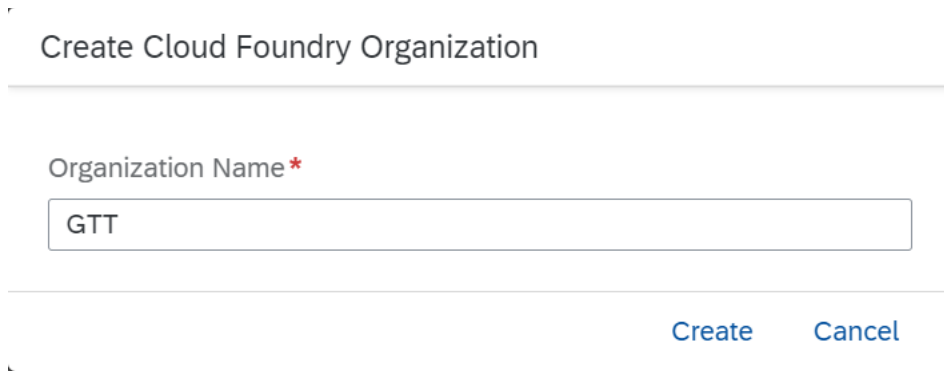
# 1.Environment Configuration in SAP Business Network Global Track and Trace

## 1.1 Enable Cloud Foundry and Create Space

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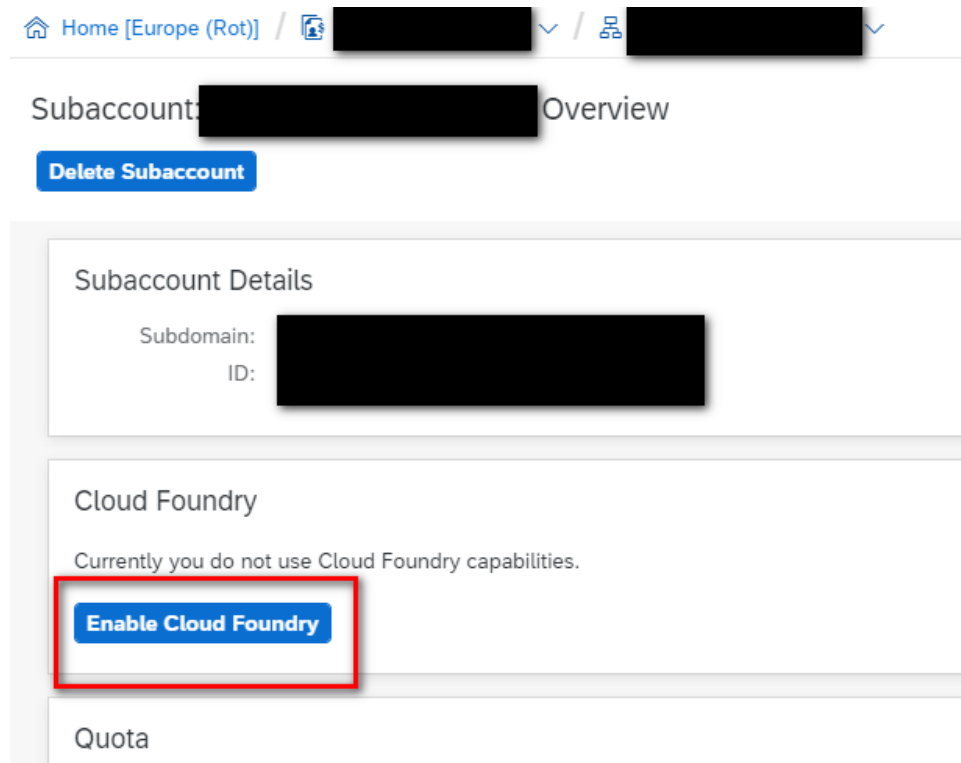
2. Fill in the Organization Name and click 'Create'

Create Cloud Foundry Organization

Organization Name*

GTT

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'



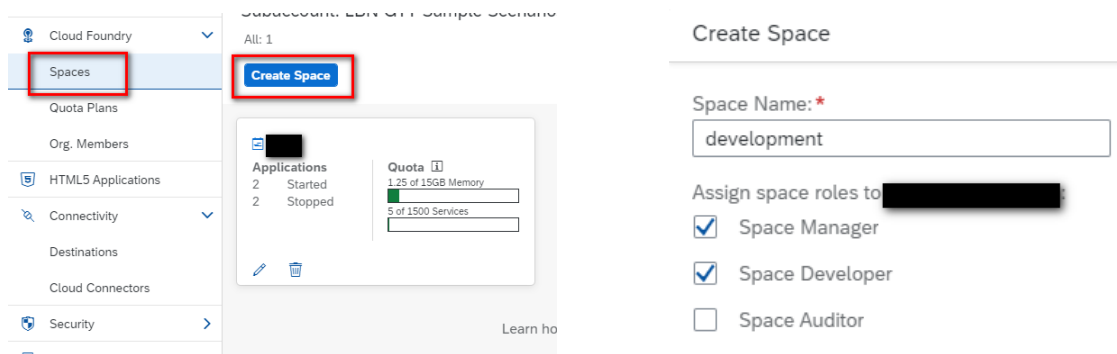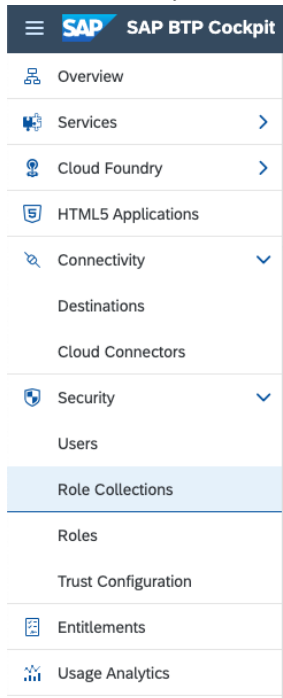2. Create 2 role collections (You can have user-defined names):
gtt-sample-all
gtt-sample-integration
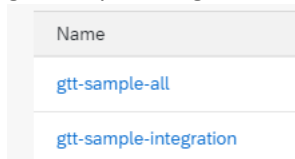


- Role template 'gtt-sample-all' is granted to users who have access to SAP Business Network Global Track and Trace and *Track PO Fulfillment* template app.
- Role template 'gtt-sample-integration' is granted to technical users only who have access to the global track and trace APIs for development purpose.

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 |
| 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 SAP Business Network Global Track and Trace and *Track PO Fulfillment* 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.

## 1.3 Maintain Technical User

1. Go to the SAP Business Technology 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

**SAP** CLOUD IDENTITY

Hello

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.

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 |
| 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 SAP Business Network Global Track and Trace.
   Open *Manage Business Profile* app.
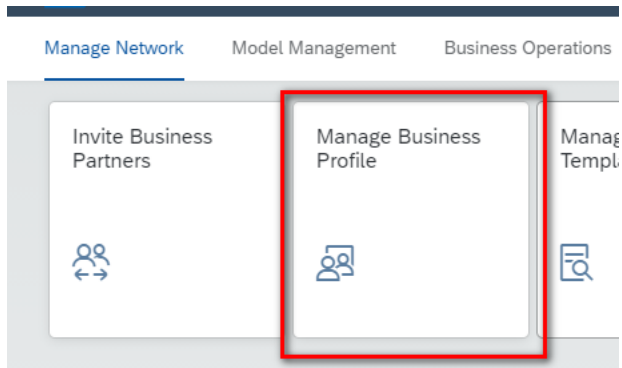


7. Add one Alternate Identifier:
   Identifier Type: Technical User
   ID: the email address of technical user



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

1. Go to Destination
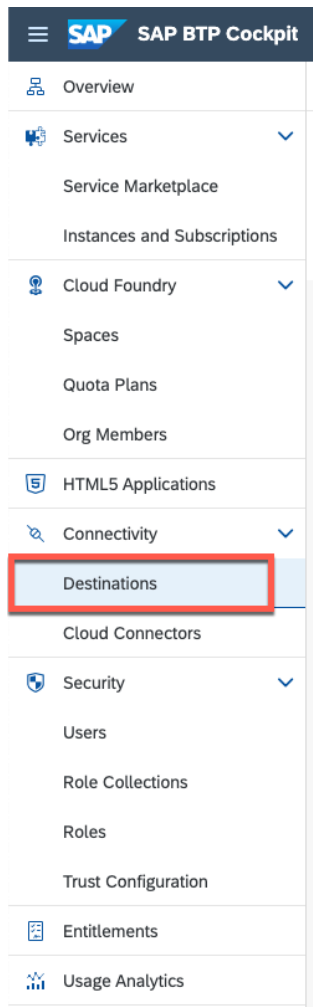


2. Configure Destination of GTT Core Engine Base URL

| 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 |

# 2.Implement *Track PO Fulfillment* Template App

## 2.1 Download Template Code

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

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

*C:\>mkdir dev*
*C:\>cd dev*
*C:\dev>git clone https://github.com/SAP-samples/logistics-business-network-gtt-samples.git sample*

*Note: make sure the target directory extract or clone is short path.*
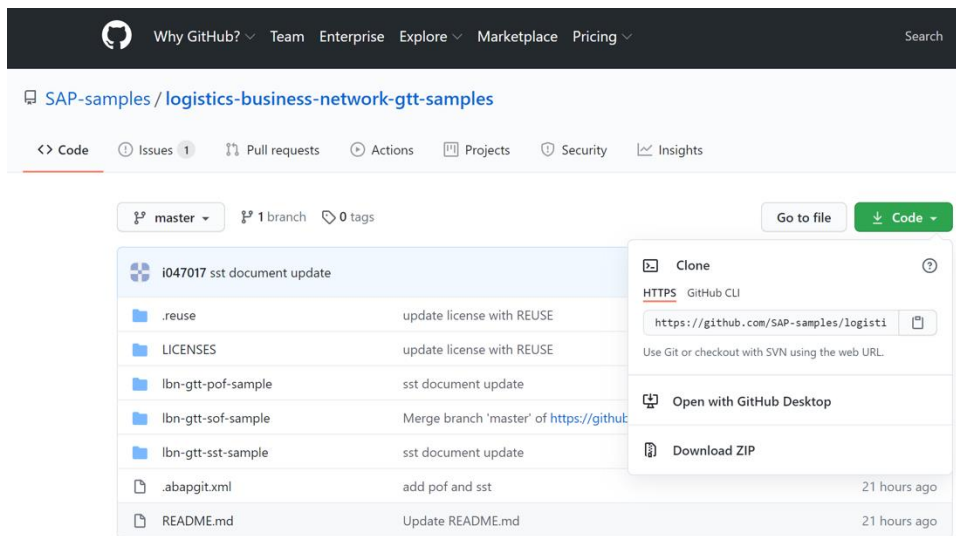


## 2.2 Download and Setup Tools

1. Download and Install git
   https://git-scm.com/download/win

2. Install JDK 8 and Maven

   JDK 8: https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

   Maven: https://maven.apache.org/download.cgi

3. Install NPM, Node JS, and grunt-cli

   Node JS **12.22.12**:

   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://registry.npmjs.org" -g

4. Install Cloud Foundry Command Line Interface
   https://developers.sap.com/tutorials/cp-cf-download-cli.html

5. Install MultiApps CF CLI Plugin
   https://github.com/cloudfoundry-incubator/multiapps-cli-plugin

6. Install Cloud MTA Build Tool (MBT)
   https://sap.github.io/cloud-mta-build-tool/download/

7. 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 *pof-model-YYYYMMDDxx.json* (e.g. pof-model-2020121901.json).
- Open *Manage Models* app on SAP Fiori Launchpad for SAP Business Network Global Track and Trace
- Click 'Import'
- Click 'Browse' to select the *pof-model-YYYYMMDDxx.json* file
- Click 'Import' to import the model
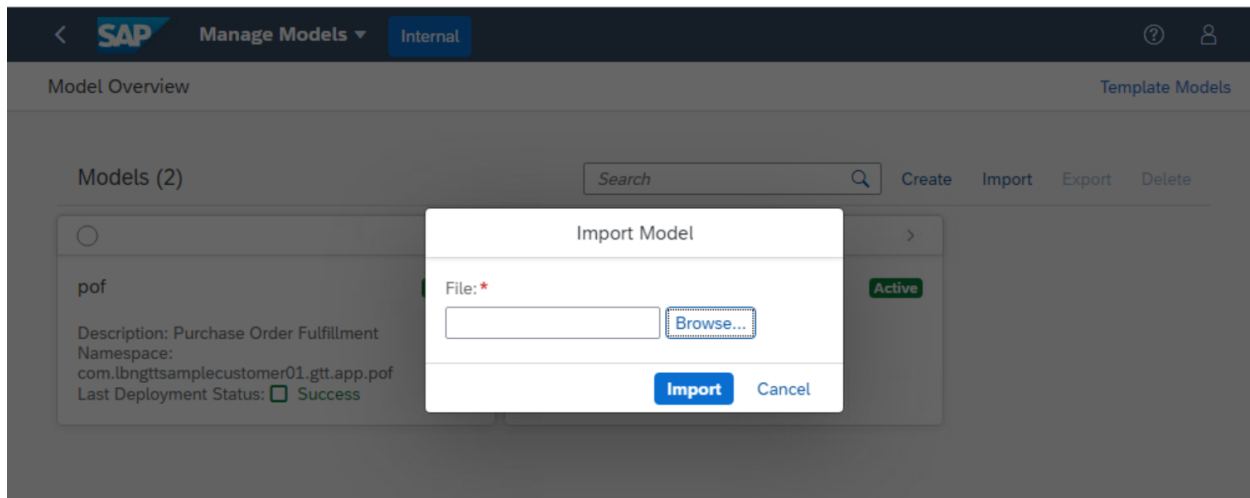


- Click 'Deploy'
- Record the model namespace for later reference, in this case, 'com.lbngttsamplecustomer01.gtt.app.pof' in the following screenshot:
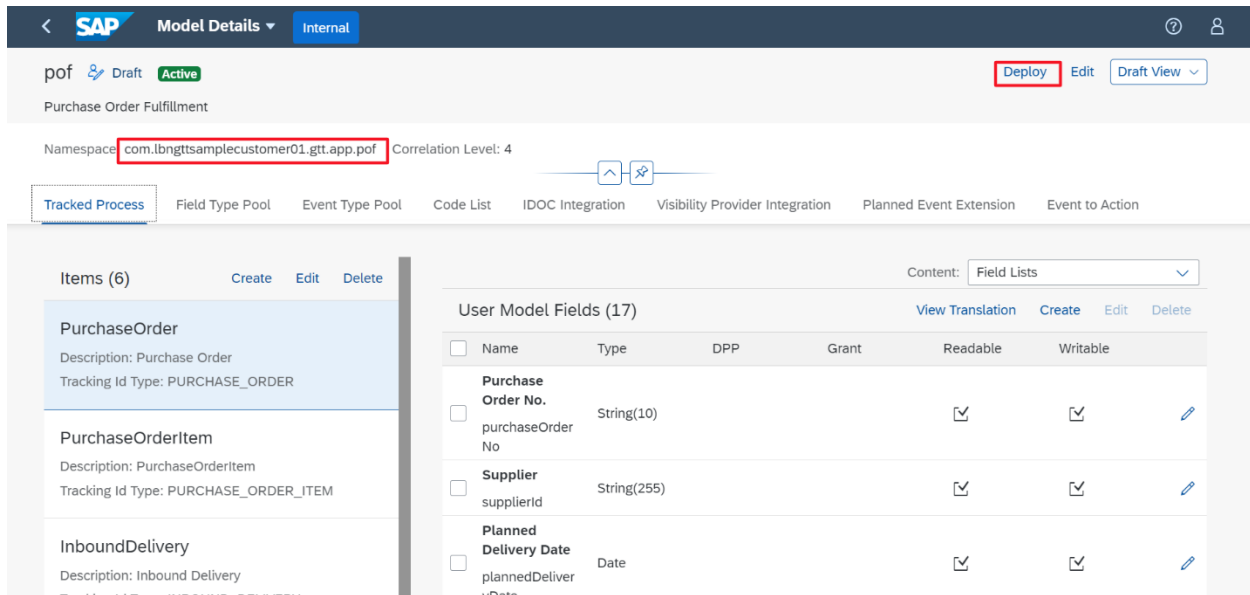


- Click 'Edit'
- Go to *IDOC Integration* tab

- Replace all the Application Object Type values with the ones you maintained in the ERP system



- Go to *Event to Action* tab
- Replace all the namespace strings, in this case, 'com.lbngttsamplecustomer01.gtt.app.pof' in *Event to Action* script with the model namespace you recorded in the previous step
- Click 'Save' to save the model and then click '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 where you want to deploy the template apps
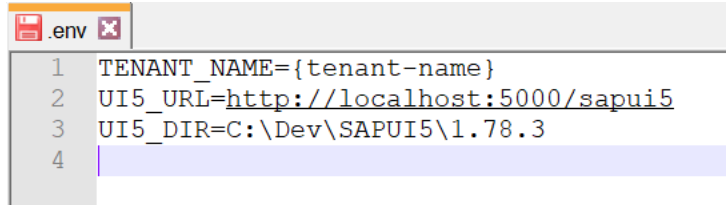
### 2.4.2 Package and Deploy *Track PO Fulfillment* Frontend App

Go to GTT-V2-Sample-TrackPurchaseOrders-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 'lbngttsamplecustomer01' in
  'com.lbngttsamplecustomer01.gtt.app.pof' 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
```

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-purchaseorders_xxx.mtar is then generated.

| | This PC > | GTT-V2-Sample-TrackPurchaseOrders-UI > mta_archives | | | |
|---|---|---|---|---|---|
| | Name | | Date modified | Type | Size |
| | gtt-v2-ui-sample-track-purchaseorders_2.1.0.mtar | | | MTAR File | 2,540 KB |
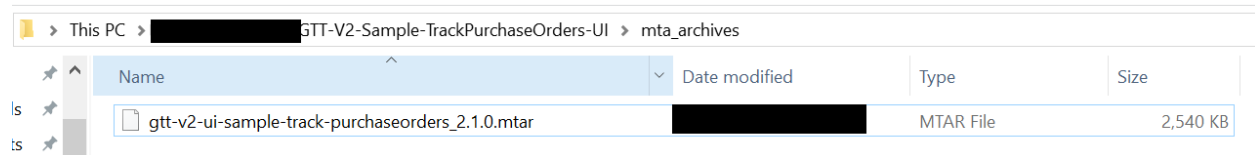
- To deploy the MTAR package in Cloud Foundry, run the following command:

  *cf deploy ./mta_archives/gtt-v2-ui-sample-track-purchaseorders_xxx.mtar*

The deployment time depends on your network performance.

### 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
  - The maximum length of unique information should be no more than 16 characters
  - Must start with alphabetic character
  - No special characters like -, /, &
  - Must in lower case

```
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 applications: Track SO Fulfillment, Track Shipments and Track PO Fulfillment. Before you build and deploy it, please deploy all the UI projects you need in advance. Otherwise, the deployment of this project will fail.

If you only need the Track PO Fulfillment app, make the following changes:
- Open the *mta.yaml* file
- Delete *lbn-gtt-sof-api* and *lbn-gtt-stt-api* in *lbn-gtt-sample-app-appRouter* and *resources* modules
- Delete *gtt-ui-sample-track-salesorders-app-host* and *gtt-ui-sample-track-shipments-app-host* in *flp* and *resources* modules

```
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-${space}.${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
33     requires:
34     - name: lbn-gtt-sample-app-uaa
35     - name: lbn-gtt-sample-app-runtime
36     - name: lbn-gtt-sample-portal-resources
37     - name: lbn-gtt-sof-api
38     - name: lbn-gtt-sst-api
39     - name: lbn-gtt-pof-api
40   - name: flp
41     type: com.sap.portal.content
42     path: flp
43     parameters:
44       stack: cflinuxfs3
45       memory: 128M
46       buildpack: https://github.com/cloudfoundry/nodejs-buildpack/releases/download/v1.6.39/nodejs-buildpack-cflinuxfs3-v1.6.39.zip
47     requires:
48     - name: lbn-gtt-sample-app-uaa
49     - name: lbn-gtt-sample-portal-resources
50     - name: gtt-ui-sample-track-salesorders-app-host
51     - name: gtt-ui-sample-track-shipments-app-host
52     - name: gtt-ui-sample-track-purchaseorders-app-host
```

```yaml
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-${space}.${default-domain}
67   - name: lbn-gtt-sst-api
68     properties:
69       url: https://gtt-sample-track-shipments-service-${space}.${default-domain}
70   - name: lbn-gtt-pof-api
71     properties:
72       url: https://gtt-sample-track-purchaseorders-service-${space}.${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            - "$XSAPPNAME.sof.r"
105            - uaa.user
106          - name: PurchaseOrderFulfillmentAppUserDisplayTemplate
107            description: Track & Trace Purchase Order Fulfillment Template App User Read Access
108            scope-references:
109            - "$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.sst* and *com.sap.gtt.app.sample.sof* in *payload/catalogs/payload/viz*
array
- Delete groups of sales Order and shipments in *payload/groups* array
- Delete groups of sales Order and shipments 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"
            }
          ]
        }
      }
    ],
```

```json
    "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"
            }
          ]
```

```
 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 PO Fulfillment* Backend Service
*Replace User / Password, Here Map License Key and Host Configuration*

- Obtain a Here Map license from [Step 4.1](#)
- Open the file *mta.yaml* under the folder 'GTT-V2-Sample-TrackPurchaseOrders-Service':

```
_schema-version: '3.1'
ID: gtt-v2-sample-track-purchaseorders-service
version: 2.3.0
modules:
- name: gtt-sample-track-purchaseorders-service
  type: java
  path: srv/
  properties:
    SPRING_PROFILES_ACTIVE: cloud
    DESTINATION_GTT: gtt_core_engine_flp_base_url
    HERE_MAP_KEY: {HERE_MAP_KEY}
  parameters:
    buildpack: java_buildpack
    health-check-timeout: 180
    host: gtt-sample-track-purchaseorders-service-{unique-identifier}
  build-parameters:
    ignore:
    - "*.md"
    build-result: target/*.jar
    timeout: 10m
  requires:
  - name: lbn-gtt-sample-app-uaa
  - name: lbn-gtt-sample-destination

resources:
- name: lbn-gtt-sample-app-uaa
  type: org.cloudfoundry.existing-service
- name: lbn-gtt-sample-destination
  type: org.cloudfoundry.existing-service
```

- Replace {here-map-key} with your license key
- 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/pof/constant/Constants.java
  - Replace 'com.lbngttsamples.gtt.app.pof' with the model namespace you got from [Step 2.3](#)

```
package com.sap.gtt.v2.sample.pof.constant;

public class Constants {
    public static final String GTT_MODEL_NAMESPACE = "com.lbngttsamples.gtt.app.pof";
    public static final String GTT_MODEL_NAMESPACE_WRITE_SERVICE = GTT_MODEL_NAMESPACE + ".pofWriteService";
    public static final String MODEL_NAMESPACE = GTT_MODEL_NAMESPACE + ".pofService";
```

- Open file ./srv/pom.xml

  Change line 24, 25
  from
      &lt;spring-boot-starter.version&gt;2.4.0&lt;/spring-boot-starter.version&gt;
      &lt;spring-security.version&gt;5.5.0-M1&lt;/spring-security.version&gt;
  to

```
<spring-boot-starter.version>2.4.3</spring-boot-starter.version>
<spring-security.version>5.5.0</spring-security.version>
```

- Run the following command to build the MTAR package:

  *mbt build*

A MTAR file *./mta_archives/gtt-v2-sample-track-purchaseorders-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-purchaseorders-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 Business Technology Platform Cockpit.



## 2.4.6 Maintain Destination of Forwarding URL

1. Obtain the UAA Token
- Go to the *Track PO Fulfillment* backend application that you have deployed

| Started | gtt-sample-track-purchaseorders-service | 1/1 | 1024 MB | 1024 MB | ▷ ⊙ 🗑 |

- Find the environment of UAA

- Copy the value of **clientid** and **clientsecret**



2. Go to 'Destinations'

3. Configure Destination of Event to Action Forwarding

| Name | e.g. gtt_pof_forwarding_int |
|---|---|
| Type | HTTP |
| Description | <optional> |
| URL | e.g. https://gtt-sample-track-purchaseorders-service.cfapps.eu10.hana.ondemand.com/sap/logistics/gtt/sample/pof/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 *Track PO Fulfillment* model if you change the name from 'gtt_pof_forwarding_int' to others.
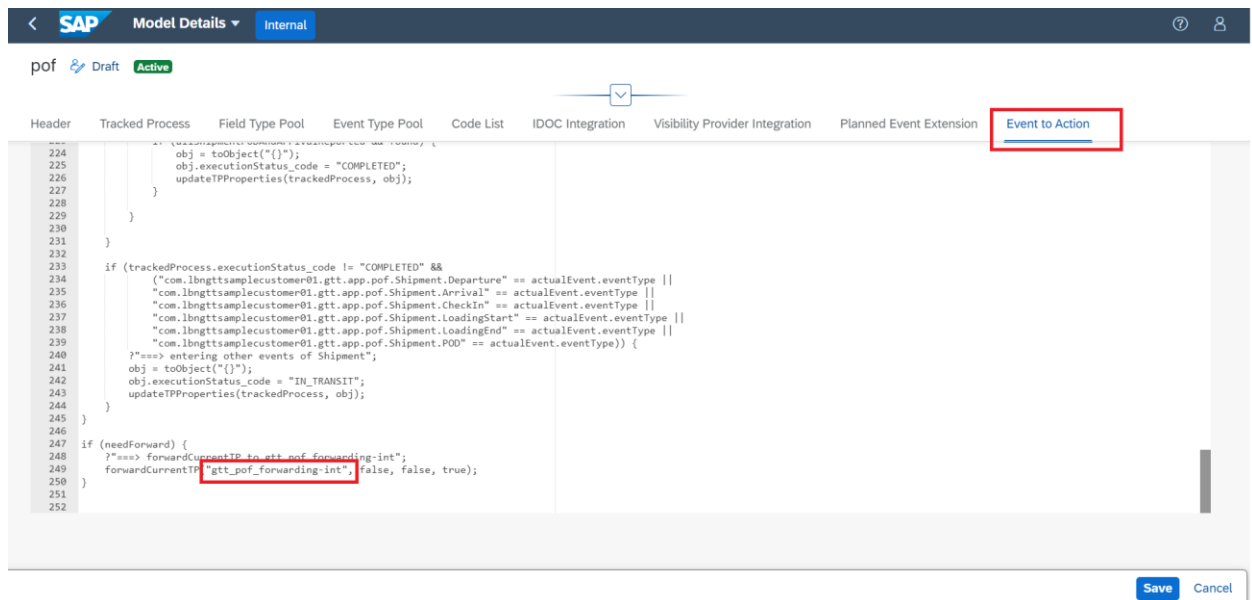


### 2.4.7 Maintain Roles and Users

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

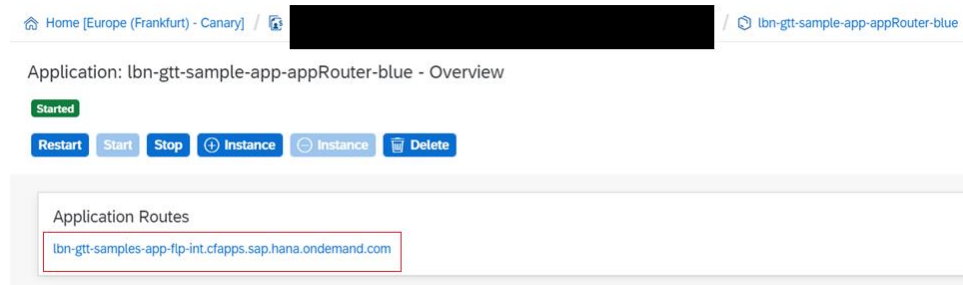| Role | Application |
|---|---|
| PurchaseOrderFulfillmentAppUserDisplayTemplate | tt_pof_{space} |

Note that the role of 'PurchaseOrderFulfillmentAppUser' is used to access the *Track PO Fulfillment* app.

If you need to access other template applications such as *Track Shipments* or *Track SO Fulfillment*, add the following roles:
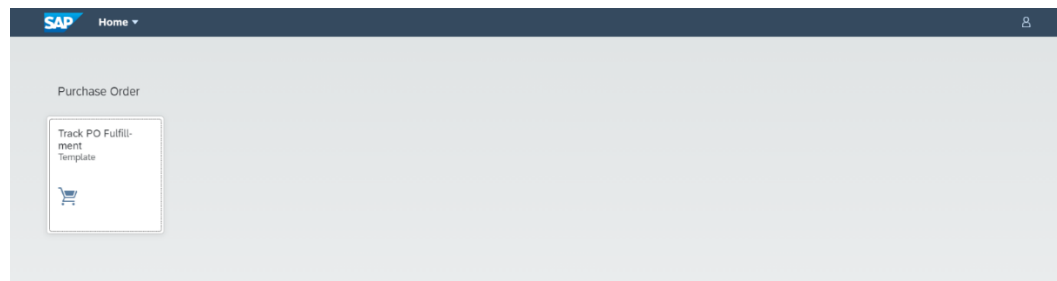
| Role | Application |
|------|-------------|
| StandaloneShipmentTrackingAppUserDisplayTemplate | tt_sst_{space} |
| StandaloneShipmentTrackingAppUserProcessTemplate | tt_sof_{space} |
| SalesOrderFulfillmentAppUserDisplayTemplate | tt_sof_{space} |

### 2.4.8 Open *Track PO 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 *Track PO Fulfillment*

- Open the *Track PO Fulfillment* template app



# 3. Implement ERP Extractor

For ERP extractor implementation, refer to the following document:

02_Extractor_Creation_Guide-TPOF.pdf

# 4. Other Configuration

## 4.1 Obtain A 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 'S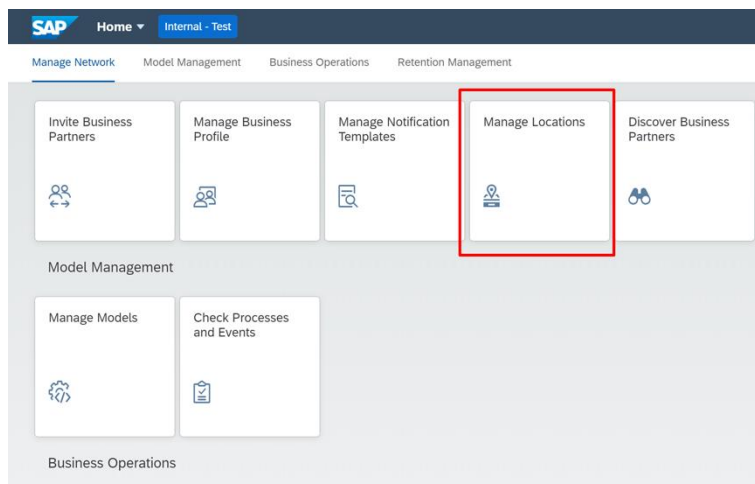BN-LBN-GTT-APP'. Then follow the SAP Note 1773689: https://launchpad.support.sap.com/#/notes/0001773689 to add logon credentials to the secure area so that we can provide you with the key in a safe way. After you get the key, you can use it in Step 2.4.4 when deploying the backend service.

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 SAP Business Network Global Track and Trace

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 details, refer to the following document:

02_Extractor_Creation_Guide-TPOF.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)'.