

# Implementation Guide

## – *Track PO Fulfillment* Template App

### Contents

0.Prerequisites .....	3
1.Environment Configuration in Logistics Business Network, Global Track and Trace Option.....	3
1.1 Enable Cloud Foundry and Create Space .....	3
1.2 Maintain Roles and Users.....	5
1.3 Maintain Technical User.....	6
1.4 Maintain Destination of GTT Core Engine Base URL.....	9
2.Implement <i>Track PO Fulfillment</i> Template App .....	10
2.1 Download Template Code .....	10
2.2 Download and Setup Tools.....	10
2.3 Deploy Model .....	11
2.4 Package and Deploy.....	13
2.4.1 Login to SAP Cloud Foundry.....	13
2.4.2 Package and Deploy <i>Track PO Fulfillment</i> Frontend App .....	14
2.4.3 Package and Deploy SAP Fiori Launchpad for <i>Template Apps</i> .....	14
2.4.4 Package and Deploy <i>Track PO Fulfillment</i> Backend Service.....	22
2.4.5 Check Deployed Apps on SAP Cloud Foundry Cockpit .....	23
2.4.6 Maintain Destination of Forwarding URL .....	23
2.4.7 Maintain Roles and Users .....	25
2.4.8 Open <i>Track PO Fulfillment</i> Template App from SAP Fiori Launchpad .....	26
3. Implement ERP Extractor .....	27
4. Other Configuration.....	28
4.1 Obtain A Here Map License .....	28
4.2 Maintain Location Master .....	28
4.3 Maintain Data Contributors .....	29

## Note

The SAP Cloud Platform brand is retired. For more information, see [A Robust Platform for RISE with SAP !\[\]\(529949c2c3dadbaa4e538e8c643454bc\_img.jpg\)](#). Parts of this content were created before our branding changes related to SAP technology were announced in January 2021.

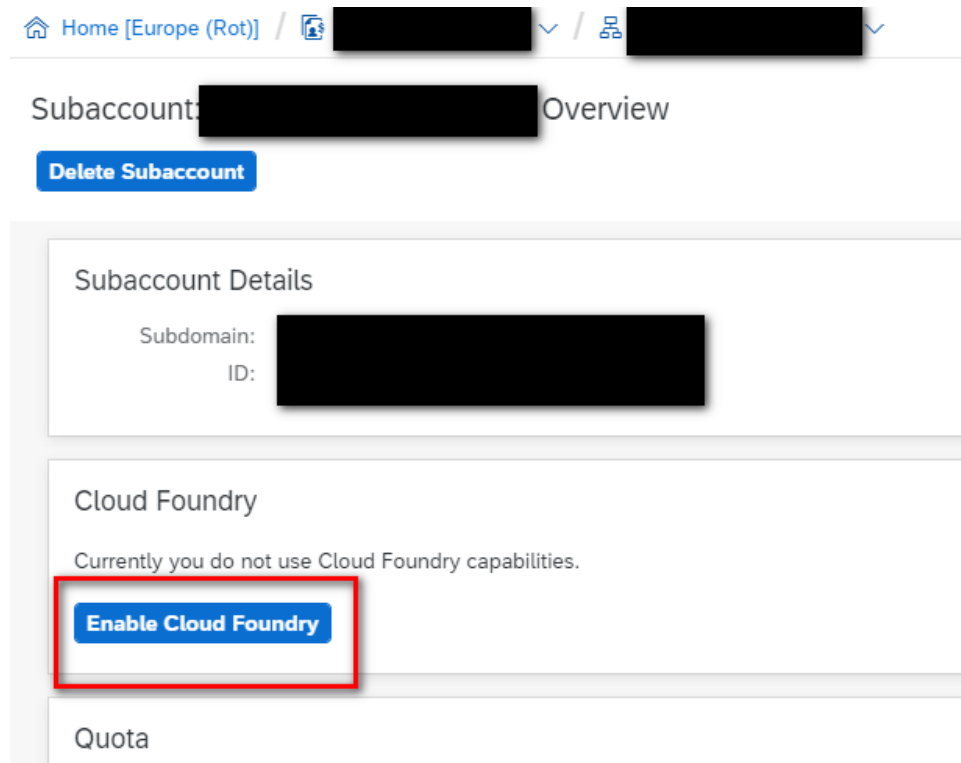
## 0.Prerequisites

- A SAP Cloud Platform global account with entitlement to the global track and trace option for SAP Logistics Business Network, 1 portal service quota and 2 GB Application Runtime quota.
- To integrate the global track and trace option 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 "SCM-LBN-GTT-COR".

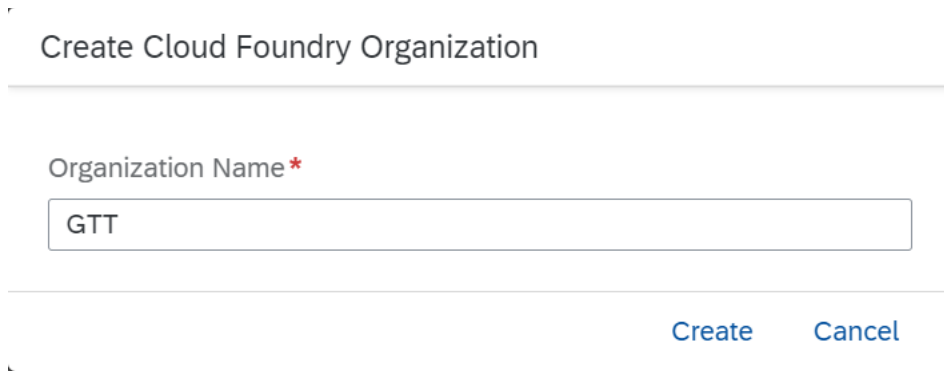
## 1.Environment Configuration in Logistics Business Network, Global Track and Trace Option

### 1.1 Enable Cloud Foundry and Create Space

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

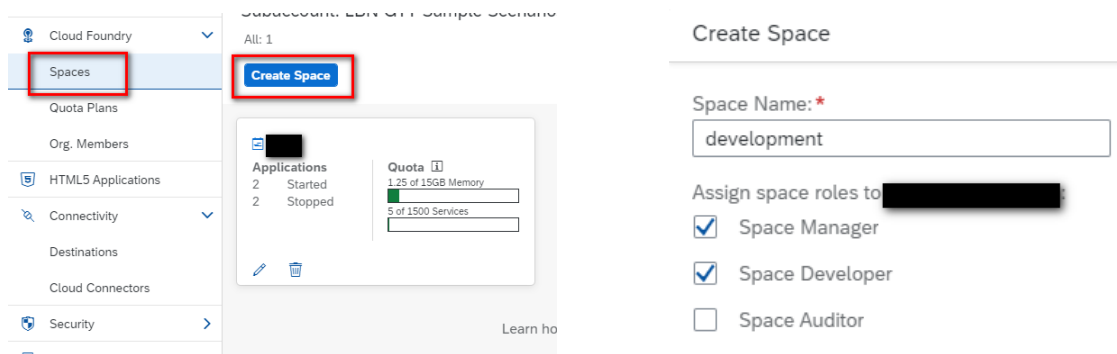


2. Fill in the Organization Name and click "Create"



The screenshot shows a web form titled "Create Cloud Foundry Organization". It has a single text input field labeled "Organization Name\*" with the value "GTT" entered. At the bottom right of the form are two buttons: "Create" and "Cancel".

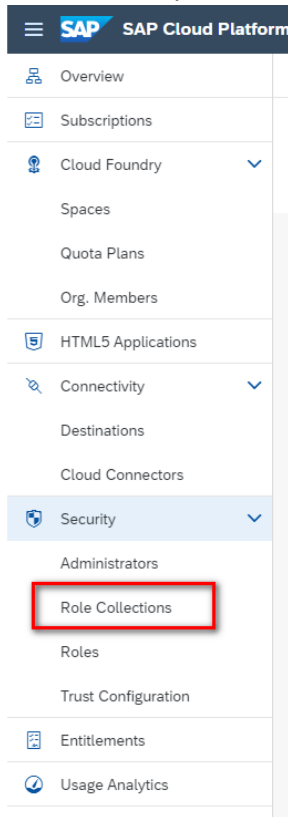
3. Go to "Cloud Foundry" -> "Spaces"  
Click "Create Space"  
Fill in the Space Name and click "Create"



The screenshot shows the Cloud Foundry console interface. On the left, a sidebar menu has "Spaces" highlighted with a red box. In the main area, a "Create Space" modal is open, also with a red box around its "Create Space" button. The modal shows a preview of the space with a name, applications, and quota. To the right, the "Create Space" form is shown with the "Space Name:" field containing "development". Below this, under "Assign space roles to", there are three checkboxes: "Space Manager" (checked), "Space Developer" (checked), and "Space Auditor" (unchecked). A redacted area is visible next to the "Assign space roles to" label.

## 1.2 Maintain Roles and Users

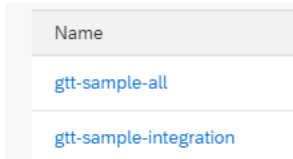
1. Go to “Security” -> “Role Collections”



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 *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

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

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

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

IT Administrator needs to maintain application users in identity provider.

For example, [user1@sap.com](mailto:user1@sap.com)



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

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

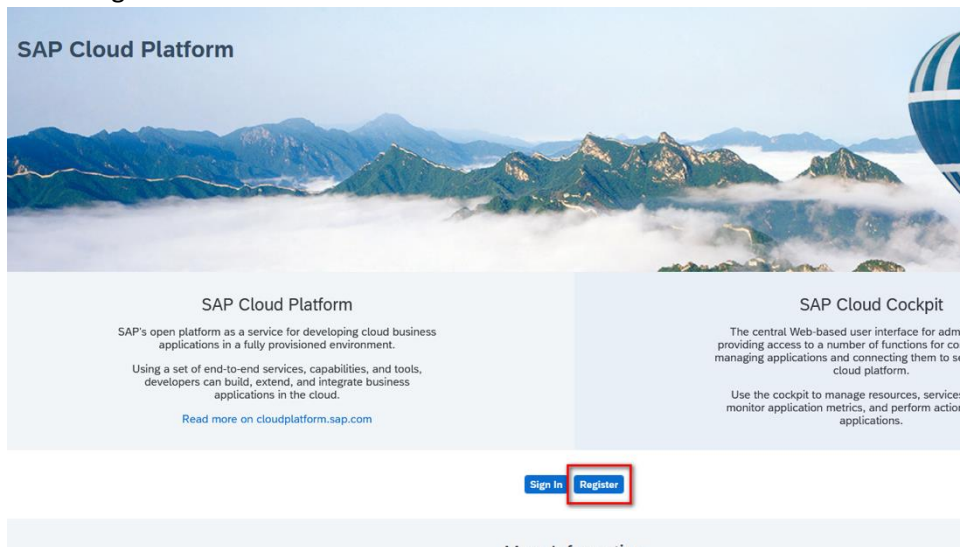
Note: The application user in this step should be different from the technical user created in [Step 1.3](#).

### 1.3 Maintain Technical User

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

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

Click "Register"



2. Fill in the mandatory fields:

Tell us about yourself

E-mail address \*

First name \*

Last name \*

Company \*

Country/Region  
United States

Phone  
+1 XXX-XXX-XXXX

Department \*

Relationship to SAP \*

Create password \*

Confirm password \*

Must have at least 8 characters.


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

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

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

Submit

3. Go to the mail inbox to activate the account

 CLOUD IDENTITY

Hello [redacted]

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

Click here to activate

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

[redacted]

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:

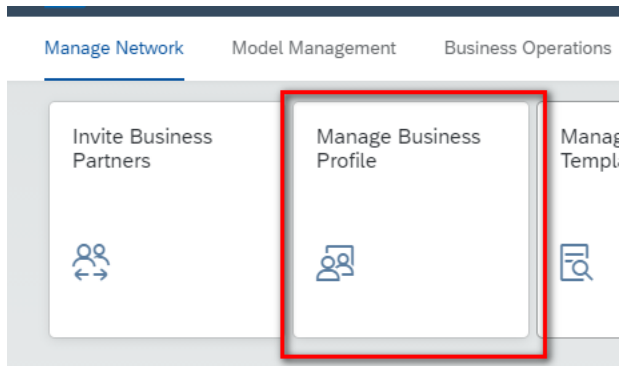
Role	Application
EMIDocIntegrationTemplate	lbn_gtt_core
MetadataServiceIntegrationTemplate	lbn_gtt_core
ReadServiceIntegrationTemplate	lbn_gtt_core
WriteServiceIntegrationTemplate	lbn_gtt_core
LocationProcessTemplate	lbnplatform-loc-broker

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

Users (1)

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

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



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

Email	Phone Number
[REDACTED]	[REDACTED]

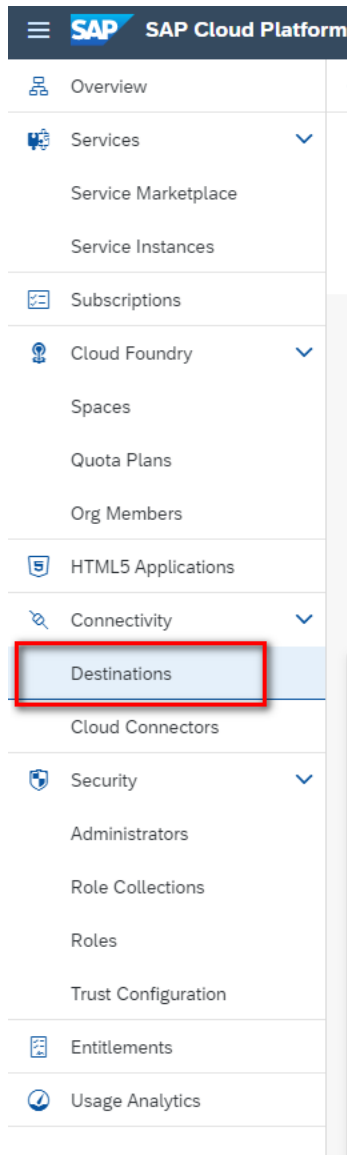
Alternate Identifiers (1)	
Identifier Type	ID
Technical User	[REDACTED].com

Note: The technical user in this step should be different from the application user in [Step 1.2.4](#).



## 1.4 Maintain Destination of GTT Core Engine Base URL

### 1. Go to Destination



### 2. Configure Destination of GTT Core Engine Base URL

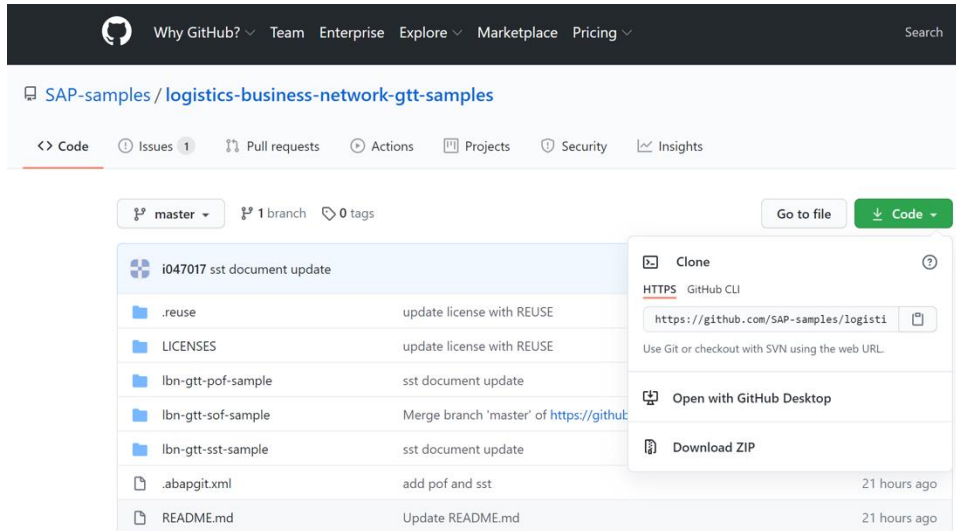
Name	gtt_core_engine_flp_base_url
Type	HTTP
Description	<optional>
URL	e.g. <a href="https://lbn-gtt-samples.gtt-flp-lbnplatform-int.cfapps.sap.hana.ondemand.com/api">https://lbn-gtt-samples.gtt-flp-lbnplatform-int.cfapps.sap.hana.ondemand.com/api</a>
Proxy Type	Internet
Authentication	BasicAuthentication
User	User from <a href="#">Step 1.3</a>
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 and extract, or git clone the template code from SAP-samples on GitHub.

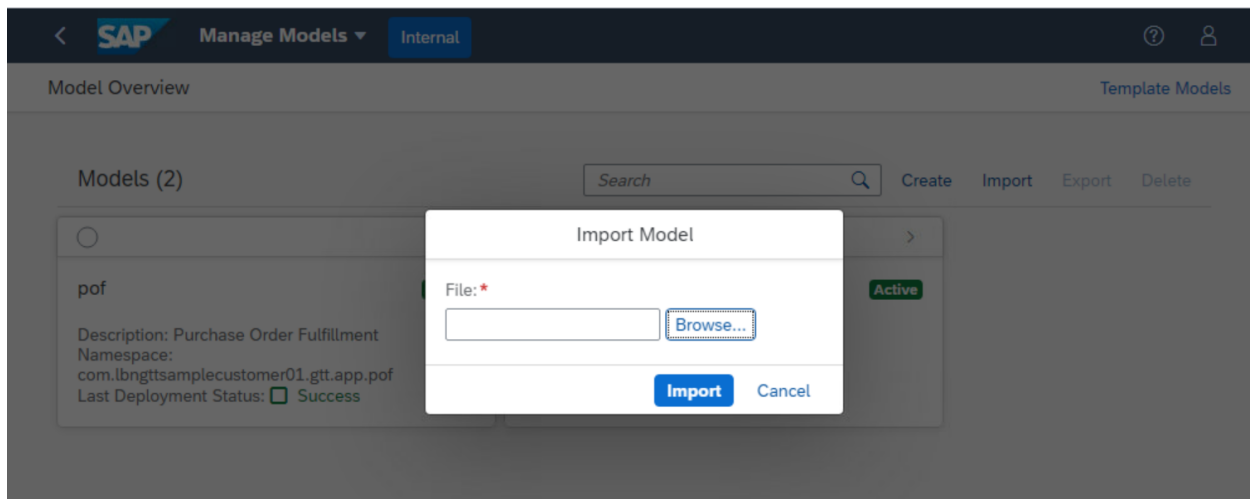


### 2.2 Download and Setup Tools

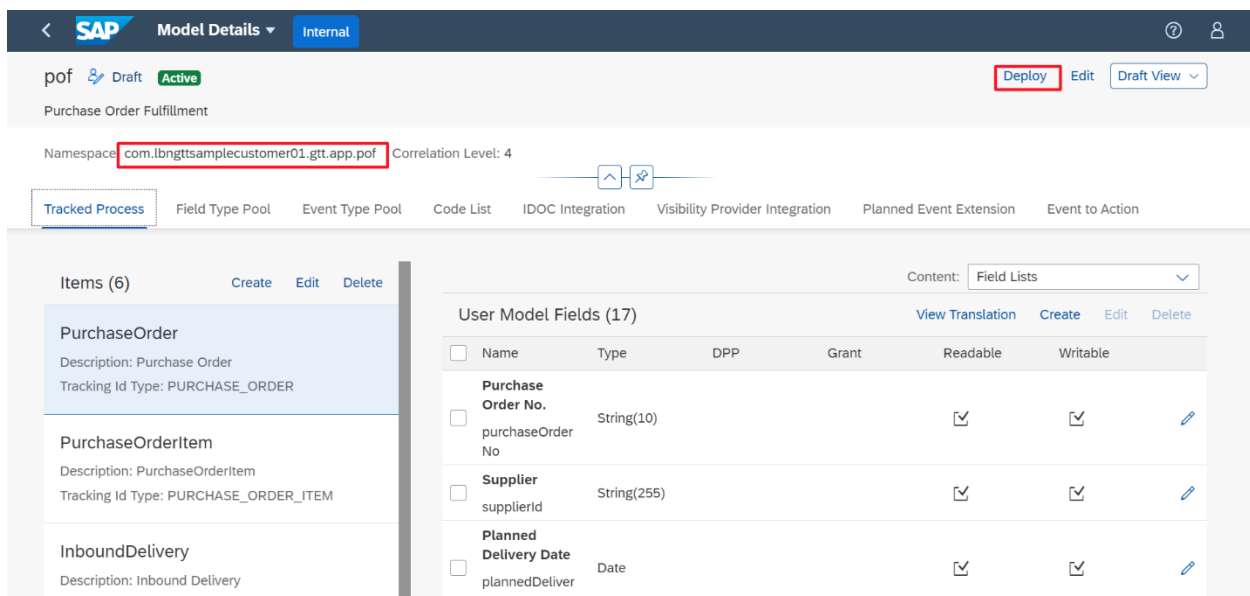
1. Install JDK 8 and Maven
2. Install NPM, Node JS, and grunt-cli  
<https://www.saplearners.com/how-to-install-node-js-and-npm/>  
Run the following command after NPM is installed:  
`npm config set @sap:registry "https://npm.sap.com" -g`
3. Install Cloud Foundry Command Line Interface  
<https://developers.sap.com/tutorials/cf-download-cli.html>
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 *pof-model-YYYYMMDDxx.json* (e.g. *pof-model-2020121901.json*).
- Open *Manage Models* app on SAP Fiori Launchpad for the global track and trace option
- 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.lbngttsamples.gtt.app.pof” in the following screenshot:



- Click “Edit”
- Go to *IDOC Integration* tab

- Replace all the Application Object Type value with the one you maintained in the ERP system

The screenshot shows the SAP Model Details interface for a model named 'pof' (Purchase Order Fulfillment). The 'Internal - Test' tab is active. The 'Tracked Process Mapping' section shows a dropdown menu for 'Tracked Process' with 'PurchaseOrder' selected. Below it, the 'ERP Object Type' dropdown is also set to 'PurchaseOrder'. The 'Tracked Process / Events (1)' table shows a single entry: 'PurchaseOrderEvent' with IDOC 'E1EHPAO'. The 'IDOC Integration' tab is selected, showing an 'Integration Switch' set to 'ON'. The 'Application Object Type' field is set to 'ZPOF\_GTT\_IN\_HD'. A table of fields maps various purchase order fields (like purchaseOrderNo, supplierId, plannedDeliveryDate) to IDOC segments (E1EHPCP) and IDOC fields (like YN\_PO\_NUMBER, YN\_PO\_SUPPLIER\_ID).

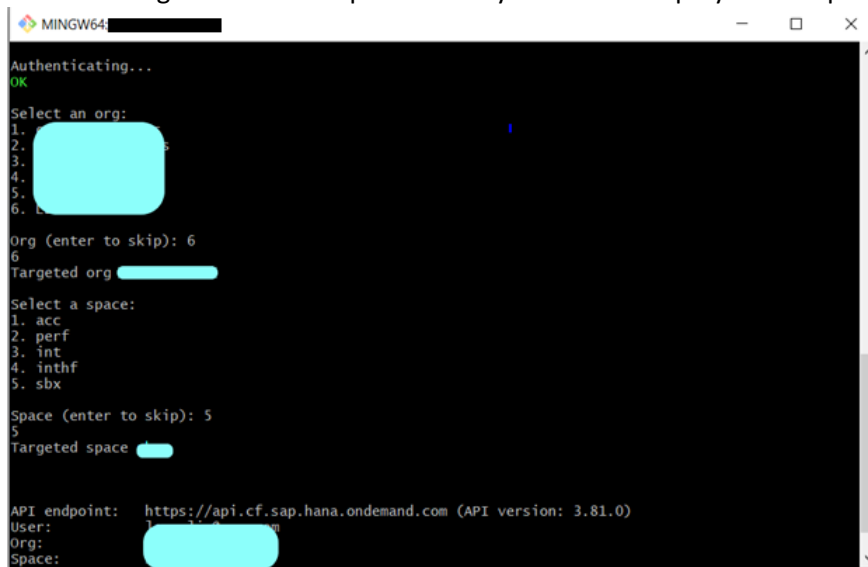
- Go to *Event to Action* tab
- Replace all the namespace strings, in this case, “com.lbngttsamples.gtt.app.pof” in *Event to Action* script with the model namespace you recorded in the previous step
- Click “Save and Deploy” to deploy the model

The screenshot shows the 'Event to Action' tab in the SAP Model Details interface. The namespace is set to 'com.lbngttsamplecustomer01.gtt.app.pof'. The event script is displayed, showing various SAP API calls and logic. The script includes comments for tracking process, item, and event. Key lines in the script are highlighted with red boxes: 'com.lbngttsamples.gtt.app.pof.Shipment.POD' and 'com.lbngttsamples.gtt.app.pof.PurchaseOrderItem.PurchaseOrderItem'. At the bottom right, there are 'Save' and 'Cancel' buttons.

## 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



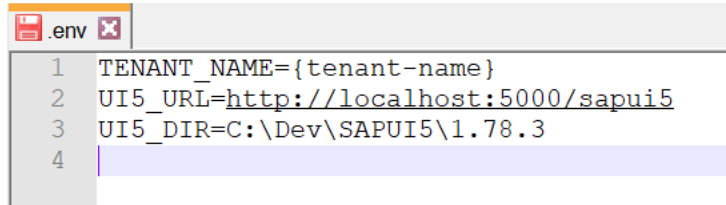
```
MINGW64 [redacted]
Authenticating...
OK
Select an org:
1. [redacted]
2. [redacted]
3. [redacted]
4. [redacted]
5. [redacted]
6. [redacted]
Org (enter to skip): 6
6
Targeted org [redacted]
Select a space:
1. acc
2. perf
3. int
4. inthf
5. sbx
Space (enter to skip): 5
5
Targeted space [redacted]

API endpoint: https://api.cf.sap.hana.ondemand.com (API version: 3.81.0)
User: [redacted]
Org: [redacted]
Space: [redacted]
```

## 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 “lbngttsamples” in “com.lbngttsamples.gtt.app.pof” shown in [Step 2.3](#).



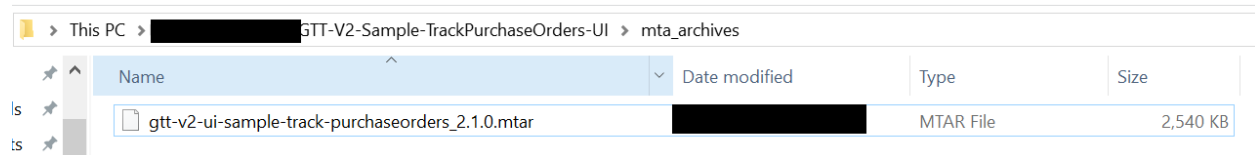
```
.env
1 TENANT_NAME={tenant-name}
2 UI5_URL=http://localhost:5000/sapui5
3 UI5_DIR=C:\Dev\SAPUI5\1.78.3
4
```

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.



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

```

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
25
26 resources:
27   - name: lbn-gtt-sample-app-runtime
28     parameters:
29       service-plan: app-runtime
30       service: html5-apps-repo
31     type: org.cloudfoundry.managed-service
32   - name: lbn-gtt-sample-portal-resources
33     parameters:
34       service-plan: standard
35       service: portal
36     type: org.cloudfoundry.managed-service
37   - name: lbn-gtt-sof-api
38     properties:
39       url: https://gtt-sample-track-salesorders-service-{unique-identifier} ${default-domain}
40   - name: lbn-gtt-stt-api
41     properties:
42       url: https://gtt-sample-track-shipments-service-{unique-identifier} ${default-domain}
43   - name: lbn-gtt-pof-api
44     properties:
45       url: https://gtt-sample-track-purchaseorders-service-{unique-identifier} ${default-domain}
46   - name: lbn-gtt-sample-app-uaa
47     type: com.sap.xs.uaa-application
48     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

```



```

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 *salesOrder* and *shipments* in *payload/groups* array
- Delete orders *salesOrder* and *shipments* in *payload/sites/payload/groupsOrder* array

```

1  {
2    "_version": "3.0.0",
3    "identification": {
4      "id": "ee7d30df-c86c-48b1-8f02-c2abc4e18f54-1578021444073",
5      "entityType": "bundle"
6    },
7    "payload": {
8      "catalogs": [
9        {
10         "_version": "3.0.0",
11         "identification": {
12           "id": "defaultCatalogId",
13           "title": "{{title}}",
14           "entityType": "catalog",
15           "i18n": "i18n/defaultCatalogId.properties"
16         },
17         "payload": {
18           "viz": [
19             {
20               "id": "com.sap.gtt.app.sample.sof",
21               "vizId": "SalesOrder-track"
22             },
23             {
24               "id": "com.sap.gtt.app.sample.sst",
25               "vizId": "Shipment-track"
26             },
27             {
28               "id": "com.sap.gtt.app.sample.pof",
29               "vizId": "PurchaseOrder-track"
30             }
31           ]
32         }
33       }
34     ],

```

```

35     "groups": [
36     {
37         "_version": "3.0.0",
38         "identification": {
39             "id": "salesOrder",
40             "title": "{{title}}",
41             "entityType": "group",
42             "i18n": "i18n/salesOrder.properties"
43         },
44         "payload": {
45             "viz": [
46                 {
47                     "id": "com.sap.gtt.app.sample.sof",
48                     "appId": "com.sap.gtt.app.sample.sof",
49                     "vizId": "SalesOrder-track"
50                 }
51             ]
52         }
53     },
54     {
55         "_version": "3.0.0",
56         "identification": {
57             "id": "shipment",
58             "title": "{{title}}",
59             "entityType": "group",
60             "i18n": "i18n/shipment.properties"
61         },
62         "payload": {
63             "viz": [
64                 {
65                     "id": "com.sap.gtt.app.sample.sst",
66                     "appId": "com.sap.gtt.app.sample.sst",
67                     "vizId": "Shipment-track"
68                 }
69             ]
70         }
71     },
72     {
73         "_version": "3.0.0",
74         "identification": {
75             "id": "purchaseOrder",
76             "title": "{{title}}",
77             "entityType": "group",
78             "i18n": "i18n/purchaseOrder.properties"
79         },
80         "payload": {
81             "viz": [
82                 {
83                     "id": "com.sap.gtt.app.sample.pof",
84                     "appId": "com.sap.gtt.app.sample.pof",
85                     "vizId": "PurchaseOrder-track"
86                 }
87             ]

```

```

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            "groupsOrder": [
120              "salesOrder",
121              "shipment",
122              "purchaseOrder"
123            ],
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: att_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";
}
```

- 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 Cloud Platform Cockpit.

The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar contains navigation links: Applications, Services, SAP HANA Cloud, SAP HANA Cloud Migrations, SAP HANA, Portal, Routes, Security Groups, Events, Members, Useful Links, and Legal Information. The main area displays the 'Applications' page for a specific space. It shows a table of deployed applications with columns: Requested State, Name, Instances, Disk Quota, Memory, and Actions. The table lists four applications: 'flp' (Stopped), 'gtt-sample-track-purchaseorders-service' (Started), 'gtt-ui-sample-track-purchaseorders-ui\_deployment' (Started), and 'lbn-gtt-sample-app-router' (Started). Each application row has a 'Deploy Application' button and a search bar.

Requested State	Name	Instances	Disk Quota	Memory	Actions
Stopped	flp	0/1	1024 MB	128 MB	[Play] [Stop] [Trash]
Started	gtt-sample-track-purchaseorders-service	1/1	1024 MB	1024 MB	[Play] [Stop] [Trash]
Started	gtt-ui-sample-track-purchaseorders-ui_deployment	1/1	1024 MB	256 MB	[Play] [Stop] [Trash]
Started	lbn-gtt-sample-app-router	1/1	256 MB	256 MB	[Play] [Stop] [Trash]

## 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

The screenshot shows a single application entry in the SAP Cloud Platform Cockpit. The application is 'gtt-sample-track-purchaseorders-service', which is in a 'Started' state. It has 1 instance, a disk quota of 1024 MB, and a memory usage of 1024 MB. The 'Actions' column shows a play button, a stop button, and a trash icon.

Requested State	Name	Instances	Disk Quota	Memory	Actions
Started	gtt-sample-track-purchaseorders-service	1/1	1024 MB	1024 MB	[Play] [Stop] [Trash]

- Find the environment of UAA

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

The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar contains navigation links: Overview, Service Bindings, Security, User-Provided Variables, Environment Variables, Events, and Logs. The main area displays 'Application: gtt-sample-track-purchaseorders-service - Service Bindings' with 'All: 2' items. A table lists the bindings:

Name	Service	Plan	Actions
lbr-gtt-sample-app-uaa	xsuaa	application	[Edit] [Delete]
lbr-gtt-sample-destination	destination	lite	[Edit] [Delete]

Below the table is a 'Hide sensitive data' button and a JSON configuration for the selected binding. The 'clientid' and 'clientsecret' fields are highlighted with red boxes:

```
{
  "tenantmode": "dedicated",
  "sburl": "https://api.authentication.eu10.hana.ondemand.com",
  "subaccountid": "XXXXXXXXXXXXXXXXXXXX",
  "clientid": "XXXXXXXXXXXXXXXXXXXX",
  "xsappname": "XXXXXXXXXXXXXXXXXXXX",
  "clientsecret": "XXXXXXXXXXXXXXXXXXXX",
  "url": "https://api.authentication.eu10.hana.ondemand.com",
  "uaadomain": "authentication.eu10.hana.ondemand.com",
  "apiurl": "https://api.authentication.eu10.hana.ondemand.com",
  "identityzone": "XXXXXXXXXXXXXXXXXXXX",
  "identityzoneid": "XXXXXXXXXXXXXXXXXXXX",
  "tenantid": "XXXXXXXXXXXXXXXXXXXX",
  "zoneid": "XXXXXXXXXXXXXXXXXXXX"
}
```

2. Go to “Destinations”

The screenshot shows the SAP Cloud Platform Cockpit navigation menu. The 'Destinations' option under the 'Connectivity' section is highlighted with a red box. The menu structure is as follows:

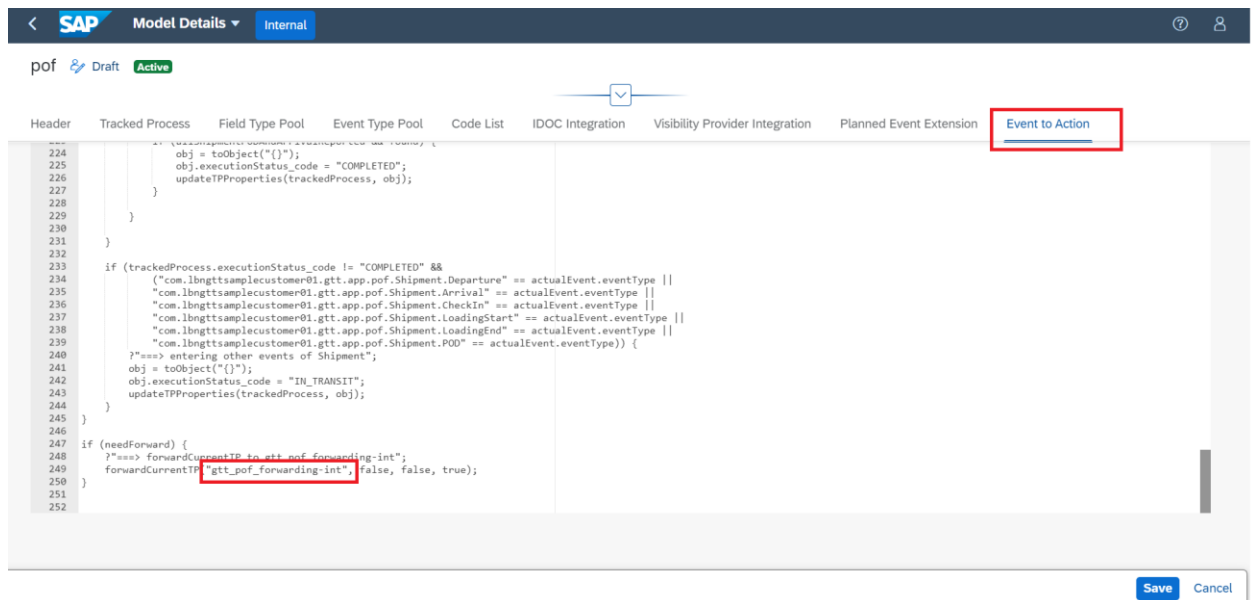
- Overview
- Services
  - Service Marketplace
  - Service Instances
- Subscriptions
- Cloud Foundry
  - Spaces
  - Quota Plans
  - Org Members
- HTML5 Applications
- Connectivity
  - Destinations**
  - Cloud Connectors
- Security
  - Administrators
  - Role Collections
  - Roles
  - Trust Configuration
- Entitlements
- Usage Analytics



### 3. Configure Destination of Event to Action Forwarding

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

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

- 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}

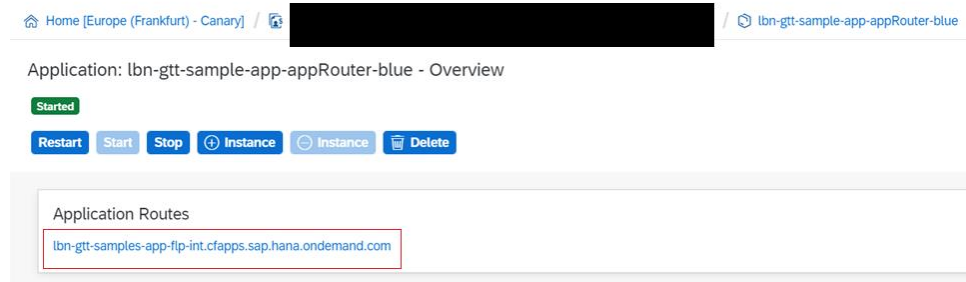
Note that the role of “PurchaseOrderFulfillmentAppUser” is used to access the *Track PO Fulfillment* app.

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

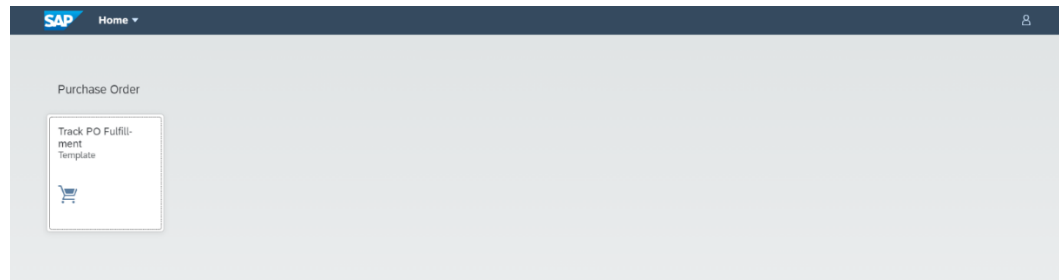
Role	Application
StandaloneShipmentTrackingAppUserDisplayTemplate	tt_sst_{space}
StandaloneShipmentTrackingAppUserProcessTemplate	tt_sof_{space}
SalesOrderFulfillmentAppUserDisplayTemplate	tt_sof_{space}

#### 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

Purchase Order No.:  Supplier:  Receiving Location:  Planned Delivery Date:  Material ID:  Material Description:  Shipment Number:

Inbound Delivery No.:

Adapt Filters **Go**

Purchase Orders (2) Purchase Order Items (2)

Purchase Order No.	Supplier	Receiving Location	Planned Delivery Date	Net Value	Completion Rate	Incoterms
9900029829	VEN_CO	0001	Oct 19, 2020	110000.00 INR	0.00 %	Costs and freight
9900066187	VEN_CO	0001	Oct 19, 2020	110000.00 INR	72.00 %	Costs and freight

### 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

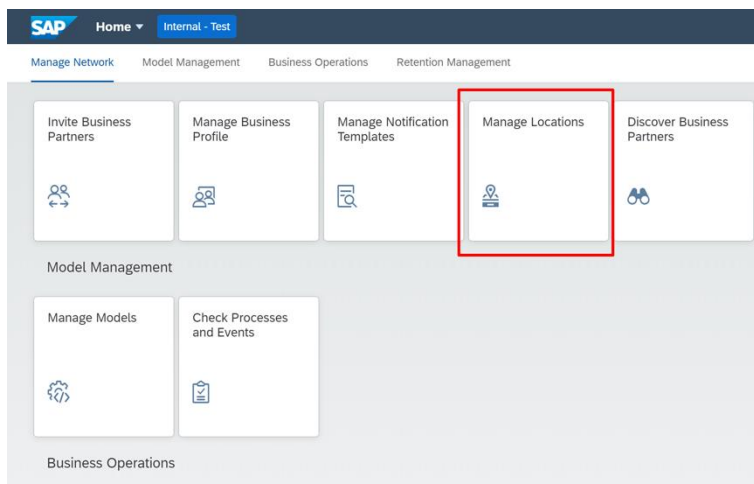
To have the map view, log the incident with the URL of Fiori Launchpad of template apps in SAP BCP system under component “SCM-LBN-GTT-APP”. Then follow the SAP Note 1773689: <https://launchpad.support.sap.com/#/notes/0001773689> to add logon credentials to the secure area so that we can provide you 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

The screenshot shows the SAP Locations table interface. At the top, there are filters for Location ID, Object Type, Description, Country/Region, and City. Below the filters, a table lists existing locations. The 'Create' button is highlighted with a red box in the top right corner of the table area.

Location ID	Description	Object Type	Country/Region	State	City	District	Street	House Number
QM7CLNT910-NL-HARBOR	NL Harbor	Logistic Location	Netherlands (the)		Leiden			
Changed On: Sep 8, 2020, 5:30:41 PM External ID: NL-HARBOR Source System: QM7CLNT910								
QM7CLNT910-MX91	CATT VZ MX91	Supplier	Germany		Herten			
Changed On: Sep 8, 2020, 5:29:24 PM External ID: MX91 Source System: QM7CLNT910								

## 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

The screenshot shows the 'Create Location' form in SAP. The 'GENERAL' tab is selected. The 'Object Type' is set to 'Logistic Location'. The 'Source System' and 'External ID' fields are visible. The 'Description' field is empty. The 'Time Zone' is set to 'UTC+01:00'. The 'Longitude' and 'Latitude' fields are highlighted with a red box, showing the format 'Example: 000.00000000'.

Object Type: \*  
Logistic Location

Source System/External ID: \*  
Source System: External ID

Description: \*

Time Zone: \*  
UTC+01:00

Longitude: \*  
+ (East) Example: 000.00000000

Latitude: \*  
+ (North) Example: 00.00000000

Country/Region: \*  
State: \*  
City: \*  
District: \*  
Street/House Number: \*  
Zip Code: \*

## 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](https://help.sap.com/gtt) and see the chapter “Onboarding and User Management (Data Contributors)”.