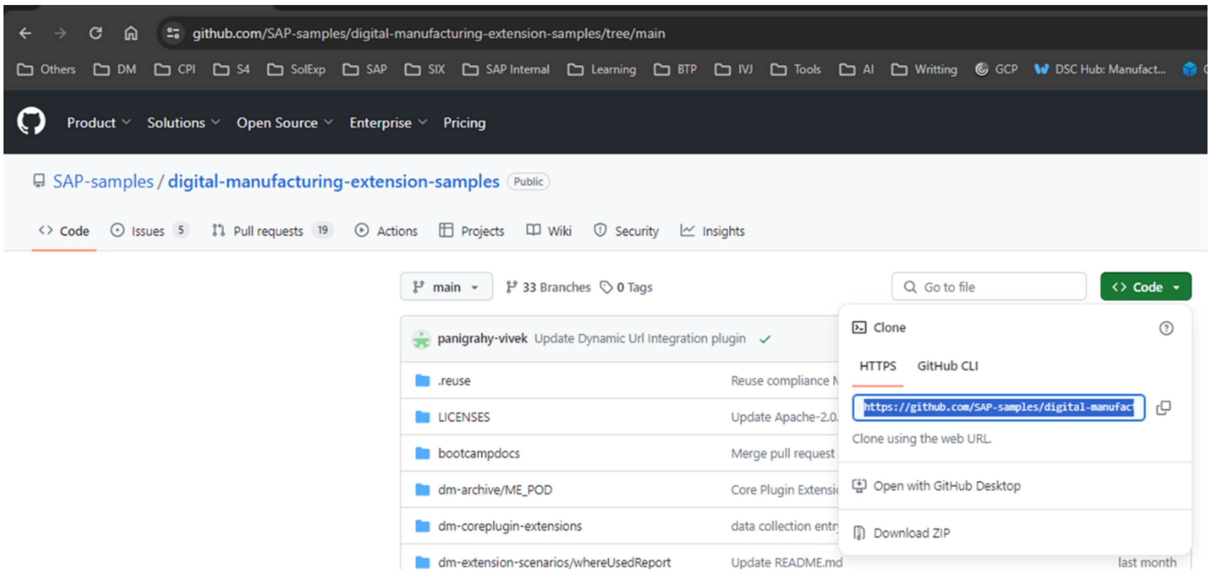
Standard POD plugin Extension

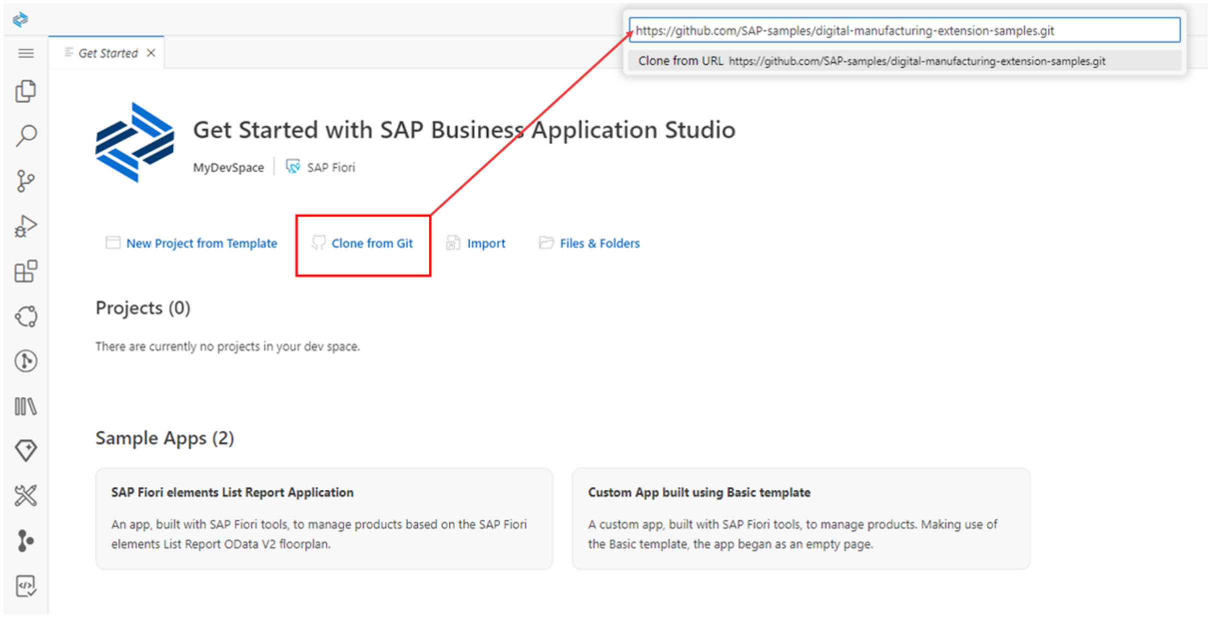
This document guide you through how to develop a simple pro code extension to the standard 'Assemble Components' POD Plugin, which should be enough to set a strong basis and enable you to extend other core POD Plugins further to address your own plugin extension use cases and other more complex requirements.

Open your Business Application Studio in BTP

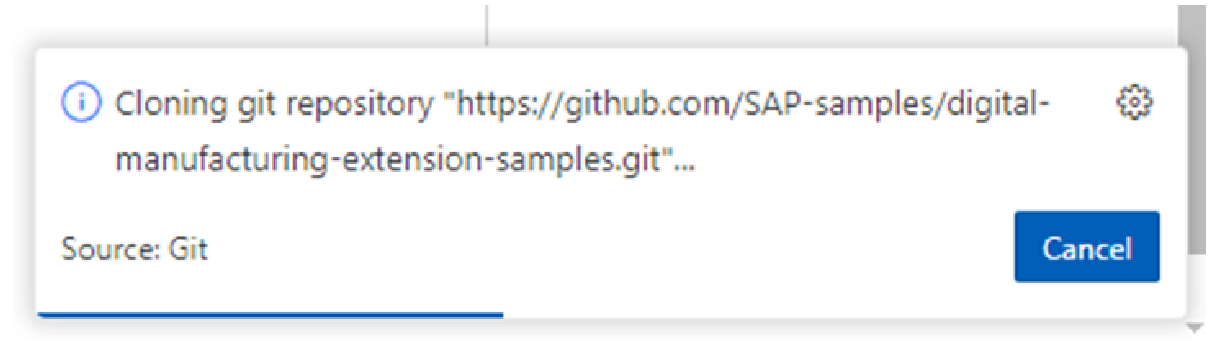
Open digital-manufacturing-extension-samples Github, click '<> Code' and copy the web URL from HTTPS tab.


github.com/SAP-samples/digital-manufacturing-extension-samples/tree/main
SAP-samples / digital-manufacturing-extension-samples
Code
Clone
HTTPS
GitHub CLI
https://github.com/SAP-samples/digital-manufac
Clone using the web URL.
Open with GitHub Desktop
Download ZIP

Back to SAP Business Application Studio, open a 'Dev Space' (make sure it is in RUNNING state), click 'Clone from Git', paste the web URL that was copied from Github in the previous step and hit 'Enter'.

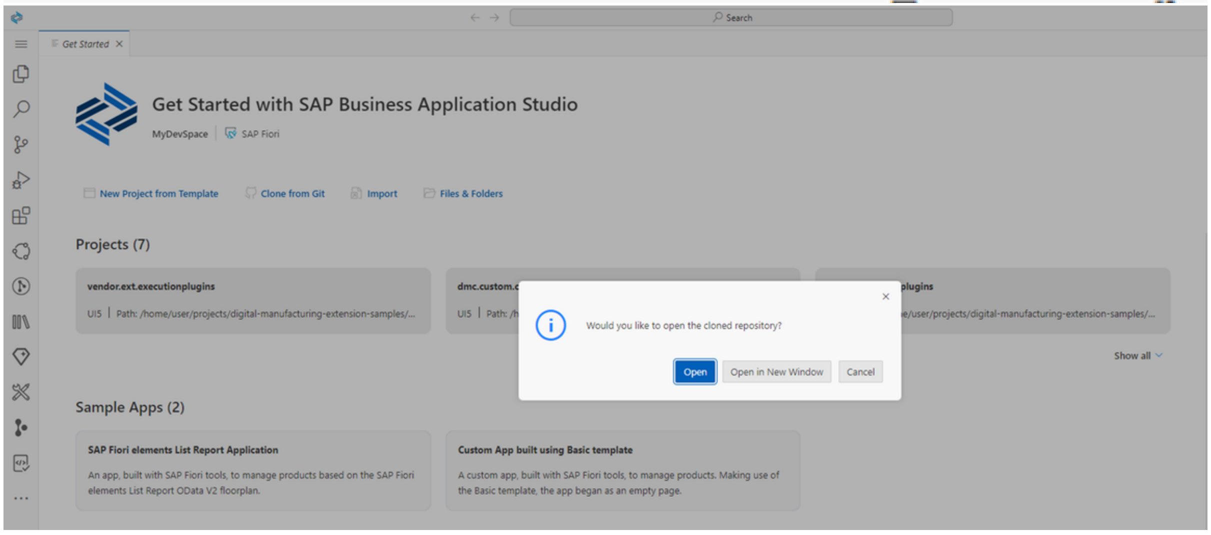

https://github.com/SAP-samples/digital-manufacturing-extension-samples.git
Clone from URL https://github.com/SAP-samples/digital-manufacturing-extension-samples.git
Get Started
Get Started with SAP Business Application Studio
MyDevSpace
SAP Fiori
New Project from Template
Clone from Git
Import
Files & Folders
Projects (0)
There are currently no projects in your dev space.
Sample Apps (2)
SAP Fiori elements List Report Application
Custom App built using Basic template

The git repository will be cloned, and once completed, can be opened.



Cloning git repository "https://github.com/SAP-samples/digital-manufacturing-extension-samples.git"...

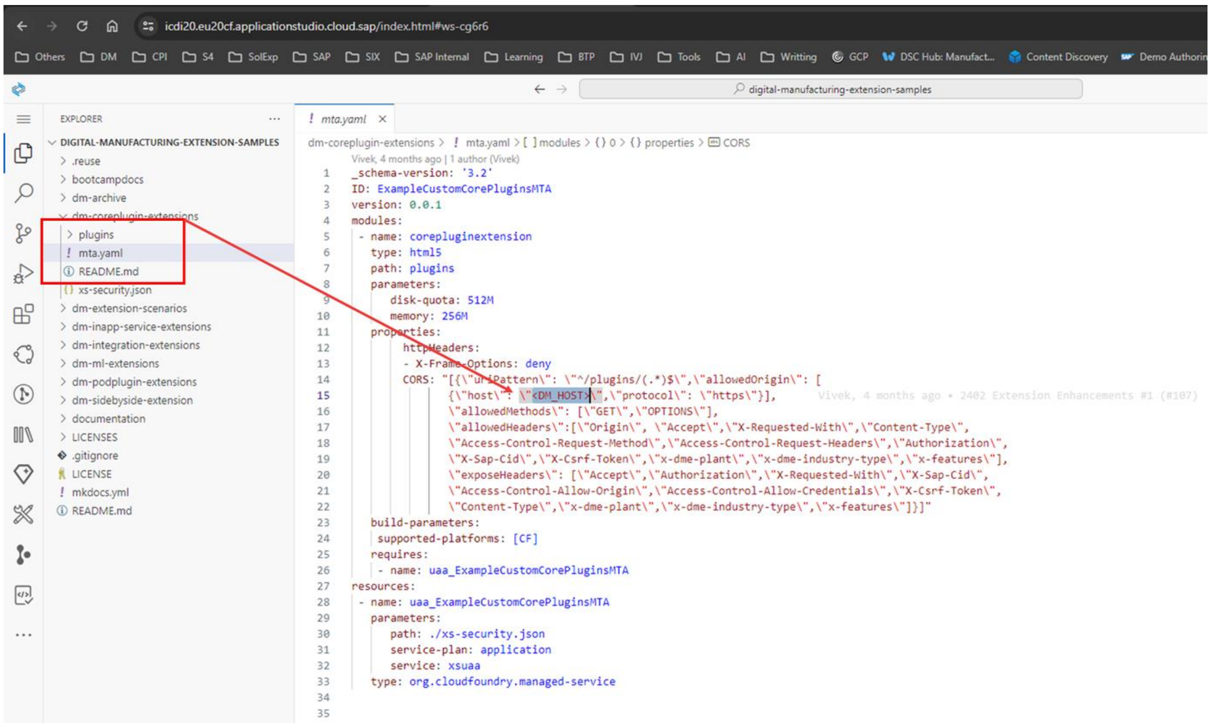
Source: Git



Would you like to open the cloned repository?

Get Started with SAP Business Application Studio
MyDevSpace
SAP Fiori
New Project from Template
Clone from Git
Import
Files & Folders
Projects (7)
vendor.ext.executionplugins
Path: /home/user/projects/digital-manufacturing-extension-samples/...
Open
Open in New Window
Cancel
Show all
Sample Apps (2)
SAP Fiori elements List Report Application
An app, built with SAP Fiori tools, to manage products based on the SAP Fiori elements List Report OData V2 floorplan.
Custom App built using Basic template
A custom app, built with SAP Fiori tools, to manage products. Making use of the Basic template, the app began as an empty page.

To prevent Cross-Origin Resource Sharing (CORS) issues, from the EXPLORER view, open mta.yaml file and replace '<DM_HOST>' with your own SAP Digital Manufacturing tenant host in Line #15.

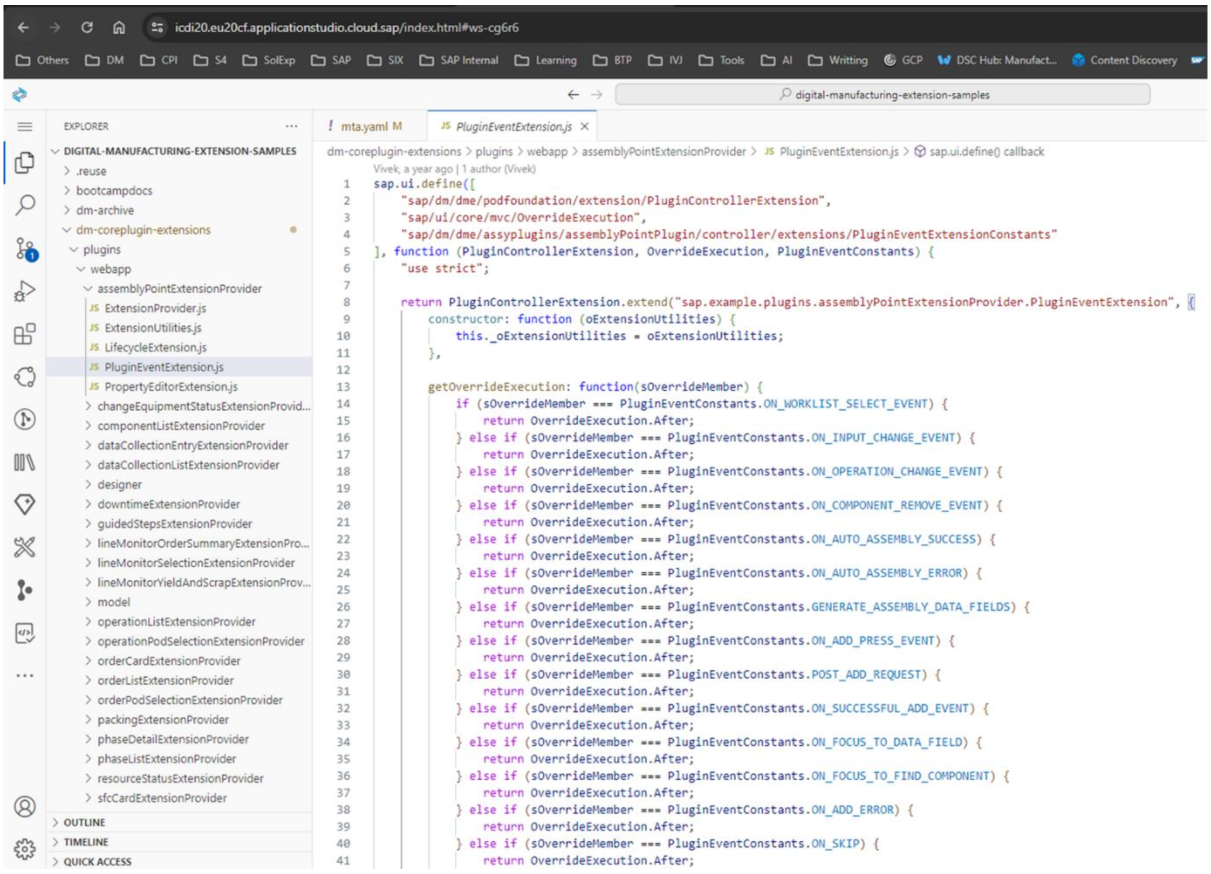


icdi20.eu20cf.applicationstudio.cloud.sap/index.html#ws-cg6r6
digital-manufacturing-extension-samples
EXPLORER
DIGITAL-MANUFACTURING-EXTENSION-SAMPLES
.reuse
bootcampdocs
dm-archive
dm-coreplugin-extensions
plugins
mta.yaml
README.md
xs-security.json
dm-extension-scenarios
dm-inapp-service-extensions
dm-integration-extensions
dm-ml-extensions
dm-podplugin-extensions
dm-sidebyside-extension
documentation
LICENSES
.gitignore
LICENSE
mkdocs.yml
README.md
mta.yaml
dm-coreplugin-extensions > mta.yaml > [ ] modules > { } 0 > { } properties > CORS
Vivek, 4 months ago | 1 author (Vivek)
Vivek, 4 months ago • 2402 Extension Enhancements #1 (#107)

```
 1  _schema-version: '3.2'
 2  ID: ExampleCustomCorePluginsMTA
 3  version: 0.0.1
 4  modules:
 5    - name: corepluginextension
 6      type: html5
 7      path: plugins
 8      parameters:
 9         disk-quota: 512M
10         memory: 256M
11      properties:
12           httpHeaders:
13           - X-Frame-Options: deny
14           CORS: "[{\"uriPattern\": \"^/plugins/(.*)$\",\"allowedOrigin\": [
15                 {\"host\": \"<DM_HOST>\",\"protocol\": \"https\"}],
16                 \"allowedMethods\": [\"GET\",\"OPTIONS\"],
17                 \"allowedHeaders\":[\"Origin\", \"Accept\",\"X-Requested-With\",\"Content-Type\",
18                 \"Access-Control-Request-Method\",\"Access-Control-Request-Headers\",\"Authorization\",
19                 \"X-Sap-Cid\",\"X-Csrf-Token\",\"x-dme-plant\",\"x-dme-industry-type\",\"x-features\"],
20                 \"exposeHeaders\": [\"Accept\",\"Authorization\",\"X-Requested-With\",\"X-Sap-Cid\",
21                 \"Access-Control-Allow-Origin\",\"Access-Control-Allow-Credentials\",\"X-Csrf-Token\",
22                 \"Content-Type\",\"x-dme-plant\",\"x-dme-industry-type\",\"x-features\"]}]"
23      build-parameters:
24       supported-platforms: [CF]
25      requires:
26       - name: uaa_ExampleCustomCorePluginsMTA
27  resources:
28   - name: uaa_ExampleCustomCorePluginsMTA
29     parameters:
30        path: ./xs-security.json
31        service-plan: application
32        service: xsuaa
33     type: org.cloudfoundry.managed-service
34
35
```

For example, if your SAP Digital Manufacturing tenant url is: "https://XXXXXXX.execution.eu20.dmc.cloud.sap/cp.portal/site#Shell-home", replace '<DM_HOST>' with "XXXXXXX.execution.eu20.dmc.cloud.sap" only.

Back to the EXPLORER view, expand dm-coreplugin-extensions > plugins > webapp > assemblyPointExtensionProvider path and open PluginEventExtension.js, where you can find the events that can be extended for the 'Assemble Components' POD Plugin.


icdi20.eu20cf.applicationstudio.cloud.sap/index.html#ws-cg6r6
Others
DM
CPI
S4
SolExp
SAP
SIX
SAP Internal
Learning
BTP
IVJ
Tools
AI
Writting
GCP
DSC Hub: Manufact...
Content Discovery
digital-manufacturing-extension-samples
EXPLORER
DIGITAL-MANUFACTURING-EXTENSION-SAMPLES
.reuse
bootcampdocs
dm-archive
dm-coreplugin-extensions
plugins
webapp
assemblyPointExtensionProvider
ExtensionProvider.js
ExtensionUtilities.js
LifecycleExtension.js
PluginEventExtension.js
PropertyEditorExtension.js
changeEquipmentStatusExtensionProvid...
componentListExtensionProvider
dataCollectionEntryExtensionProvider
dataCollectionListExtensionProvider
designer
downtimeExtensionProvider
guidedStepsExtensionProvider
lineMonitorOrderSummaryExtensionPro...
lineMonitorSelectionExtensionProvider
lineMonitorYieldAndScrapExtensionProv...
model
operationListExtensionProvider
operationPodSelectionExtensionProvider
orderCardExtensionProvider
orderListExtensionProvider
orderPodSelectionExtensionProvider
packingExtensionProvider
phaseDetailExtensionProvider
phaseListExtensionProvider
resourceStatusExtensionProvider
sfcCardExtensionProvider
OUTLINE
TIMELINE
QUICK ACCESS
mta.yaml M
PluginEventExtension.js
dm-coreplugin-extensions > plugins > webapp > assemblyPointExtensionProvider > PluginEventExtension.js > sap.ui.define() callback
Vivek, a year ago | 1 author (Vivek)

```
1   sap.ui.define([
2       "sap/dm/dme/podfoundation/extension/PluginControllerExtension",
3       "sap/ui/core/mvc/OverrideExecution",
4       "sap/dm/dme/assyplugins/assemblyPointPlugin/controller/extensions/PluginEventExtensionConstants"
5   ], function (PluginControllerExtension, OverrideExecution, PluginEventConstants) {
6       "use strict";
7
8       return PluginControllerExtension.extend("sap.example.plugins.assemblyPointExtensionProvider.PluginEventExtension", {
9           constructor: function (oExtensionUtilities) {
10              this._oExtensionUtilities = oExtensionUtilities;
11          },
12
13          getOverrideExecution: function(sOverrideMember) {
14              if (sOverrideMember === PluginEventConstants.ON_WORKLIST_SELECT_EVENT) {
15                  return OverrideExecution.After;
16              } else if (sOverrideMember === PluginEventConstants.ON_INPUT_CHANGE_EVENT) {
17                  return OverrideExecution.After;
18              } else if (sOverrideMember === PluginEventConstants.ON_OPERATION_CHANGE_EVENT) {
19                  return OverrideExecution.After;
20              } else if (sOverrideMember === PluginEventConstants.ON_COMPONENT_REMOVE_EVENT) {
21                  return OverrideExecution.After;
22              } else if (sOverrideMember === PluginEventConstants.ON_AUTO_ASSEMBLY_SUCCESS) {
23                  return OverrideExecution.After;
24              } else if (sOverrideMember === PluginEventConstants.ON_AUTO_ASSEMBLY_ERROR) {
25                  return OverrideExecution.After;
26              } else if (sOverrideMember === PluginEventConstants.GENERATE_ASSEMBLY_DATA_FIELDS) {
27                  return OverrideExecution.After;
28              } else if (sOverrideMember === PluginEventConstants.ON_ADD_PRESS_EVENT) {
29                  return OverrideExecution.After;
30              } else if (sOverrideMember === PluginEventConstants.POST_ADD_REQUEST) {
31                  return OverrideExecution.After;
32              } else if (sOverrideMember === PluginEventConstants.ON_SUCCESSFUL_ADD_EVENT) {
33                  return OverrideExecution.After;
34              } else if (sOverrideMember === PluginEventConstants.ON_FOCUS_TO_DATA_FIELD) {
35                  return OverrideExecution.After;
36              } else if (sOverrideMember === PluginEventConstants.ON_FOCUS_TO_FIND_COMPONENT) {
37                  return OverrideExecution.After;
38              } else if (sOverrideMember === PluginEventConstants.ON_ADD_ERROR) {
39                  return OverrideExecution.After;
40              } else if (sOverrideMember === PluginEventConstants.ON_SKIP) {
41                  return OverrideExecution.After;
```

Feel free to browse around the other folders for the other plugins, and get familiar with them as well.

In this blog post, I will extend 'ON_ADD_PRESS_EVENT' event from Line #28 above, which is triggered once workers press the 'Add' button in the 'Assemble Components' POD Plugin during Component Assembly steps in Work Center and Operation Activity PODs.

Note that ON_ADD_PRESS_EVENT event returns OverrideExecution.After:

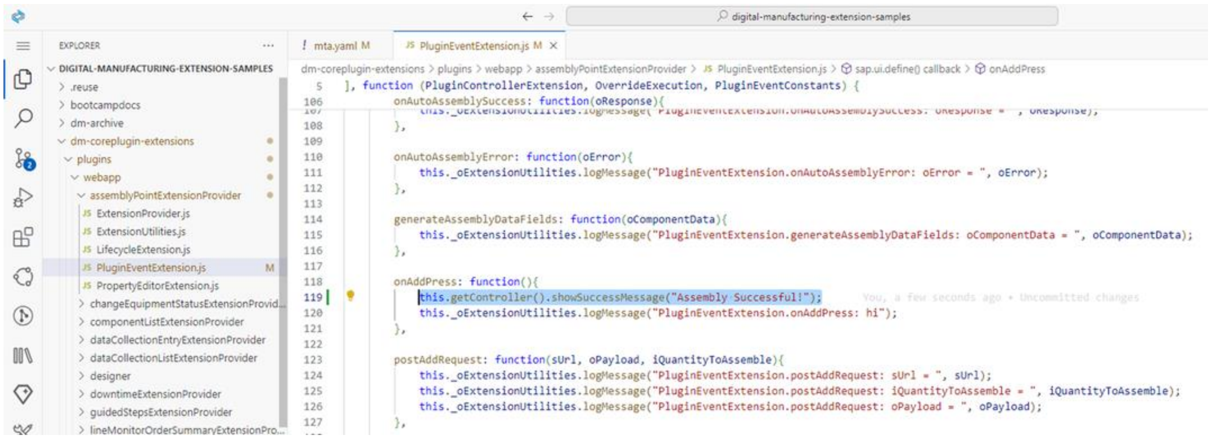
```
} else if (sOverrideMember === PluginEventConstants.ON_ADD_PRESS_EVENT) {
    return OverrideExecution.After;
}
```

Here, you can use:

- **Before**: Where your extension is appended **before** the standard execution flow.
- **After**: Where your extension is appended **after** the standard execution flow.

- **Instead**: Where your extension is executed **instead** of the standard execution flow.

To extend ON_ADD_PRESS_EVENT event, onAddPress function must be extended, which for educational purposes, I extended with a very simple showSuccessMessage() call:


digital-manufacturing-extension-samples
EXPLORER
DIGITAL-MANUFACTURING-EXTENSION-SAMPLES
.reuse
bootcampdocs
dm-archive
dm-coreplugin-extensions
plugins
webapp
assemblyPointExtensionProvider
ExtensionProvider.js
ExtensionUtilities.js
LifecycleExtension.js
PluginEventExtension.js
PropertyEditorExtension.js
changeEquipmentStatusExtensionProvid...
componentListExtensionProvider
dataCollectionEntryExtensionProvider
dataCollectionListExtensionProvider
designer
downtimeExtensionProvider
guidedStepsExtensionProvider
lineMonitorOrderSummaryExtensionPro...
mta.yaml M
PluginEventExtension.js M
dm-coreplugin-extensions > plugins > webapp > assemblyPointExtensionProvider > PluginEventExtension.js > sap.ui.define() callback > onAddPress

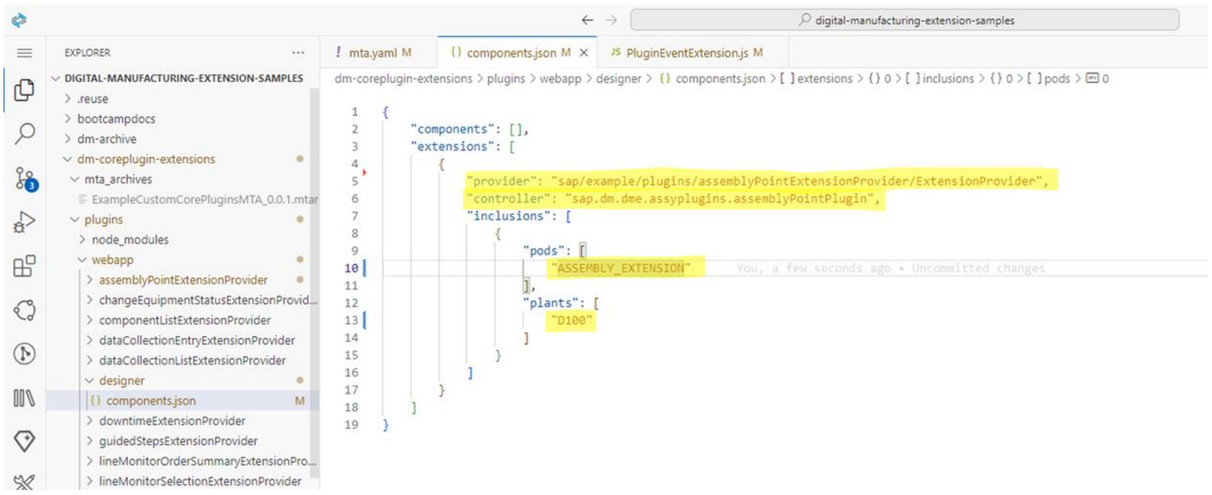
```javascript
], function (PluginControllerExtension, OverrideExecution, PluginEventConstants) {
        onAutoAssemblySuccess: function(oResponse){
            this._oExtensionUtilities.logMessage("PluginEventExtension.onAutoAssemblySuccess: oResponse = ", oResponse);
        },

        onAutoAssemblyError: function(oError){
            this._oExtensionUtilities.logMessage("PluginEventExtension.onAutoAssemblyError: oError = ", oError);
        },

        generateAssemblyDataFields: function(oComponentData){
            this._oExtensionUtilities.logMessage("PluginEventExtension.generateAssemblyDataFields: oComponentData = ", oComponentData);
        },

        onAddPress: function(){
            this.getController().showSuccessMessage("Assembly Successful!");      You, a few seconds ago • Uncommitted changes
            this._oExtensionUtilities.logMessage("PluginEventExtension.onAddPress: hi");
        },

        postAddRequest: function(sUrl, oPayload, iQuantityToAssemble){
            this._oExtensionUtilities.logMessage("PluginEventExtension.postAddRequest: sUrl = ", sUrl);
            this._oExtensionUtilities.logMessage("PluginEventExtension.postAddRequest: iQuantityToAssemble = ", iQuantityToAssemble);
            this._oExtensionUtilities.logMessage("PluginEventExtension.postAddRequest: oPayload = ", oPayload);
        },
```

Refer Below code for sample ADD event extensions which tell you how to call Production Process/DM APIs and Core Methods.

```javascript
    onAddPress: function(){

        let oPSM = this.getController().getPodSelectionModel();

        let oFormData = this.getController().dataModel.getData();

        let aAssemblyData = [];

        let oModesMap = {};

        const oConfig = this.getController().getConfiguration();


        if(oFormData.dataFields.ERP_SERIAL_NUMBER !=undefined)

        {

            // this.PPD_BASE_URL = this.getController().getPublicApiRestDataSourceUri() +
'/pe/api/v1/process/processDefinitions/start?


        var sUrl = this.getController().getPublicApiRestDataSourceUri() +
'/pe/api/v1/process/processDefinitions/start?key=REG_3a5d6569-3312-4186-9e12-
42b8be6fa2c5&async=false';

        var oPayload = {

          InPlant: this.getController().getPodController().getUserPlant(),

          InOrder: "1100000002",
          InComponent: oFormData.componentName
        };


        return new Promise((resolve, reject) => {
          this.getController().ajaxPostRequest(sUrl, oPayload, resolve, reject);
        }).then(function(oResponse) {
          if(oResponse.SERNP=="ZZ00"){
              this._oCoreExtension.base.processAddPress();
        }
            else{
                 MessageToast.show("Cannot assemble the component as Component is Serialized");
                return;
            }
        }.bind(this));



    } else{
                MessageToast.show("Required Data fields cannot be empty");
           }
        this._oExtensionUtilities.logMessage("PluginEventExtension.onAddPress: hi");
    },
```

Extensions can be enabled for specific POD Plugins, Plants and PODs, enabling flexibility and scalability.

To enable an extension, navigate to dm-coreplugin-extensions > plugins > webapp > designer path and open components.json file to declare the desired POD(s) and Plant(s) where each specific extension should be enabled for:



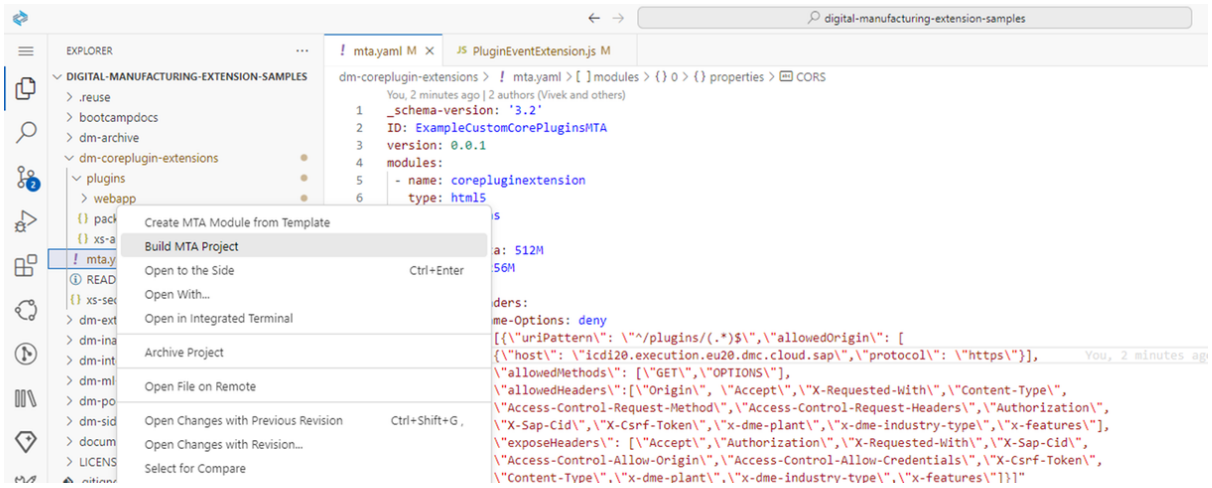
```json
{
    "components": [],
    "extensions": [
        {
            "provider": "sap/example/plugins/assemblyPointExtensionProvider/ExtensionProvider",
            "controller": "sap.dm.dme.assyplugins.assemblyPointPlugin",
            "inclusions": [
                {
                    "pods": [
                        "ASSEMBLY_EXTENSION"
                    ],
                    "plants": [
                        "D100"
                    ]
                }
            ]
        }
    ]
}
```

digital-manufacturing-extension-samples
EXPLORER
DIGITAL-MANUFACTURING-EXTENSION-SAMPLES
mta.yaml M
components.json M
PluginEventExtension.js M
dm-coreplugin-extensions > plugins > webapp > designer > components.json > extensions > 0 > inclusions > 0 > pods > 0
You, a few seconds ago • Uncommitted changes

And that´s it! The POD Plugin extension is ready to be deployed.

Deploy it

To deploy the extension, navigate to and right-click mta.yaml file from the EXPLORER view, and select 'Build MTA Project'.

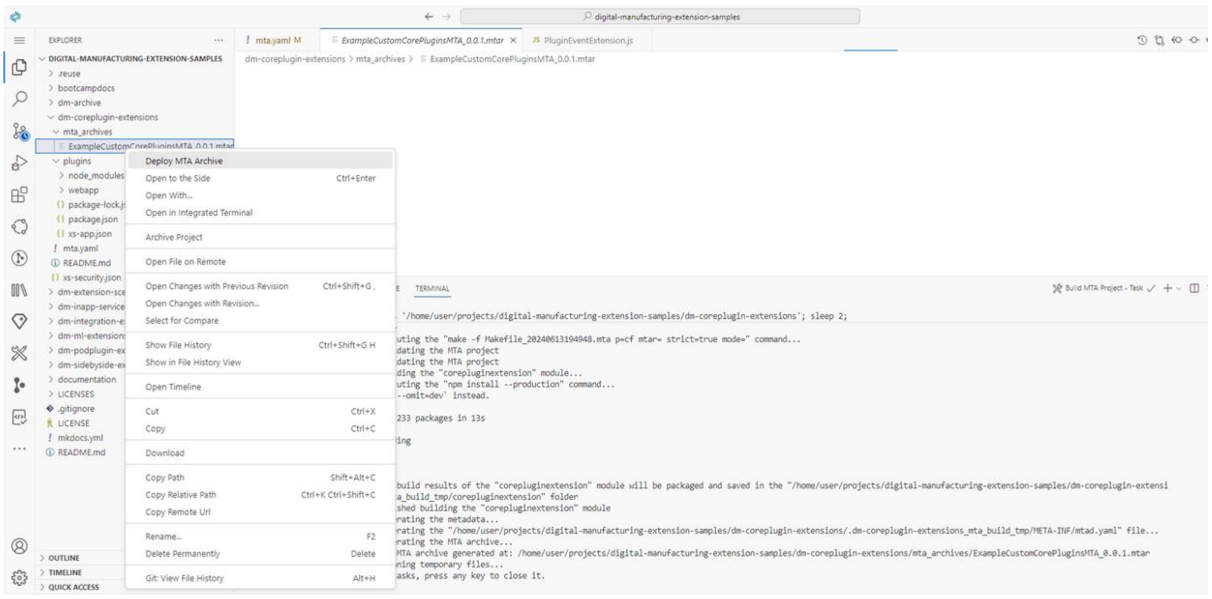


```yaml
_schema-version: '3.2'
ID: ExampleCustomCorePluginsMTA
version: 0.0.1
modules:
  - name: corepluginextension
    type: html5
```

```
s
a: 512M
56M
ders:
me-Options: deny
[{\"uriPattern\": \"^/plugins/(.*)$\",\"allowedOrigin\": [
{\"host\": \"icdi20.execution.eu20.dmc.cloud.sap\",\"protocol\": \"https\"}],
\"allowedMethods\": [\"GET\",\"OPTIONS\"],
\"allowedHeaders\":[\"Origin\", \"Accept\",\"X-Requested-With\",\"Content-Type\",
\"Access-Control-Request-Method\",\"Access-Control-Request-Headers\",\"Authorization\",
\"X-Sap-Cid\",\"X-Csrf-Token\",\"x-dme-plant\",\"x-dme-industry-type\",\"x-features\"],
\"exposeHeaders\": [\"Accept\",\"Authorization\",\"X-Requested-With\",\"X-Sap-Cid\",
\"Access-Control-Allow-Origin\",\"Access-Control-Allow-Credentials\",\"X-Csrf-Token\",
\"Content-Type\",\"x-dme-plant\",\"x-dme-industry-type\",\"x-features\"]}]"
```

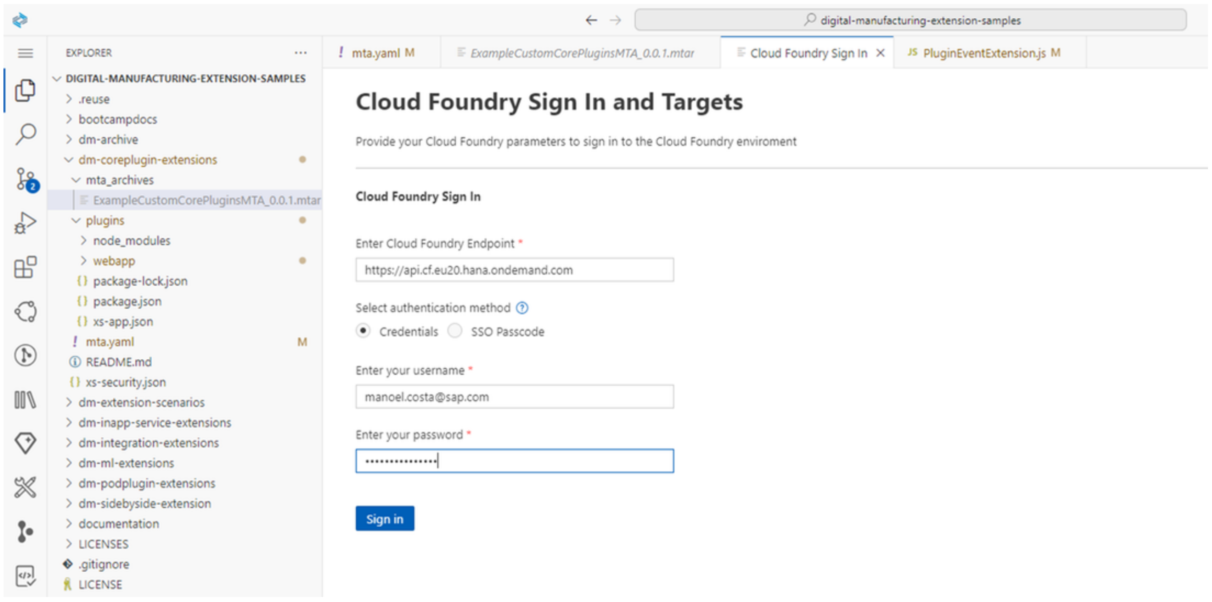
digital-manufacturing-extension-samples
EXPLORER
DIGITAL-MANUFACTURING-EXTENSION-SAMPLES
mta.yaml M
PluginEventExtension.js M
dm-coreplugin-extensions > mta.yaml > modules > 0 > properties > CORS
You, 2 minutes ago | 2 authors (Vivek and others)
Create MTA Module from Template
Build MTA Project
Open to the Side Ctrl+Enter
Open With...
Open in Integrated Terminal
Archive Project
Open File on Remote
Open Changes with Previous Revision Ctrl+Shift+G ,
Open Changes with Revision...
Select for Compare

Once the build is complete, expand 'mta_archives' folder to find the recently created .mtar build file, right-click it and select 'Deploy MTA Archive'.



```
 '/home/user/projects/digital-manufacturing-extension-samples/dm-coreplugin-extensions'; sleep 2;

uting the "make -f Makefile_20240613194948.mta p=cf mtar= strict=true mode=" command...
dating the MTA project
dating the MTA project
ding the "corepluginextension" module...
uting the "npm install --production" command...
--omit=dev` instead.

233 packages in 13s

ing


build results of the "corepluginextension" module will be packaged and saved in the "/home/user/projects/digital-manufacturing-extension-samples/dm-coreplugin-extensi
a_build_tmp/corepluginextension" folder
shed building the "corepluginextension" module
rating the metadata...
rating the "/home/user/projects/digital-manufacturing-extension-samples/dm-coreplugin-extensions/.dm-coreplugin-extensions_mta_build_tmp/META-INF/mtad.yaml" file...
rating the MTA archive...
MTA archive generated at: /home/user/projects/digital-manufacturing-extension-samples/dm-coreplugin-extensions/mta_archives/ExampleCustomCorePluginsMTA_0.0.1.mtar
ning temporary files...
asks, press any key to close it.
```

When doing it for the 1st time only, Sign In to SAP BTP Subaccount Cloud Foundry Environment and select a Target.



Cloud Foundry Sign In and Targets

Provide your Cloud Foundry parameters to sign in to the Cloud Foundry enviroment

Cloud Foundry Sign In

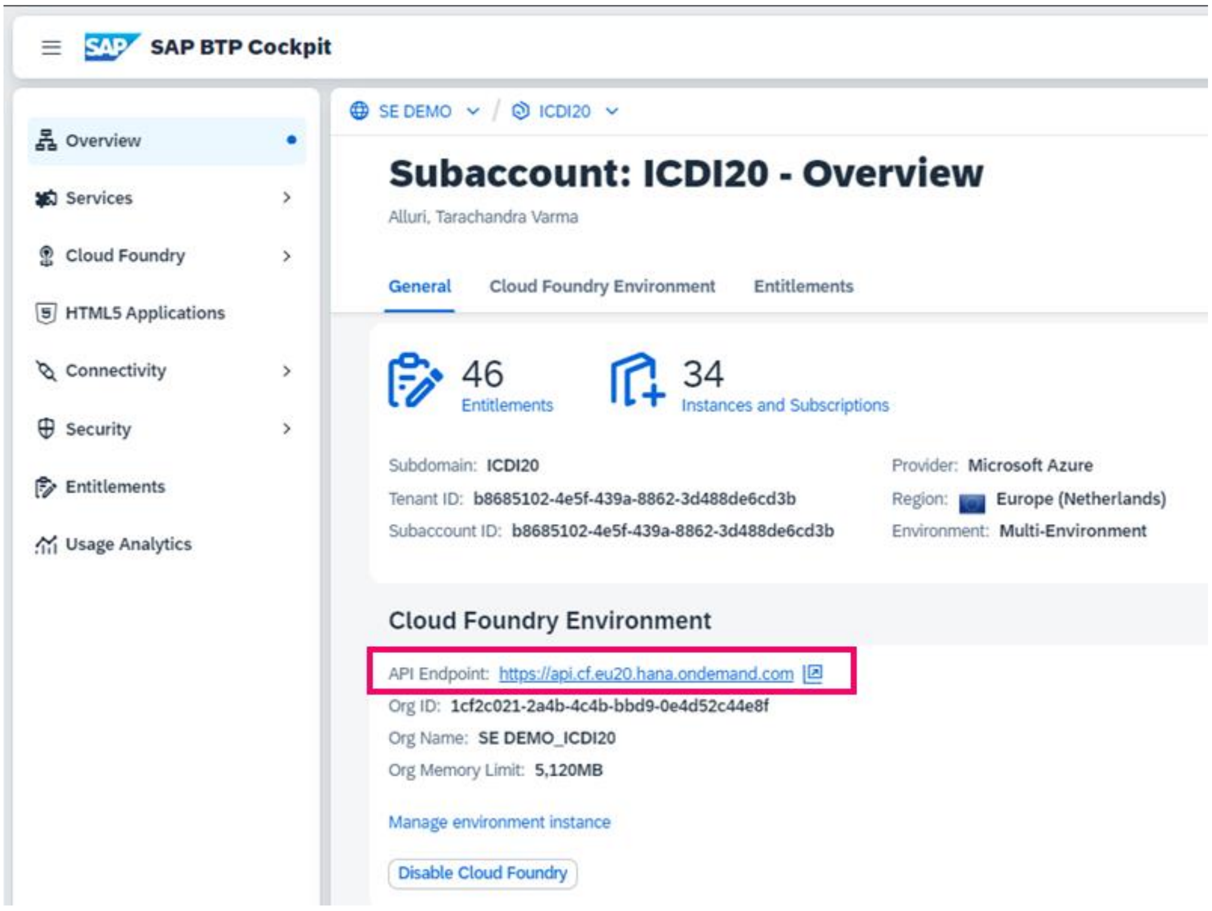
Enter Cloud Foundry Endpoint *
https://api.cf.eu20.hana.ondemand.com

Select authentication method
[● Credentials] ○ SSO Passcode

Enter your username *
manoel.costa@sap.com

Enter your password *
•••••••••••••••

The Cloud Foundry Endpoint information comes from the SAP BTP Cockpit, see below, and authentication can be done via Credentials or SSO Passcode as you wish.



Subaccount: ICDI20 - Overview

Alluri, Tarachandra Varma

46 Entitlements

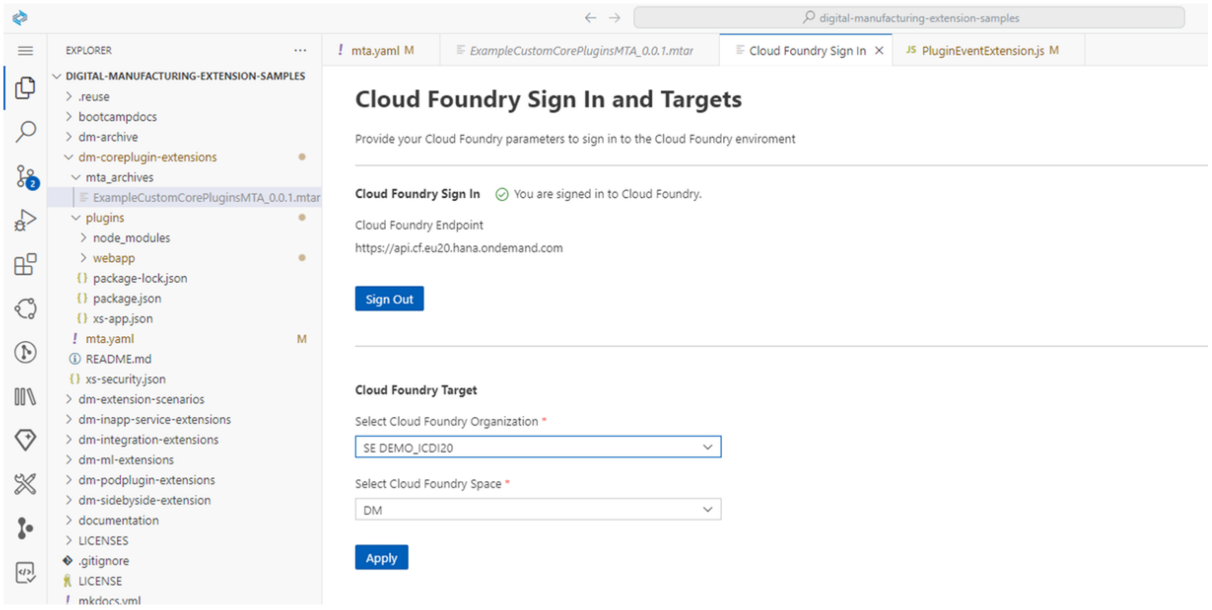
34 Instances and Subscriptions

Subdomain: ICDI20
Tenant ID: b8685102-4e5f-439a-8862-3d488de6cd3b
Subaccount ID: b8685102-4e5f-439a-8862-3d488de6cd3b
Provider: Microsoft Azure
Region: Europe (Netherlands)
Environment: Multi-Environment

Cloud Foundry Environment

API Endpoint: https://api.cf.eu20.hana.ondemand.com
Org ID: 1cf2c021-2a4b-4c4b-bbd9-0e4d52c44e8f
Org Name: SE DEMO_ICDI20
Org Memory Limit: 5,120MB

Once Cloud Foundry Sign In is done, select a Cloud Fountry Target, meaning
the Organization and Space where the extension will be deployed to, and click 'Apply'.



Cloud Foundry Sign In and Targets

Provide your Cloud Foundry parameters to sign in to the Cloud Foundry enviroment

Cloud Foundry Sign In ✓ You are signed in to Cloud Foundry.

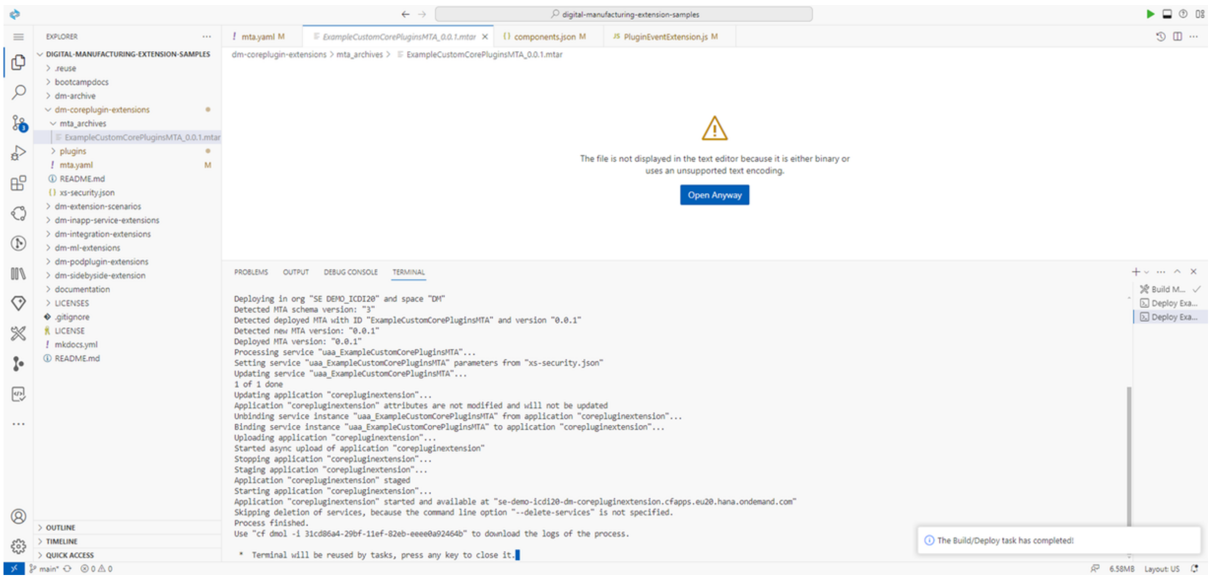
Cloud Foundry Endpoint
https://api.cf.eu20.hana.ondemand.com

Cloud Foundry Target

Select Cloud Foundry Organization *
SE DEMO_ICDI20

Select Cloud Foundry Space *
DM

The extension will now be deployed to the SAP BTP Subaccount Cloud Foundry Environment, which might take a few minutes, hang tight.

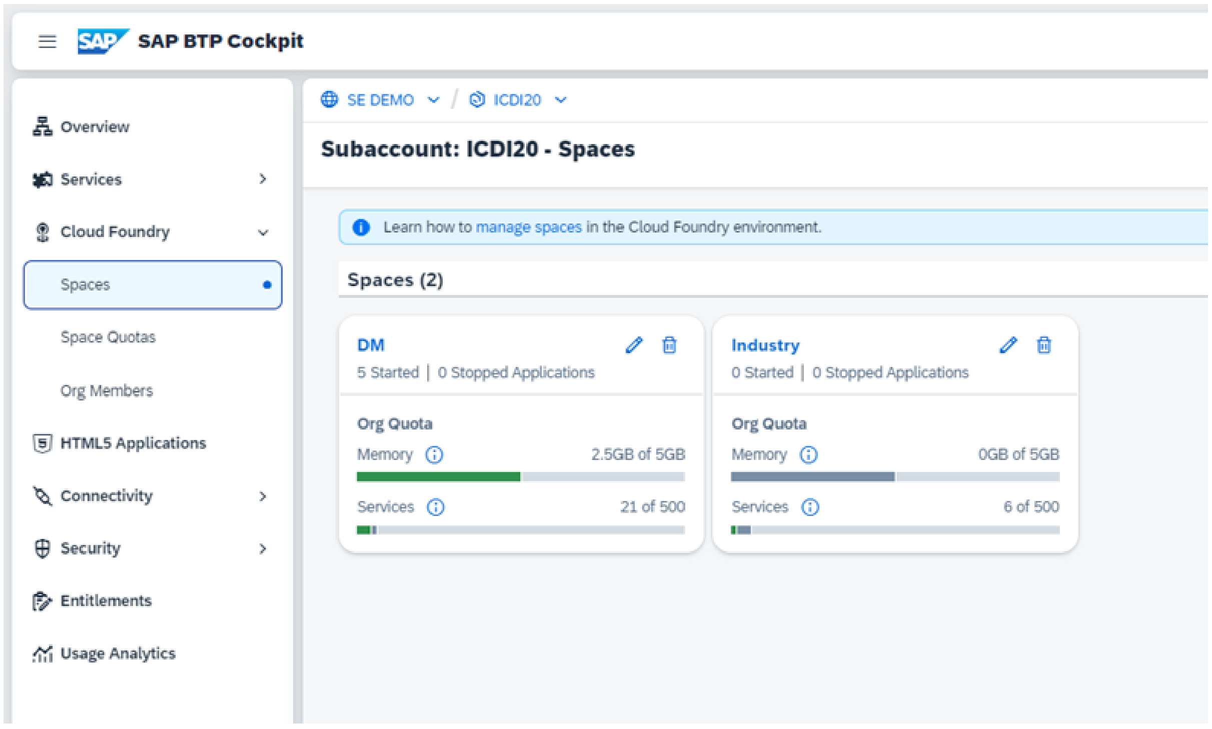


The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding.

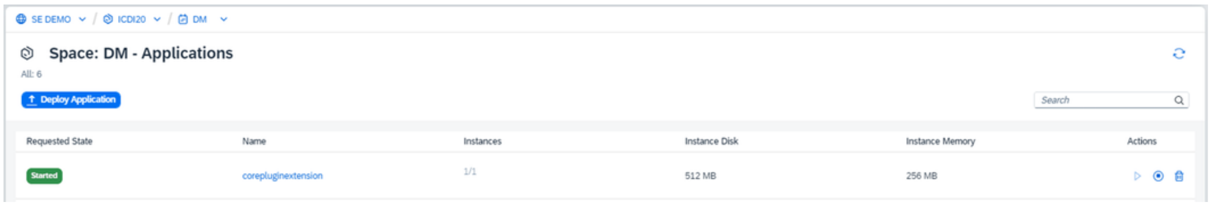
```
Deploying in org "SE DEMO_ICDI20" and space "DM"
Detected MTA schema version: "3"
Detected deployed MTA with ID "ExampleCustomCorePluginsMTA" and version "0.0.1"
Detected new MTA version: "0.0.1"
Deployed MTA version: "0.0.1"
Processing service "uaa_ExampleCustomCorePluginsMTA"...
Setting service "uaa_ExampleCustomCorePluginsMTA" parameters from "xs-security.json"
Updating service "uaa_ExampleCustomCorePluginsMTA"...
1 of 1 done
Updating application "corepluginextension"...
Application "corepluginextension" attributes are not modified and will not be updated
Unbinding service instance "uaa_ExampleCustomCorePluginsMTA" from application "corepluginextension"...
Binding service instance "uaa_ExampleCustomCorePluginsMTA" to application "corepluginextension"...
Uploading application "corepluginextension"...
Started async upload of application "corepluginextension"
Stopping application "corepluginextension"...
Staging application "corepluginextension"...
Application "corepluginextension" staged
Starting application "corepluginextension"...
Application "corepluginextension" started and available at "se-demo-icdi20-dm-corepluginextension.cfapps.eu20.hana.ondemand.com"
Skipping deletion of services, because the command line option "--delete-services" is not specified.
Process finished.
Use "cf dmol -i 31cd86a4-29bf-11ef-82eb-eeee0a92464b" to download the logs of the process.

 *  Terminal will be reused by tasks, press any key to close it.
```

The Build/Deploy task has completed!

Once completed, you can navigate to the respective SAP BTP Cockpit and check the selected Cloud Foundry Target Space:



To find the application deployment as 'Started':



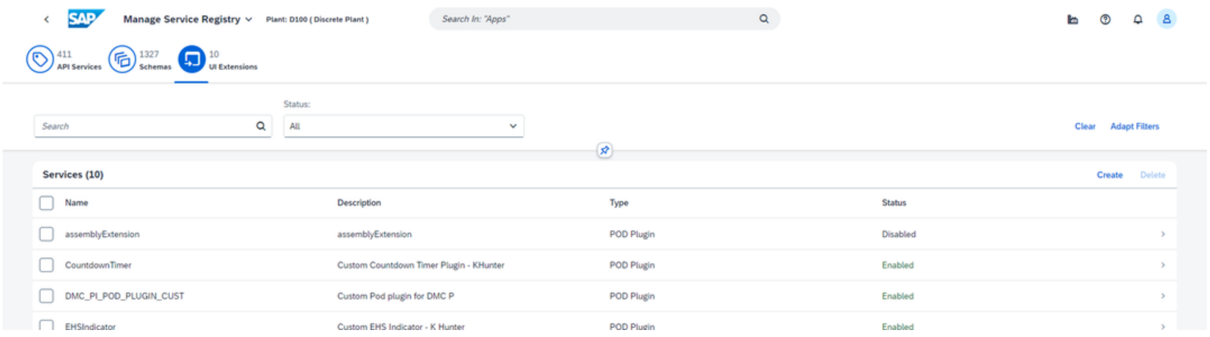
Requested State	Name	Instances	Instance Disk	Instance Memory	Actions
Started	corepluginextension	1/1	512 MB	256 MB	

If you don´t see it, please go back and review your whole setup.

Enable it

Once deployed, the extension must be enabled in in SAP Digital Manufacturing Service Registry.

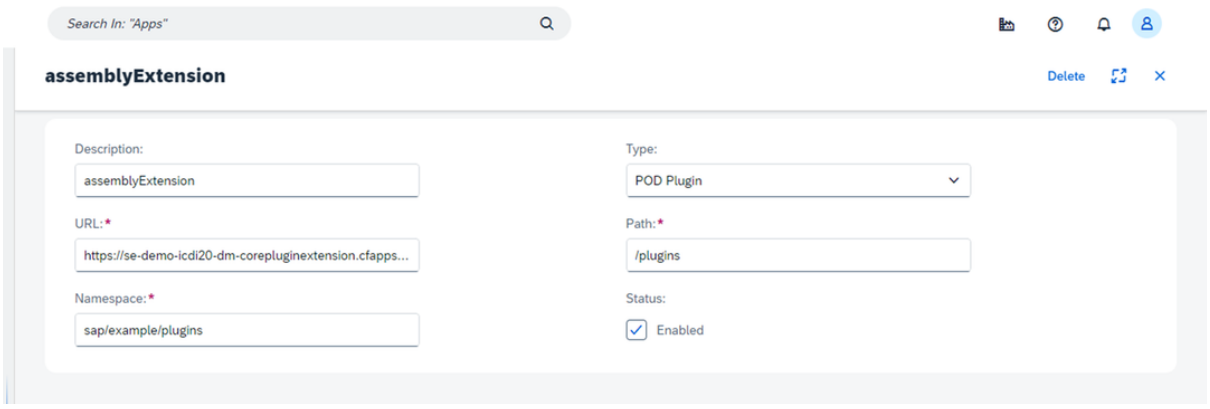
To do so, open Manage Service Registry app, select 'UI Extensions' and click 'Create':



Services (10)

Name	Description	Type	Status
assemblyExtension	assemblyExtension	POD Plugin	Disabled
CountdownTimer	Custom Countdown Timer Plugin - KHunter	POD Plugin	Enabled
DMC_PI_POD_PLUGIN_CUST	Custom Pod plugin for DMC P	POD Plugin	Enabled
EHSIndicator	Custom EHS Indicator - K Hunter	POD Plugin	Enabled

Enter the 'UI Extension' info accordingly, as shown below.

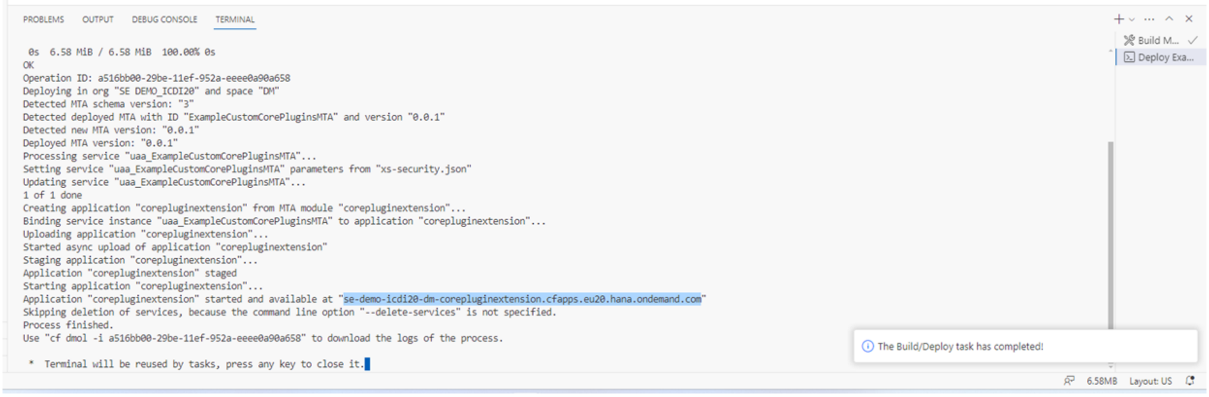


assemblyExtension

Description: assemblyExtension	Type: POD Plugin
URL:* https://se-demo-icdi20-dm-corepluginextension.cfapps...	Path:* /plugins
Namespace:* sap/example/plugins	Status: ✓ Enabled

Where:

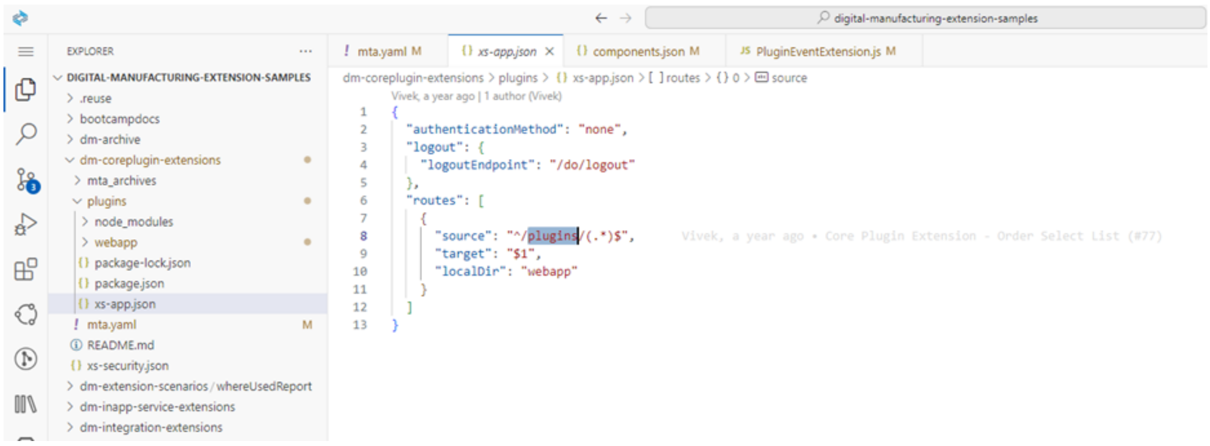
- **Name:** Anything
- **Description**: Anything
- **Type**: 'POD Plugin'
- **URL**: Shows up in SAP Business Application Studio right after the deployment is completed, see
  below:



```
 0s  6.58 MiB / 6.58 MiB  100.00% 0s
OK
Operation ID: a516bb00-29be-11ef-952a-eeee0a90a658
Deploying in org "SE DEMO_ICDI20" and space "DM"
Detected MTA schema version: "3"
Detected deployed MTA with ID "ExampleCustomCorePluginsMTA" and version "0.0.1"
Detected new MTA version: "0.0.1"
Deployed MTA version: "0.0.1"
Processing service "uaa_ExampleCustomCorePluginsMTA"...
Setting service "uaa_ExampleCustomCorePluginsMTA" parameters from "xs-security.json"
Updating service "uaa_ExampleCustomCorePluginsMTA"...
1 of 1 done
Creating application "corepluginextension" from MTA module "corepluginextension"...
Binding service instance "uaa_ExampleCustomCorePluginsMTA" to application "corepluginextension"...
Uploading application "corepluginextension"...
Started async upload of application "corepluginextension"
Staging application "corepluginextension"...
Application "corepluginextension" staged
Starting application "corepluginextension"...
Application "corepluginextension" started and available at "se-demo-icdi20-dm-corepluginextension.cfapps.eu20.hana.ondemand.com"
Skipping deletion of services, because the command line option "--delete-services" is not specified.
Process finished.
Use "cf dmol -i a516bb00-29be-11ef-952a-eeee0a90a658" to download the logs of the process.

 *  Terminal will be reused by tasks, press any key to close it.
```

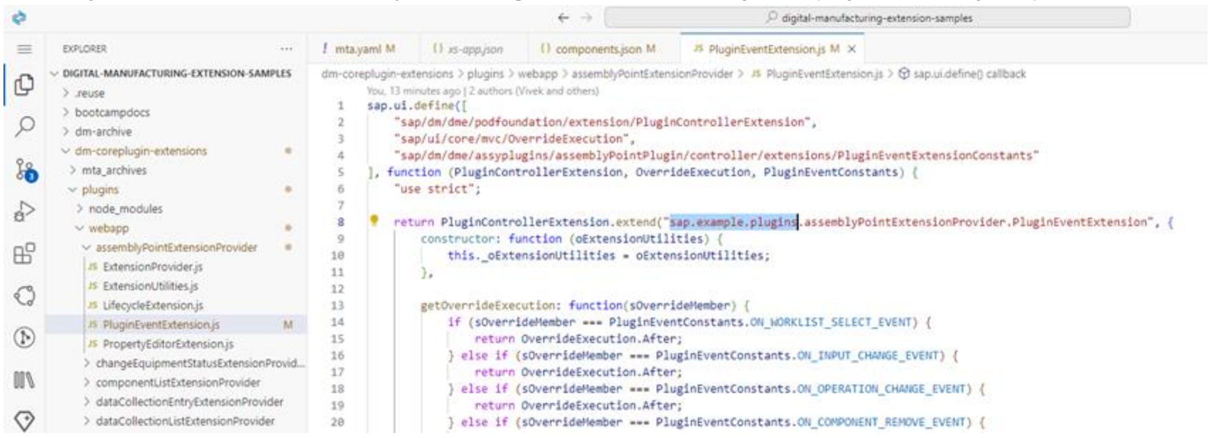
The Build/Deploy task has completed!

- **Path**: Is defined in **xs-app.json** as 'source', see below:



```
{
  "authenticationMethod": "none",
  "logout": {
    "logoutEndpoint": "/do/logout"
  },
  "routes": [
    {
      "source": "^/plugins/(.*)$",
      "target": "$1",
      "localDir": "webapp"
    }
  ]
}
```

- **Namespace**: Is defined, for example, in **PluginEventExtension.js** file **(replace "." by "/")**



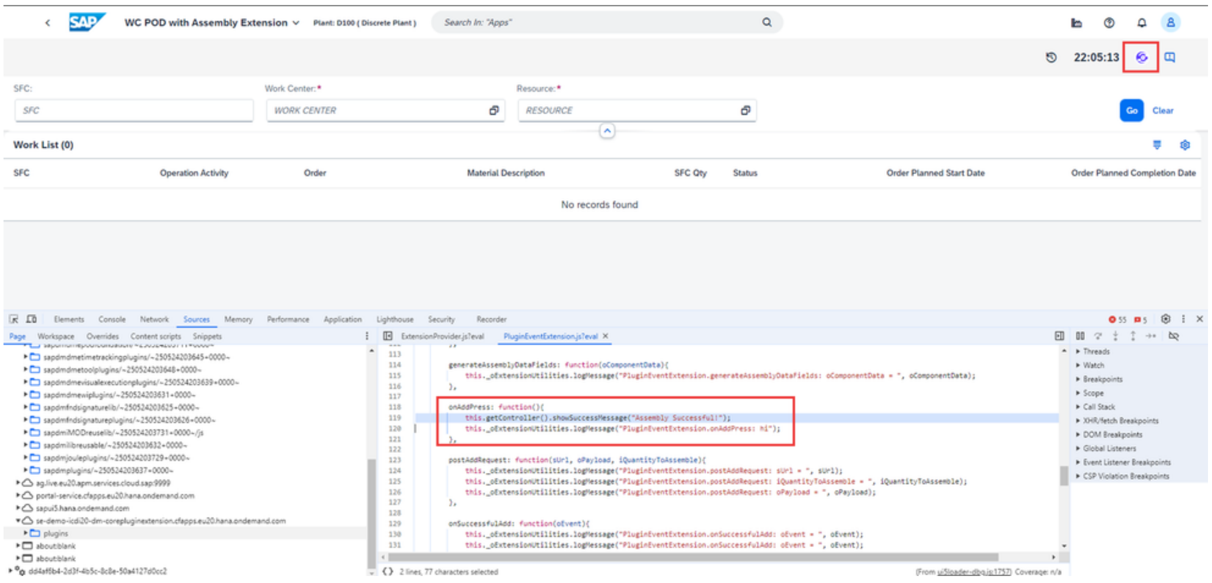
```
sap.ui.define([
    "sap/dm/dme/podfoundation/extension/PluginControllerExtension",
    "sap/ui/core/mvc/OverrideExecution",
    "sap/dm/dme/assyplugins/assemblyPointPlugin/controller/extensions/PluginEventExtensionConstants"
], function (PluginControllerExtension, OverrideExecution, PluginEventConstants) {
    "use strict";

    return PluginControllerExtension.extend("sap.example.plugins.assemblyPointExtensionProvider.PluginEventExtension", {
        constructor: function (oExtensionUtilities) {
            this._oExtensionUtilities = oExtensionUtilities;
        },

        getOverrideExecution: function(sOverrideMember) {
            if (sOverrideMember === PluginEventConstants.ON_WORKLIST_SELECT_EVENT) {
                return OverrideExecution.After;
            } else if (sOverrideMember === PluginEventConstants.ON_INPUT_CHANGE_EVENT) {
                return OverrideExecution.After;
            } else if (sOverrideMember === PluginEventConstants.ON_OPERATION_CHANGE_EVENT) {
                return OverrideExecution.After;
            } else if (sOverrideMember === PluginEventConstants.ON_COMPONENT_REMOVE_EVENT) {
```

- **Status**: Check as **Enabled**.

and click 'Create' to create it.

Consume it

Once the 'UI Extension' is created, you can open the very specific POD at the very specific Plant for which you previously enabled the extension for at components.json file, where you should be able to notice two important things:



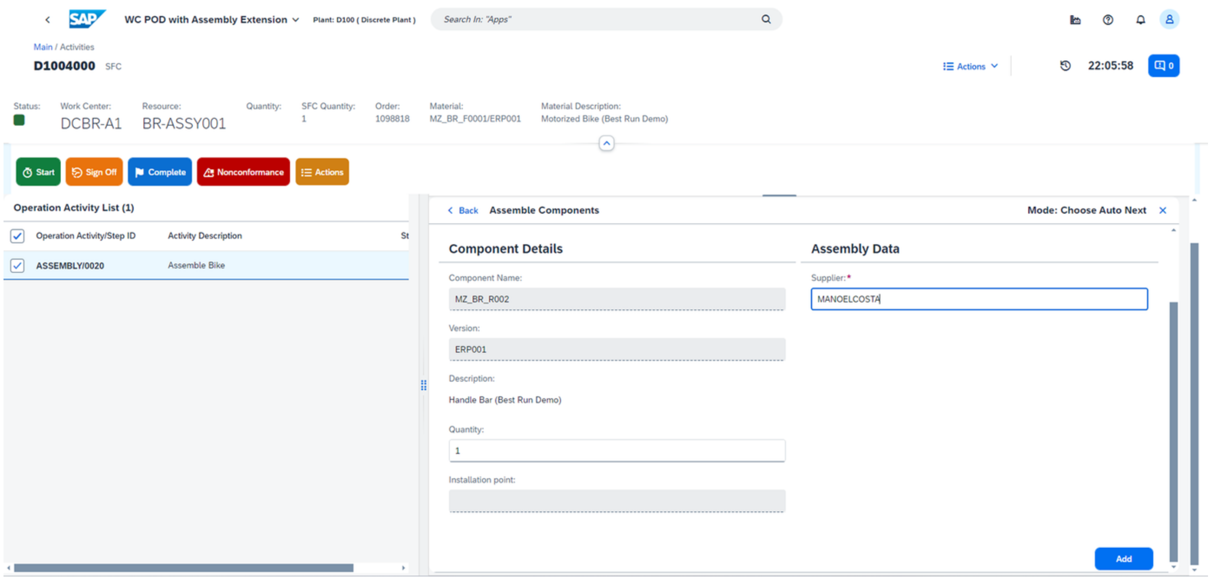
```
generateAssemblyDataFields: function(oComponentData){
    this._oExtensionUtilities.logMessage("PluginEventExtension.generateAssemblyDataFields: oComponentData = ", oComponentData);
},

onAddPress: function(){
    this.getController().showSuccessMessage("Assembly Successful!");
    this._oExtensionUtilities.logMessage("PluginEventExtension.onAddPress: hi");
},

postAddRequest: function(sUrl, oPayload, iQuantityToAssemble){
    this._oExtensionUtilities.logMessage("PluginEventExtension.postAddRequest: sUrl = ", sUrl);
    this._oExtensionUtilities.logMessage("PluginEventExtension.postAddRequest: iQuantityToAssemble = ", iQuantityToAssemble);
    this._oExtensionUtilities.logMessage("PluginEventExtension.postAddRequest: oPayload = ", oPayload);
},

onSuccessfulAdd: function(oEvent){
    this._oExtensionUtilities.logMessage("PluginEventExtension.onSuccessfulAdd: oEvent = ", oEvent);
    this._oExtensionUtilities.logMessage("PluginEventExtension.onSuccessfulAdd: oEvent = ", oEvent);
```

WC POD with Assembly Extension
Plant: D100 (Discrete Plant)
22:05:13
SFC:
Work Center:*
Resource:*
Go
Clear
Work List (0)
SFC | Operation Activity | Order | Material Description | SFC Qty | Status | Order Planned Start Date | Order Planned Completion Date
No records found
Elements Console Network Sources Memory Performance Application Lighthouse Security Recorder
ExtensionProvider.js?eval
PluginEventExtension.js?eval
2 lines, 77 characters selected

1. In the POD top right corner, you should see a purple icon, meaning that extensions are active for this POD.
2. In Developer Console > Sources tab, if you scroll down to the bottom, you should be able to see your plugin extension is properly loaded and available.

If you don´t see both of these, please go back and review your whole setup.

Proceed as normal and navigate to 'Assemble Components' plugin, which was extended in this example, to give it a try.

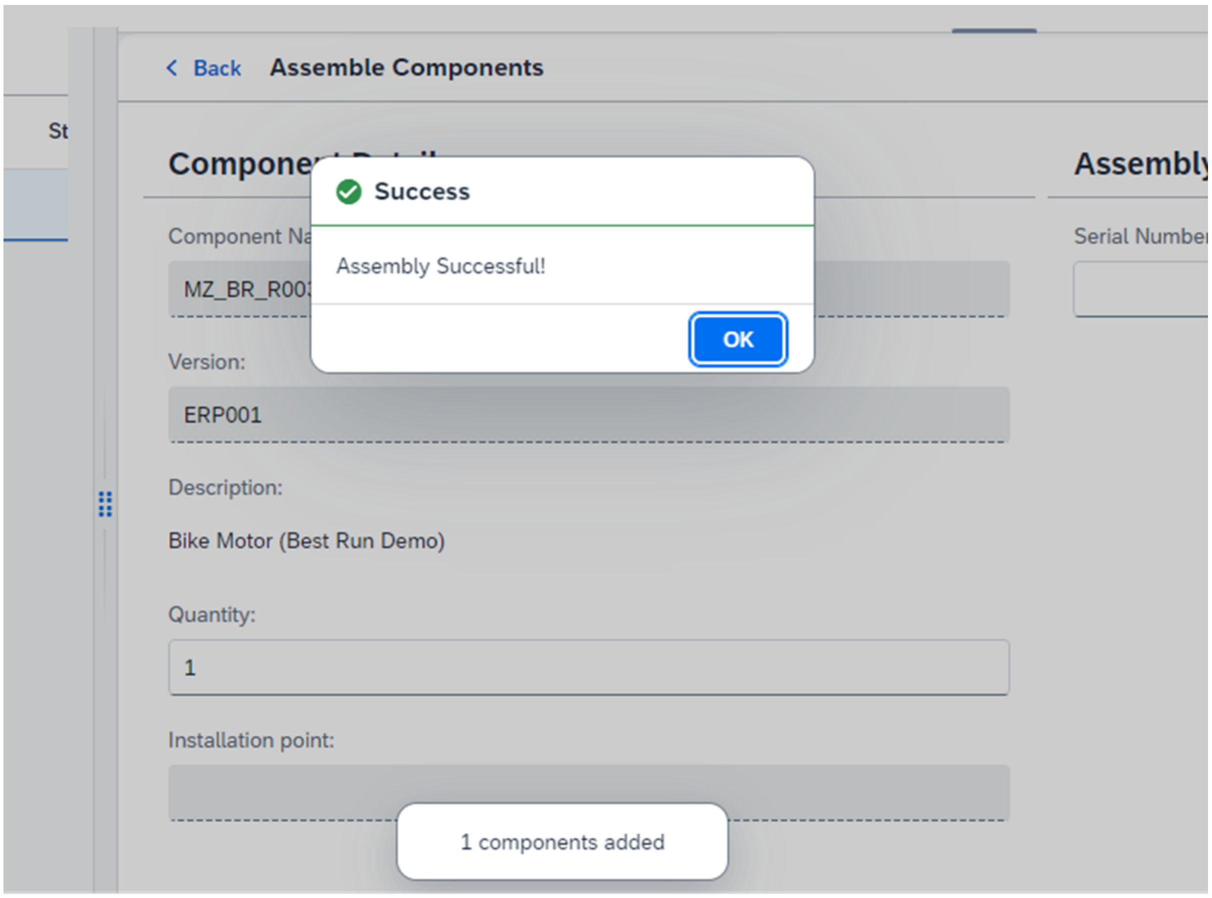


SFC	D1004000
Status:	🟩
Work Center:	DCBR-A1
Resource:	BR-ASSY001
SFC Quantity:	1
Order:	1098818
Material:	MZ_BR_F0001/ERP001
Material Description:	Motorized Bike (Best Run Demo)

Operation Activity/Step ID	Activity Description
ASSEMBLY/0020	Assemble Bike

Component Name:	MZ_BR_R002
Version:	ERP001
Description:	Handle Bar (Best Run Demo)
Quantity:	1
Installation point:	
Supplier:*	MANOELCOSTA

And, as soon as 'Add' button is clicked, you should see the extension being triggered.



Component Na

MZ_BR_R00

Assembly Successful!

Version:

ERP001

Description:

Bike Motor (Best Run Demo)

Quantity:

1

Installation point:

1 components added

Please keep in mind that, for educational purposes, in this blog post I covered a very simple extension to the 'Assembly Components' POD Plugin, but these same steps and principles can be used for further extensions and more complex use cases.