

Technical Articles



Sushma Kudum

February 14, 2019 6 minute read

Steps to create a HDI container type DB using SAP WEB IDE in the Cloud Foundry

[Follow](#)

[RSS feed](#)

[Like](#)

6 Likes 1,284 View 2 Comments

Hi Autodidact,

This blog is for enthusiastic people who want to learn to create HDI container type Database using SAP WEB IDE and deploy the same in Cloud Foundry and can use SAP WEB IDE to perform DML/DDL operations on DB.

We will go through step by step processes to achieve the same.

Pre-requisites required:-

Accounts and infrastructure:-

1. SAP Cloud platform Cloud Foundry account.
2. SAP Web Ide Full stack service enabled.

Skills:-

1. Basic knowledge of Cloud Foundry and SAP Cloud Platform.

2. Basic knowledge of SQL.

Let's start:

Login into Cockpit and navigate to your trial account and then to services and then open SAP WEB IDE full stack.

The screenshot shows the SAP Cloud Platform Trial Cockpit. On the left, there's a sidebar with icons for Regions, Services, Useful Links, and Legal Information. The main area is titled "Your SAP Cloud Platform Trial". It displays two trial options: "Cloud Foundry Trial" and "Neo Trial". The "Cloud Foundry Trial" box has a blue bar at the bottom, while the "Neo Trial" box has an orange bar. Below the trials, there's a large image of a mountain range with clouds and a hot air balloon. The bottom section contains three cards: "Hello Sushma", "Environments", and "Next Steps".

Your SAP Cloud Platform Trial

Cloud Foundry Trial
Infrastructure: AWS, GCP, or Azure
Trial Duration: 90 days limited trial

Neo Trial
Infrastructure: SAP
Trial Duration: Unlimited

Hello Sushma
This is the central Web-based user interface for administrators, providing access to a number of functions for configuring and managing applications and connecting them to databases in the cloud platform.

Environments
SAP Cloud Platform provides two different development environments:
Cloud Foundry and Neo

Next Steps

- Deploy your first application
- Launch SAP Web IDE

Overview

Applications

Java Applications

HTML5 Applications

HANA XS Applications

Subscriptions

Services

Solutions

SAP HANA / SAP ASE

Database Systems

Databases & Schemas

Useful Links

 Git repositories.

 through networks with high latency.

 with a low-code, graphical toolset.

SAP Web IDE Full-Stack

Enabled

 Create and extend SAP full-stack applications for browsers and mobile devices.

DevOps

Agent Activation for Dynatrace

Enabled

 Connect your Java applications to a Dynatrace SaaS monitoring environment.

Java Apps Lifecycle Management

Enabled

 Manage the lifecycle of Java applications by using a REST API.

Java Profiling

Enabled

 Profile and analyze your Java applications.

Click on SAP Web IDE Full -Stack grid and then “Go to Service” link.

 Overview

JavaScript Rules

 [Learn more](#)

 [Provide Feedback](#)

 [Useful Links](#)

 [Legal Information](#)

<https://webidearn-i50539trial.dispatcher.hanatrial.ondemand.com>

With SAP Cloud Platform Web IDE Full-Stack, you can easily develop, test, build, deploy, and extend role-based, consumer-grade apps for business users. Create applications rapidly and deliver an outstanding user experience. Developers can extend or build SAP Fiori apps, create new SaaS solutions, extend SAP S/4HANA cloud services, develop hybrid mobile applications, and build IoT apps for SAP Leonardo, using the UI development toolkit for HTML5 (SAPUI5) for desktop and mobile devices, the SAP HANA toolset, and Java programming language and technologies. Integrate with other services that run on SAP Cloud Platform, such as SAP Fiori Cloud apps, Git integration, mobile services, IoT services, and more.

Cloud Foundry: Currently not available for Cloud Foundry environment

NEO:

 Brazil (São Paulo)	 Canada (Toronto)
 Europe (Amsterdam)	
 Europe (Frankfurt)	 Europe (Rot)
 Europe (Rot) - Trial	 Japan (Tokyo)
 KSA (Riyadh)	 Russia (Moscow)
 UAE (Dubai)	 US East (Ashburn)
 US East (Sterling)	
 US West (Chandler)	
 Australia (Sydney)	

Take Action

[Configure Service](#)

[Logs](#)

[Go to Service](#)

Additional Resources

[Documentation](#)

[Getting Started](#)

It will navigate to SAP web ide in that select my workbench which will navigate you to editor page where you start creating a project as below.

Create an MTAR project from the project template

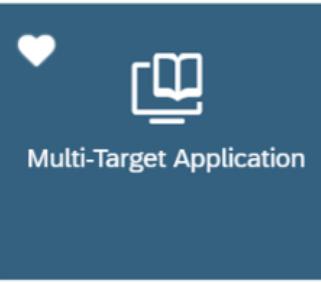


New >
Import >
Convert >
Close All Alt+Shift+W
Save Ctrl+S
Save All Ctrl+Shift+S
Git >

Folder Ctrl+Alt+Shift+N
Extension Project Ctrl+Alt+Shift+E
Project from Template Ctrl+Alt+Shift+O
Project from...
Project from Template (Ctrl+Alt+Shift+O)
Quick Start with Layout Editor
Adaptation Project

New Multi-Target Application

Template Selection



Multi-Target Application



SAPUI5 Application



SAP Fiori Worklist
Application

Previous

Next

Template Selection

Basic Information

Template Customization

Confirmation

X

New Multi-Target Application

Basic Information

Project Name*

FirstFullStackApp

Previous

Next

New Multi-Target Application

Template Customization

MTA Details

Application ID*

FirstFullStackApp

Application Version*

0.0.1

Description

 Use HTML5 Application Repository

Previous

Next

Finish

Run

New Multi-Target Application

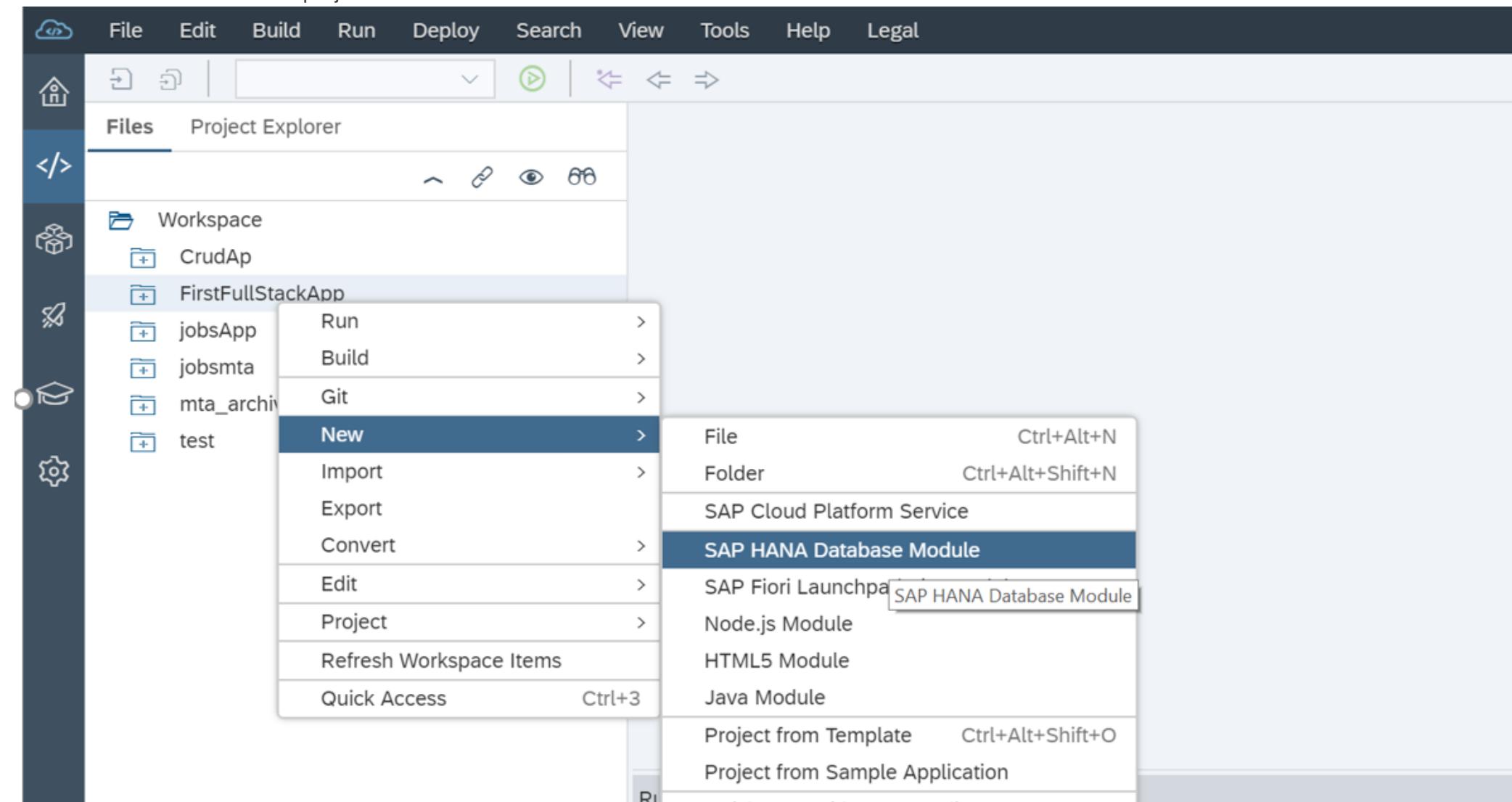
Confirmation

A new project named FirstFullStackApp will be created in your workspace.

[Previous](#)[Finish](#)

It will create a project folder with the mta.yaml file.

Create a DB module inside the project. as shown below.



We are trying to create a DB for storing employee and their role details.

Basic Information

Template Customization

Confirmation

X

New SAP HANA Database Module

Basic Information

Module Name*

Employee

Previous

Next

Next

New SAP HANA Database Module

Template Customization

Namespace

Schema Name

SAP HANA Databas... ▼

Build module after creation

[Previous](#)

[Next](#)

[Finish](#)

[Finish](#)

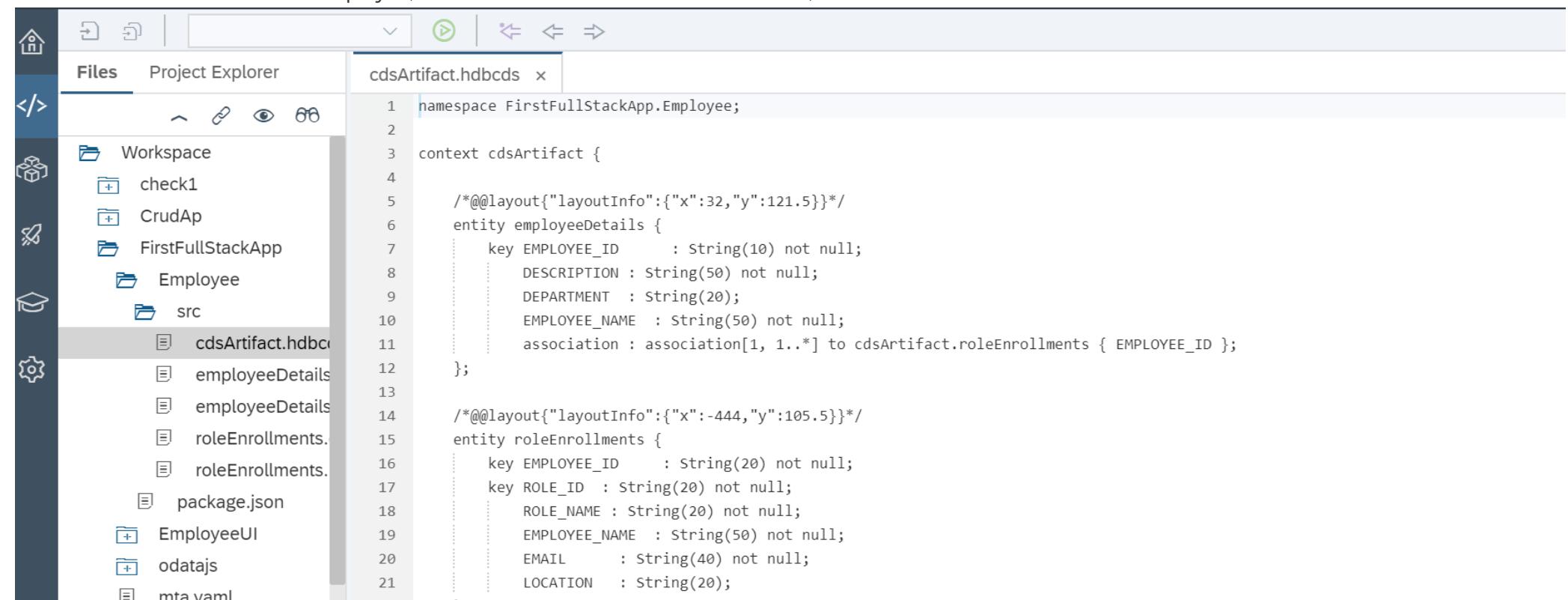
New SAP HANA Database Module

Confirmation

A new module named Employee will be created in your workspace.

[Previous](#)[Finish](#)

This will create a folder named Employee, Now create a new file named cdsAtrifcat, under src as mentioned below.



The screenshot shows the SAP Studio IDE interface. On the left is a dark sidebar with various icons. The main area has a 'Files' tab selected in the top navigation bar. Below it is a 'Project Explorer' view showing a workspace structure. Inside 'FirstFullStackApp' > 'Employee' > 'src', there is a file named 'cdsArtifact.hdbc'. The code editor on the right displays the content of this file:

```
namespace FirstFullStackApp.Employee;

context cdsArtifact {

    /*@@layout{"layoutInfo":{"x":32,"y":121.5}*/
    entity employeeDetails {
        key EMPLOYEE_ID      : String(10) not null;
        DESCRIPTION          : String(50) not null;
        DEPARTMENT           : String(20);
        EMPLOYEE_NAME        : String(50) not null;
        association          : association[1, 1..*] to cdsArtifact.roleEnrollments { EMPLOYEE_ID };
    };

    /*@@layout{"layoutInfo":{"x":-444,"y":105.5}*/
    entity roleEnrollments {
        key EMPLOYEE_ID      : String(20) not null;
        key ROLE_ID           : String(20) not null;
        ROLE_NAME             : String(20) not null;
        EMPLOYEE_NAME         : String(50) not null;
        EMAIL                 : String(40) not null;
        LOCATION              : String(20);
    };
}
```

```
namespace FirstFullStackApp.Employee;

context cdsArtifact {

    /*@@layout{"layoutInfo":{"x":32,"y":121.5}*/
    entity employeeDetails {
        key EMPLOYEE_ID      : String(10) not null;
        DESCRIPTION          : String(50) not null;
        DEPARTMENT           : String(20);
        EMPLOYEE_NAME        : String(50) not null;
        association          : association[1, 1..*] to cdsArtifact.roleEnrollments { EMPLOYEE_ID };
    };

    /*@@layout {"layoutInfo":{"x":-444,"y":105.5}*/
    entity roleEnrollments {
        key EMPLOYEE_ID      : String(20) not null;
    };
}
```

```

key ROLE_ID : String(20) not null;
  ROLE_NAME : String(20) not null;
  EMPLOYEE_NAME : String(50) not null;
  EMAIL      : String(40) not null;
  LOCATION   : String(20);
};

};

```

Create a new file named employeeDetails.hdbtabledata

The screenshot shows the SAP Cloud Platform Studio interface. The left sidebar contains icons for Cloud, File, Edit, Build, Run, Deploy, Search, View, Tools, Help, and Legal. Below these are icons for Home, Project Explorer, Workspace, Check, CrudAp, FirstFullStackApp, Employee, src, cdsArtifact.hdbc, employeeDetails.csv, employeeDetails.hdbtabledata (which is selected), roleEnrollments.csv, roleEnrollments.hdbtabledata, and package.json.

The main area shows the file `employeeDetails.hdbtabledata` with the following JSON content:

```

1  {
2
3  "format_version": 1,
4
5  "imports":
6
7  [ {
8
9  "target_table" : "FirstFullStackApp.Employee::cdsArtifact.employeeDetails",
10
11 "source_data" : { "data_type" : "CSV", "file_name" : "FirstFullStackApp.Employee::employeeDetail",
12
13 "import_settings" : { "import_columns" : ["EMPLOYEE_ID","DESCRIPTION","EMPLOYEE_NAME"] },
14
15 "column_mappings" : {"EMPLOYEE_ID" : 1,"DESCRIPTION" : 2,"EMPLOYEE_NAME" : 3}
16
17 }
18

```

{

"format_version": 1,

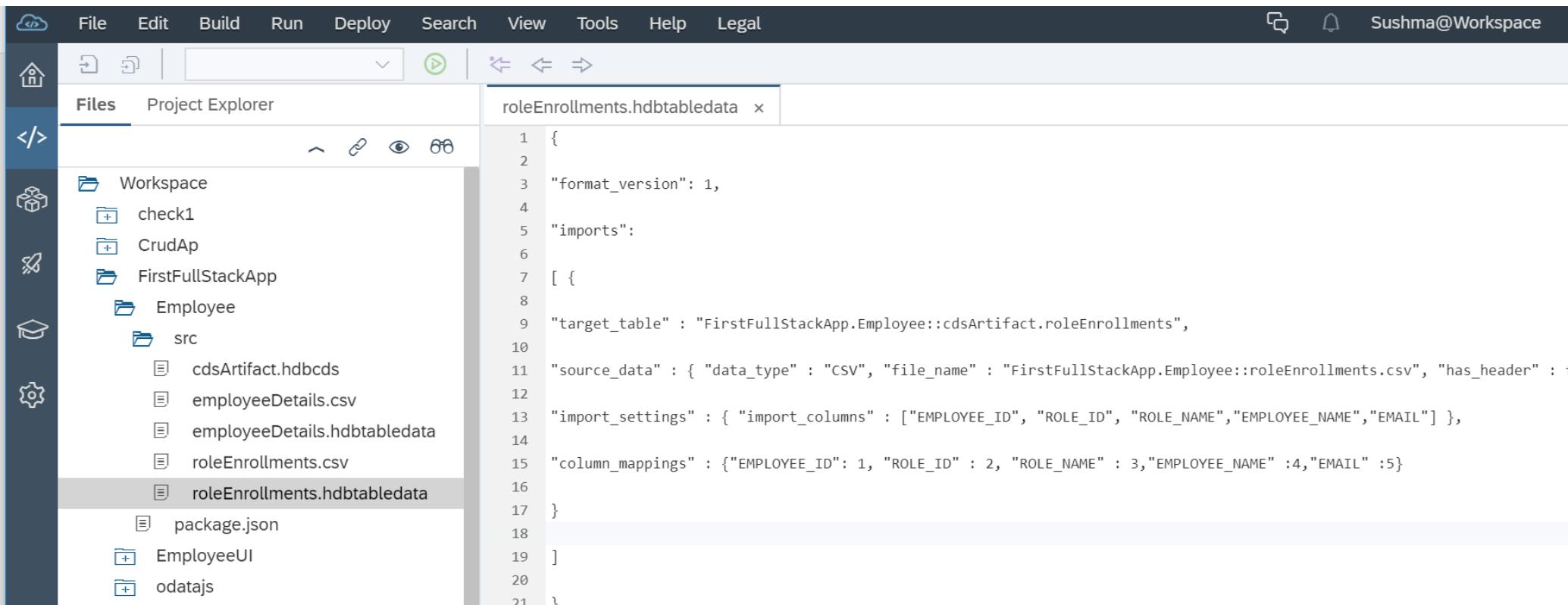
"imports":

```
[ {
```

```
    "target_table" : "FirstFullStackApp.Employee::cdsArtifact.employeeDetails",  
  
    "source_data" : { "data_type" : "CSV", "file_name" : "FirstFullStackApp.Employee::employeeDetails.csv", "has_header" : false },  
  
    "import_settings" : { "import_columns" : [ "EMPLOYEE_ID", "DESCRIPTION", "EMPLOYEE_NAME" ] },  
  
    "column_mappings" : { "EMPLOYEE_ID" : 1, "DESCRIPTION" : 2, "EMPLOYEE_NAME" : 3 }  
  
}
```

```
]  
  
}
```

Create a file named roleEnrollments.hdbtabledata



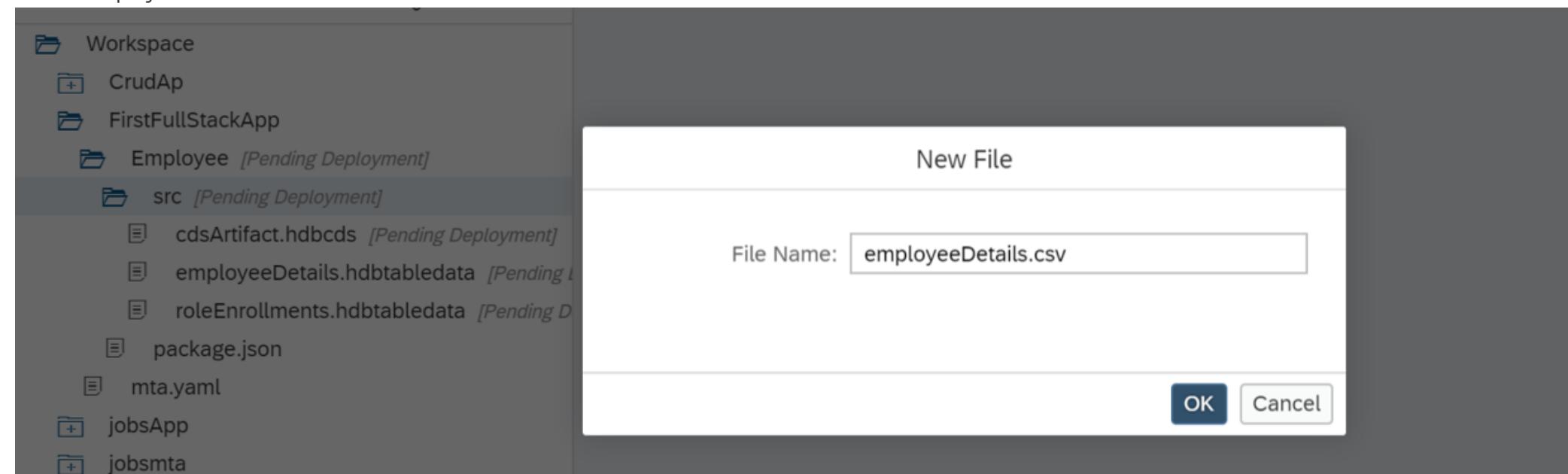
The screenshot shows the SAP Studio interface with the following details:

- Top Bar:** File, Edit, Build, Run, Deploy, Search, View, Tools, Help, Legal. On the right, there are notifications and the user Sushma@Workspace.
- Sidabar:** Home, Project Explorer, Workspace, Employee, src, package.json, EmployeeUI, odatajs.
- Project Explorer:** Shows the file structure: workspace, check1, CrudAp, FirstFullStackApp, Employee, src, cdsArtifact.hdbcids, employeeDetails.csv, employeeDetails.hdbtabledata, roleEnrollments.csv, roleEnrollments.hdbtabledata, package.json, EmployeeUI, odatajs. The file "roleEnrollments.hdbtabledata" is selected and highlighted in grey.
- Code Editor:** The content of the selected file is displayed:

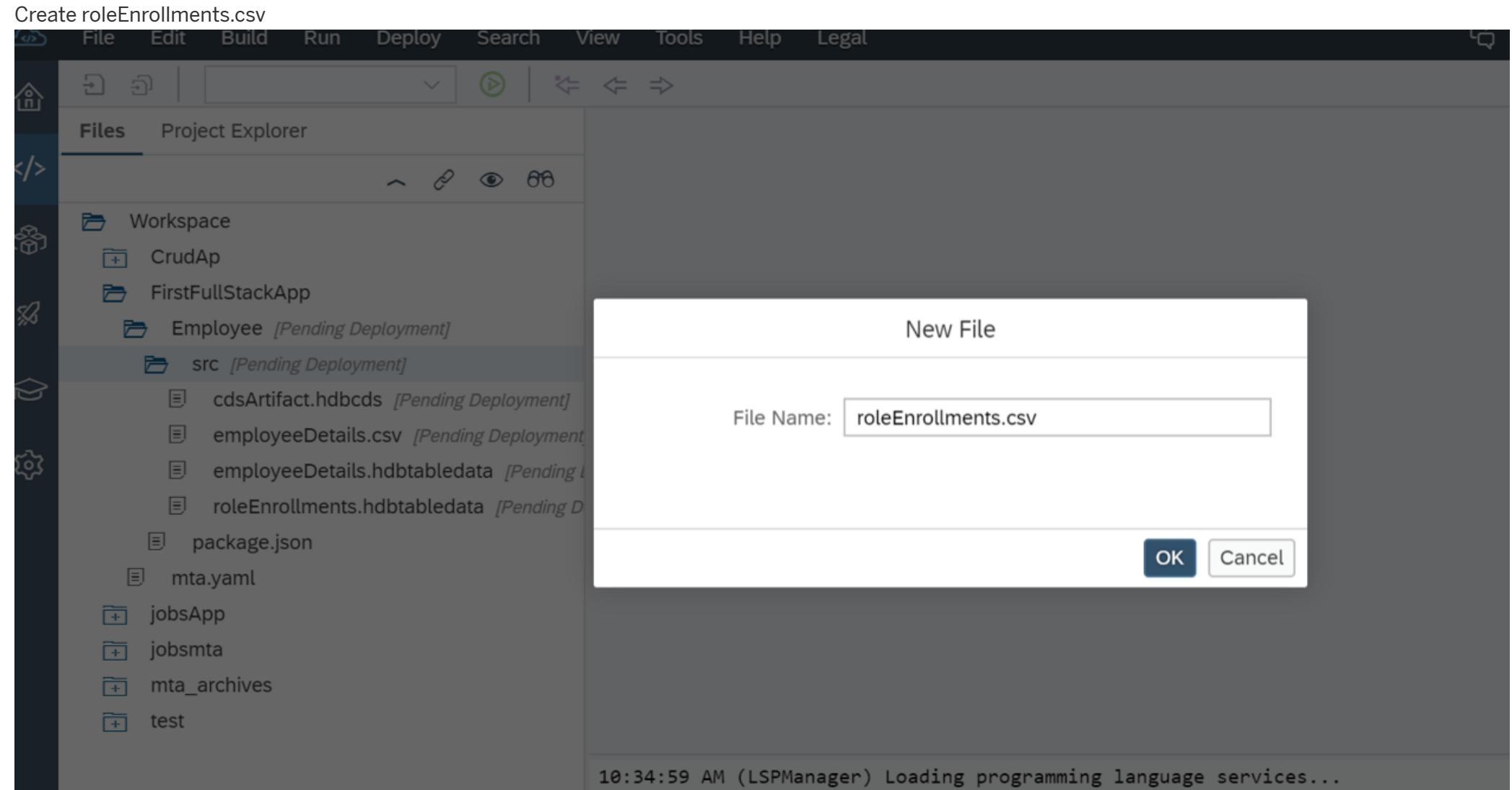
```
1 {  
2     "format_version": 1,  
3     "imports":  
4     [ {  
5         "target_table" : "FirstFullStackApp.Employee::cdsArtifact.roleEnrollments",  
6         "source_data" : { "data_type" : "CSV", "file_name" : "FirstFullStackApp.Employee::roleEnrollments.csv", "has_header" : false },  
7         "import_settings" : { "import_columns" : [ "EMPLOYEE_ID", "ROLE_ID", "ROLE_NAME", "EMPLOYEE_NAME", "EMAIL" ] },  
8         "column_mappings" : { "EMPLOYEE_ID": 1, "ROLE_ID" : 2, "ROLE_NAME" : 3, "EMPLOYEE_NAME" : 4, "EMAIL" : 5 }  
9     }  
10    ]  
11}  
12}
```

```
{  
  
  "format_version": 1,  
  
  "imports":  
  
  [ {  
  
    "target_table" : "FirstFullStackApp.Employee::cdsArtifact.roleEnrollments",  
  
    "source_data" : { "data_type" : "CSV", "file_name" : "FirstFullStackApp.Employee::roleEnrollments.csv", "has_header" : false },  
  
    "import_settings" : { "import_columns" : ["EMPLOYEE_ID", "ROLE_ID", "ROLE_NAME","EMPLOYEE_NAME","EMAIL"] },  
  
    "column_mappings" : {"EMPLOYEE_ID": 1, "ROLE_ID" : 2, "ROLE_NAME" : 3,"EMPLOYEE_NAME" :4,"EMAIL" :5}  
  
  }  
  
 ]  
  
 }
```

Now for above-created tables, we will create data by using CSV file as shown below under the src folder.



```
"E100","desc1","Arun"  
"E101","desc2","Anand"  
"E102","desc3","Ram"  
"E103","desc4","Ananya"
```



```
"E100","J5136","JavaCodes","Arun","arun.rage@sap.com"  
"E101","J5137","BusinessAnaylist","Anand","anand.kista@sap.com"  
"E102","J5138","Designer","Ram","ram.raja@sap.com"  
"E103","J5139","Fresher","Ananya","ananya.sudheer@sap.com"
```

Folder structure will look like this

FirstFullStackApp

Employee [Pending Deployment]

src [Pending Deployment]

 cdsArtifact.hdbcds [Pending Deployment]

 employeeDetails.csv [Pending Deployment]

 employeeDetails.hdbtabledata [Pending Deployment]

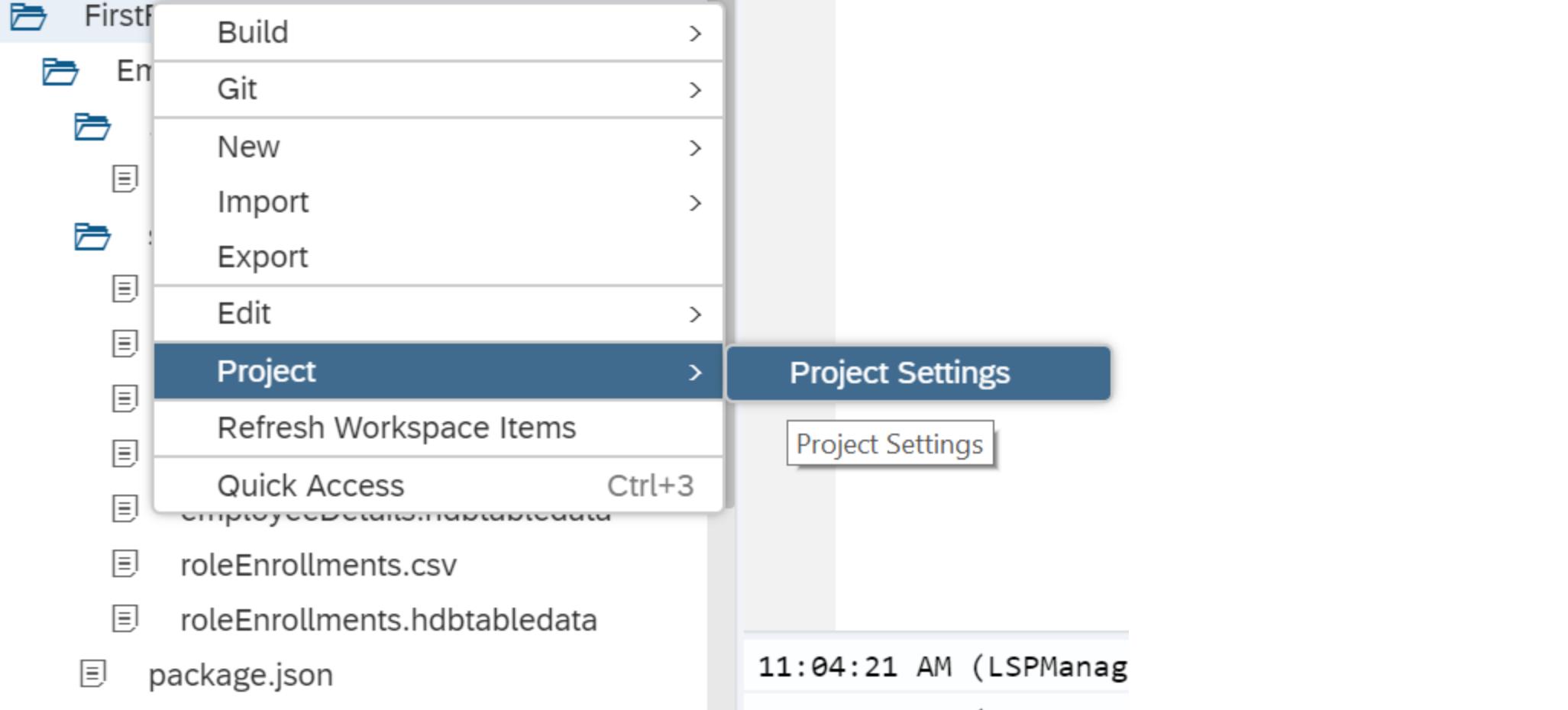
 roleEnrollments.csv [Pending Deployment]

 roleEnrollments.hdbtabledata [Pending Deployment]

 package.json

 mta.yaml

Right click on Project (FirstFullStackApp) and navigate to project setting to install cloud Foundry builder as shown below. This will help us to build and deploy our DB module to Cloud Foundry further.



Project

Cloud Foundry

Project Types

Beautifier

JavaScript

Code Checking

JavaScript

XML

General

Languages

Cloud Foundry
Configure the default Cloud Foundry space for running your projects. [Learn More](#)

Cloud Foundry Settings

Use the default settings defined in the Workspace Preferences.
API Endpoint: <https://api.cf.eu10.hana.ondemand.com>
Organization: I505386trial_trial
Space: dev

Use the following settings for this project.

* API Endpoint:

Organization:

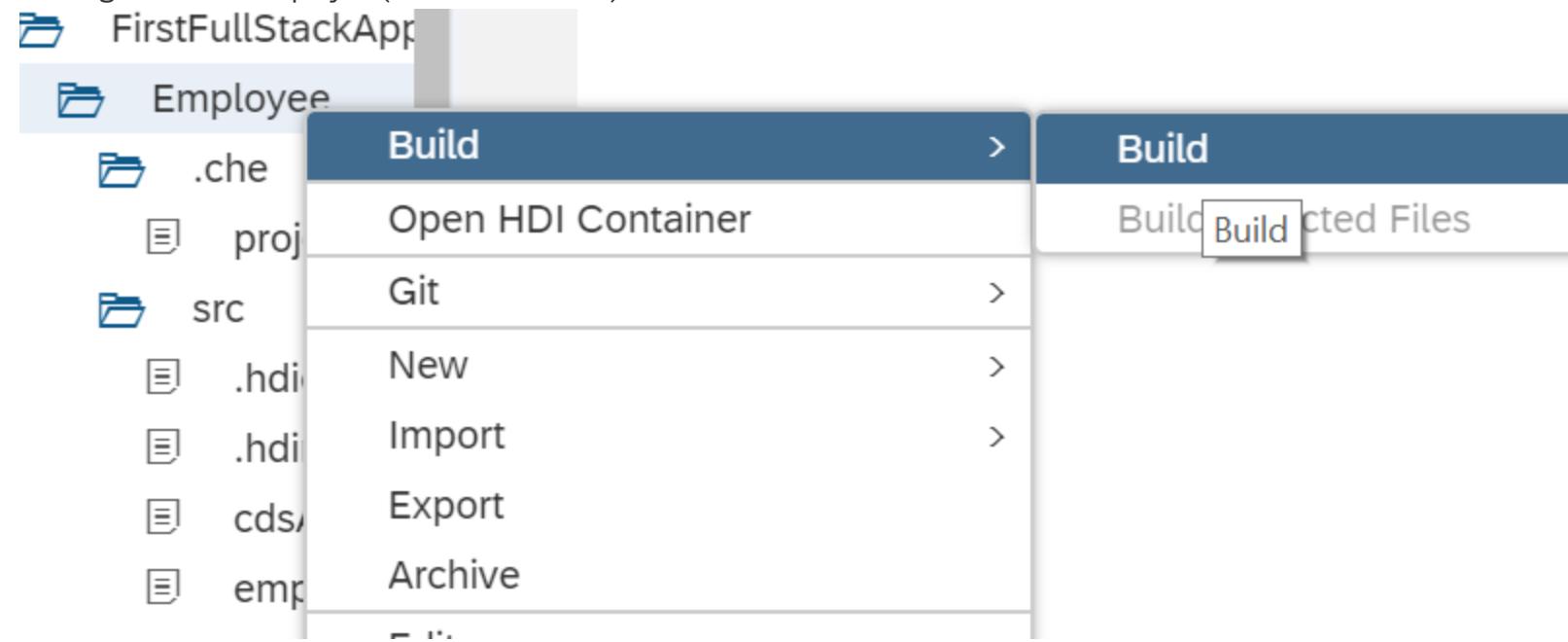
Space:

Click on "install builder" button

Note: If you find default settings are not present or not as per your cloud Foundry trial account in the cockpit, then from your trail account space you can get the required API endpoint details and paste the same here and click on build installer button.

Once the builder is installed successfully click on the save button.

Now right click on Employee (DB module folder) and build it as shown below.



Once the build is successful, In cockpit->cloud foundry-> space check for the deployed DB module

The screenshot shows the SAP Cloud Foundry cockpit interface. On the left, there's a sidebar with various navigation options like Portal, Routes, Security Groups, Events, and Members. The main area is titled "Deploy Application" and shows a table of service instances. One instance is listed with the name "webide-builder-di-sapwebide-EU-1-trial-sifd0kqc1rwh80jN", status "Started", 1/1 instance, 4096 MB disk quota, and 2048 MB memory.

Requested State	Name	Instances	Disk Quota	Memory	Actions
Started	webide-builder-di-sapwebide-EU-1-trial-sifd0kqc1rwh80jN	1/1	4096 MB	2048 MB	

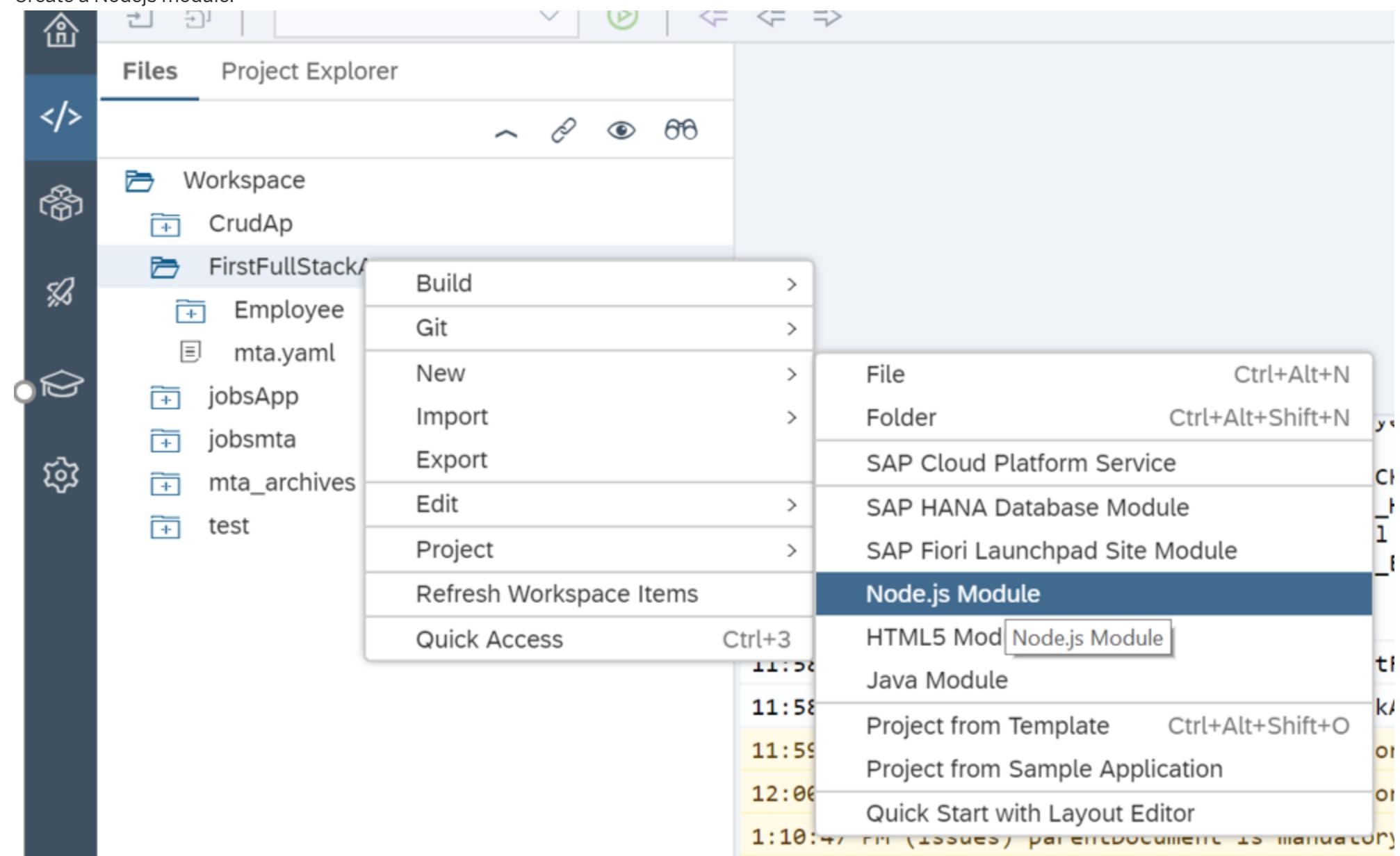
Check for instance created.

The screenshot shows the SAP Cloud Foundry cockpit interface. The left sidebar has "Service Instances" selected. The main area is titled "Space: dev - Service Instances" and shows a table of service instances. One instance is listed with the name "FirstFullStackAp...", service "hanatrial", plan "hdi-shared", referencing applications "jBt4n0jMvkpz4MtDl StackApp-odatajs", service keys "2", and last operation "Created".

Name	Service	Plan	Referencing Applications	Service Keys	Last Operation	Actions
FirstFullStackAp...	hanatrial	hdi-shared	jBt4n0jMvkpz4MtDl StackApp-odatajs	2	Created	

Now your HDI container type DB is ready in Cloud Foundry. We can access it in another module if required by using data service.

Below are the steps to create OData service.



New Node.js Module

Basic Information

Module Name*

[Previous](#)[Next](#)

1:10:47 PM (issues) parentDocument is mandatory parameter for getIssues

Project Explorer Basic Information Template Customization Confirmation X

New Node.js Module

Template Customization

Module settings:

Version: * 1.0.0

Description: An optional module description

Main JS file: * server.js

Enable XSJS support

Enables the XSJS support by adding Node.js dependencies to required libraries

Previous Next Finish

```
1:10:47 PM (issues) parentDocument is mandatory parameter for getIssues
```

New Node.js Module

Confirmation

A new module named odatajs will be created in your workspace.

[Previous](#)[Finish](#)[Finish](#)

1:10:47 PM (issues) parentDocument is mandatory parameter for getIssues

The screenshot shows the SAP Studio interface with the following details:

- Top Bar:** File, Edit, Build, Run, Deploy, Search, View, Tools, Help, Legal.
- Sidemenu:** Home, Files (selected), Project Explorer, Refresh, Filter, View, Help.
- Project Explorer:** Workspace (expanded) containing:
 - CrudAp
 - FirstFullStackApp
 - Employee
 - odatajs
 - mta.yaml
 - jobsApp
 - jobsmta
 - mta_archive
 - test
- Context Menu (Open at Employee node):** Run, Build, Git, New, Import, Export, Archive, Edit, Project, Refresh Workspace Items, Quick Access (Ctrl+3).
- Submenu for New:** File (selected), Folder (Ctrl+Alt+N), Project from Template, Project from Sample Application, Quick Start with Layout Editor.
- File Status:** MAKE SUCCEEDED (0 Warnings), 2 Files updated, 1 File changed.
- Log Output:** Making... ok, Starting make in the container "FIRSTFULLSTACKAPP", 59:06 AM (issues) parentDocument is mandatory, 00:46 PM (issues) parentDocument is mandatory, 10:47 PM (issues) parentDocument is mandatory, 1:27:30 PM (issues) parentDocument is mandatory.

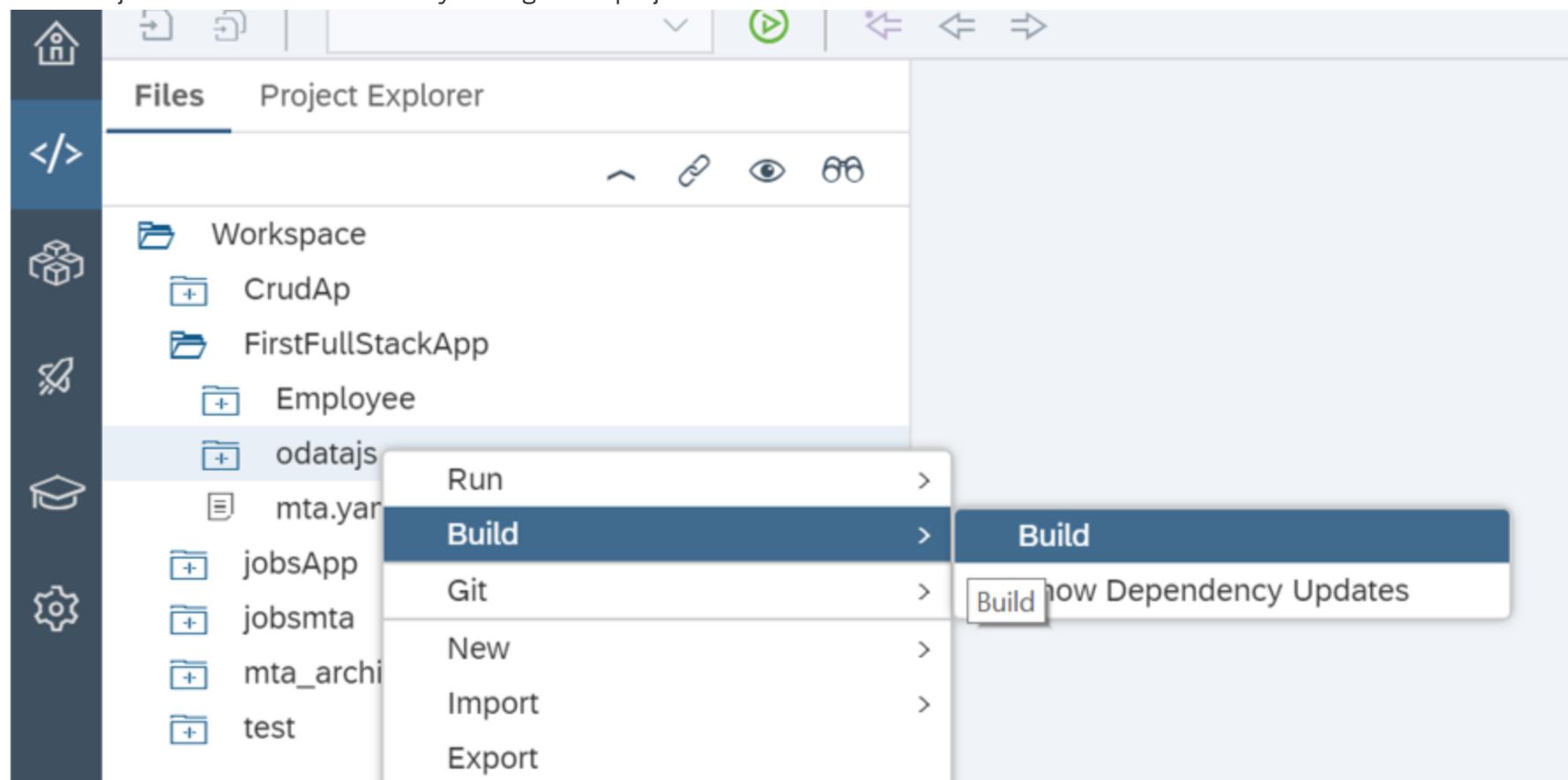
The screenshot shows the SAP Studio interface. The left sidebar contains icons for Home, Back, Forward, Run script start, Workspace, Check1, CrudAp, FirstFullStackApp (with Employee, EmployeeUI, odatajs, lib, xsodata, index.xsjs, node_modules, test, package.json), and Settings. The top bar includes icons for file operations and navigation. The Project Explorer tab is selected, showing the 'Files' section with 'Project Explorer'. The main area displays the content of 'service.xsodata'.

```
1 service
2 {
3
4 "FirstFullStackApp.Employee::cdsArtifact.employeeDetails" as "employeeDetails" navigates ("role_Enrollments" as "roleEnrollments");
5 "FirstFullStackApp.Employee::cdsArtifact.roleEnrollments" as "roleEnrollments" ;
6 association "role_Enrollments" principal "employeeDetails"("EMPLOYEE_ID")
7 multiplicity "1" dependent "roleEnrollments"("EMPLOYEE_ID") multiplicity "*";
8
9
10 }
```

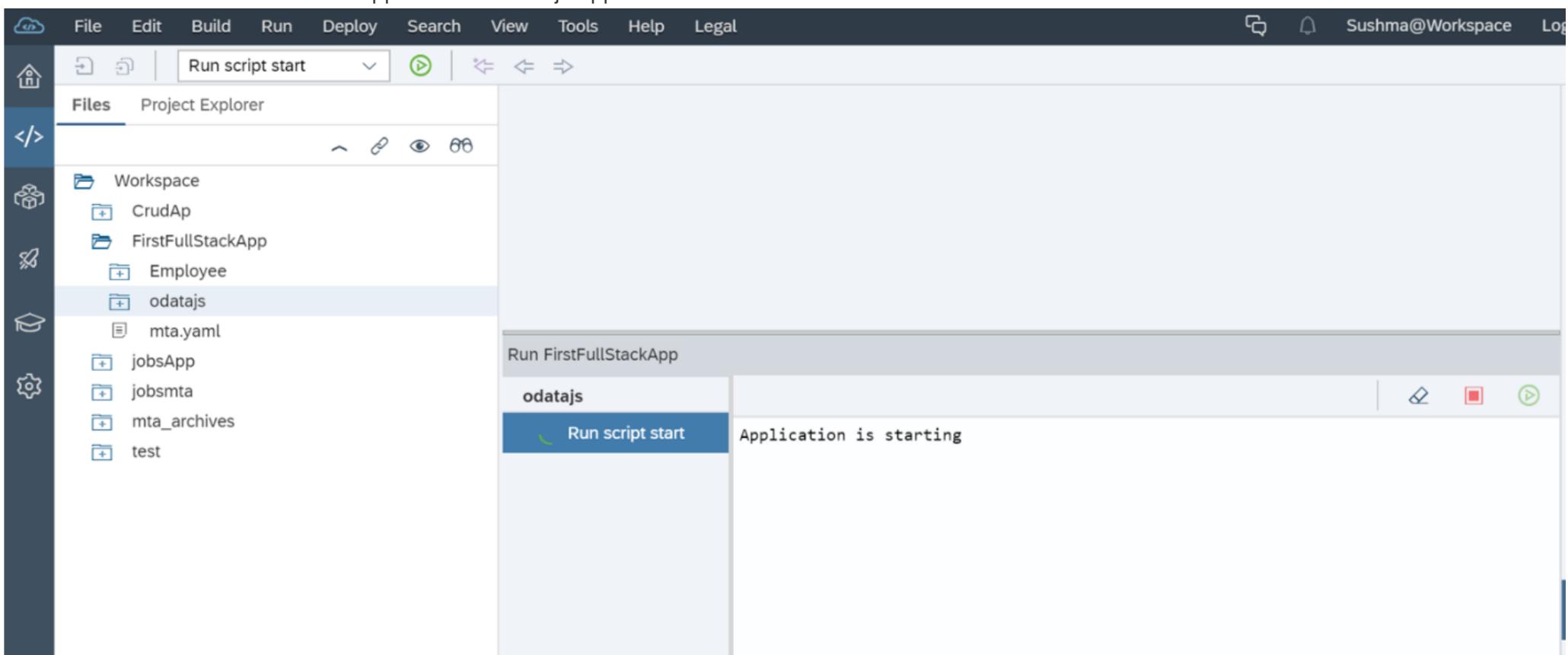
```
service
{
  "FirstFullStackApp.Employee::cdsArtifact.employeeDetails" as "employeeDetails" navigates ("role_Enrollments" as "roleEnrollments");
  "FirstFullStackApp.Employee::cdsArtifact.roleEnrollments" as "roleEnrollments" ;
  association "role_Enrollments" principal "employeeDetails"("EMPLOYEE_ID")
  multiplicity "1" dependent "roleEnrollments"("EMPLOYEE_ID") multiplicity "*";
}
```

The screenshot shows the SAP Cloud Platform Studio interface. On the left, there's a vertical toolbar with icons for home, workspace, crud, first full stack app, employee, odatajs, mta.yaml, jobsApp, jobsmta, mta_archives, and test. The 'Files' tab is selected in the top navigation bar. The Project Explorer shows a workspace with subfolders like CrudAp, FirstFullStackApp, Employee, odatajs, and mta.yaml. The mta.yaml file is currently open in the main editor area. The code in the editor is as follows:

```
path: Employee
requires:
- name: hdi_Employee
30
- name: odatajs
type: nodejs
path: odatajs
34 requires:
- name: hdi_Employee
provides:
- name: odatajs_api
properties:
url: ${default-url}
40
resources:
- name: rt_FirstFullstackApp_appRouter
parameters:
service-plan: app-runtime
service: html5-apps-repo
```



Once build is successful then Run the application as Nodejs Application.



Once the application started successfully, we can check the metadata.

replace index.js in the URL from the browser it opened after running successfully, with /xsodata/service.xsodata/\$metadata.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="2.0">
    <Schema xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
      xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="default">
      <EntityType Name="employeeDetailsType">
        <Key>
          <PropertyRef Name="EMPLOYEE_ID"/>
        </Key>
        <Property Name="EMPLOYEE_ID" Type="Edm.String" Nullable="false" MaxLength="10"/>
        <Property Name="DESCRIPTION" Type="Edm.String" Nullable="false" MaxLength="50"/>
        <Property Name="DEPARTMENT" Type="Edm.String" MaxLength="20"/>
        <Property Name="EMPLOYEE_NAME" Type="Edm.String" Nullable="false" MaxLength="50"/>
        <Property Name="association.EMPLOYEE_ID" Type="Edm.String" MaxLength="20"/>
        <NavigationProperty Name="roleEnrollments" Relationship="default.role_EnrollmentsType" FromRole="employeeDetailsPrincipal" ToRole="roleEnrollmentsDependent"/>
      </EntityType>
      <EntityType Name="roleEnrollmentsType">
        <Key>
          <PropertyRef Name="EMPLOYEE_ID"/>
          <PropertyRef Name="ROLE_ID"/>
        </Key>
        <Property Name="EMPLOYEE_ID" Type="Edm.String" Nullable="false" MaxLength="20"/>
        <Property Name="ROLE_ID" Type="Edm.String" Nullable="false" MaxLength="20"/>
        <Property Name="ROLE_NAME" Type="Edm.String" Nullable="false" MaxLength="20"/>
        <Property Name="EMPLOYEE_NAME" Type="Edm.String" Nullable="false" MaxLength="50"/>
        <Property Name="EMAIL" Type="Edm.String" Nullable="false" MaxLength="40"/>
        <Property Name="LOCATION" Type="Edm.String" MaxLength="20"/>
      </EntityType>
      <Association Name="role_EnrollmentsType">
        <End Type="default.employeeDetailsType" Role="employeeDetailsPrincipal" Multiplicity="1"/>
        <End Type="default.roleEnrollmentsType" Role="roleEnrollmentsDependent" Multiplicity="*"/>
      </Association>
      <EntityContainer Name="v2" m:IsDefaultEntityContainer="true">
        <EntitySet Name="employeeDetails" EntityType="default.employeeDetailsType"/>
        <EntitySet Name="roleEnrollments" EntityType="default.roleEnrollmentsType"/>
      </EntityContainer>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

By using this URL we can create "Destination" in cloud Foundry which can be used further in other modules.

The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar has a dark theme with various icons and labels: Applications, Services (selected), Service Marketplace, Service Instances (selected), User-Provided Services, Portal, Routes, Security Groups, Events, and Members. The main content area is titled "Space: dev - Service Instances" and shows "All: 5". It includes a search bar with "All Services" dropdown and a "Search" input field. A table lists service instances with columns: Name, Service, Plan, Referencing Applications, Service Keys, Last Operation, and Actions. Two rows are visible:

Name	Service	Plan	Referencing Applications	Service Keys	Last Operation	Actions
employeeodata	destination	lite	jBt4n0jMvkpz4MtDl StackApp-odatajs	0	Created	
FirstFullStackAp...	hanatrial	hdi-shared	jBt4n0jMvkpz4MtDl StackApp-odatajs webide-builder-di-sapwebide-EU-1-trial-sifd0kqc1rwh80jN	2	Created	

if you are not willing to use OData service then you can skip the step of the creation of Odata module itself and directly navigate to " Accessing DB explorer " and check for tables as shown below.

Accessing DB from SAP WEB IDE.

Quick Access

Ctrl+3

- Home
- Development
- Storyboard
- Database Explorer
- Learning Center

Preferences

Ctrl+.
Extensi

Preferences (Ctrl+.)

Ctrl+Shift+E

- Extensibility Pane with Mock Data

- Workspace Manager

- SAP Cloud Platform Cockpit

- UI Theme Designer

- Create Data Source

Global Preferences

- Code Check
- Code Editor
- Core Data Services
- Data Preview
- Database Explorer
- Default Editors
- Git Committer
- Keyboard Shortcuts
- MDX Console
- SQL Code Completion

Extensions

Enable the SAP Web IDE extensions you want, save, and refresh the browser.



SAP HANA Database Explorer

ON

Use the SAP HANA database explorer to query information about the database, as well as view...
[More Information](#)



SQL Code Completion

SQL Console

SQLScript Debugger

Workspace Preferences

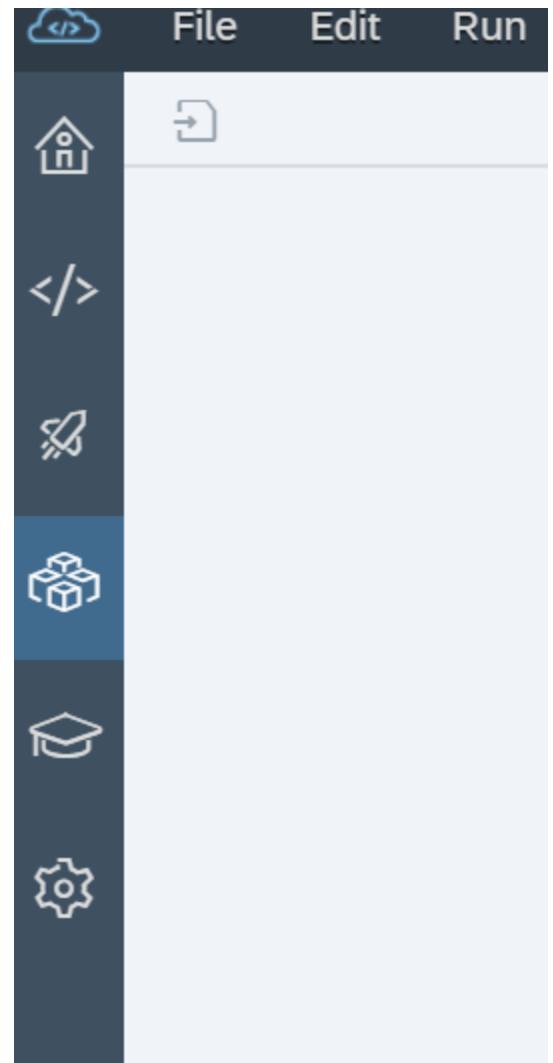
Cloud Foundry

Extensions



SAP SE

This will add DB explorer to you SAP Web IDE. Now you can see DB icon on the left side of the screen.



Click on the icon-> then click on “+” symbol where can add your DB.

Add Database

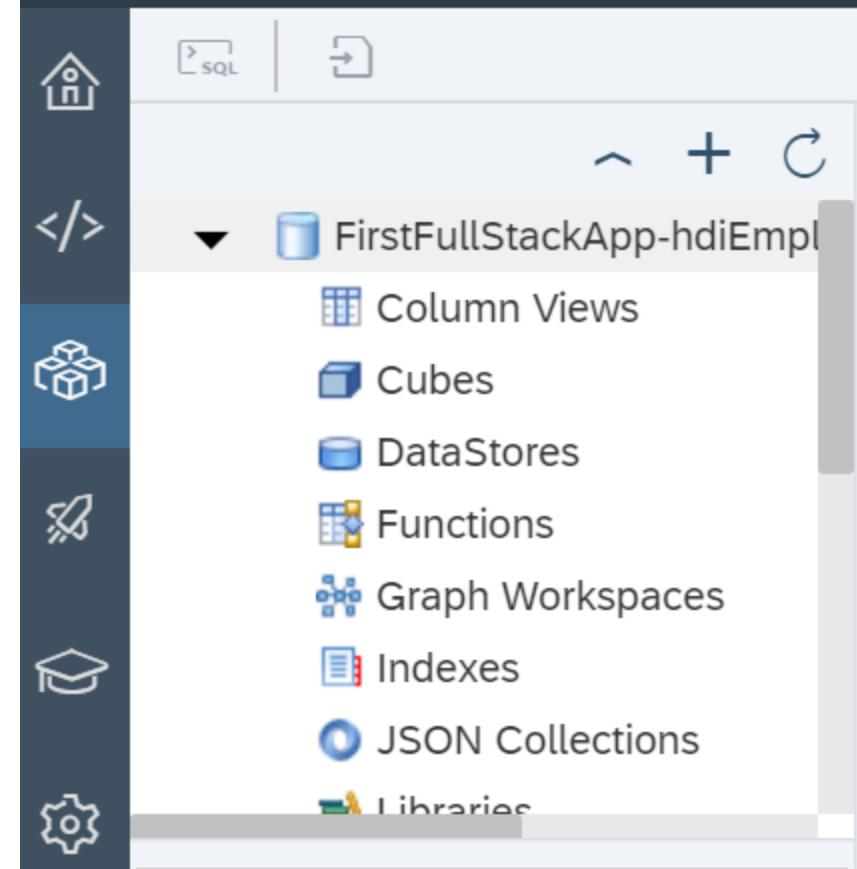
Database Type **HDI Container**

HDI Containers:

Search		Org	Space
Name			
FirstFullStackApp-hdiEmployee-l505tl0+uplhcrSof17v		I505386trial_trial	dev

OK **Cancel**

Select the container and click on OK.



On click of tables in this list will list down all the tables available. Right click on the respective table and open data will display the data present inside tables.

The screenshot shows the SAP HANA Studio interface with the 'Raw Data' tab selected. On the left, there is a sidebar with icons for Remote Subscriptions, Sequences, Synonyms, Table Types, Tables (which is selected), Tasks, Triggers, and Views. Below this is a search bar for 'Search Tables' with a dropdown showing 'FirstFullStackApp.Employee::cdsArtifact.employeeDetails' and 'FirstFullStackApp.Employee::cdsArtifact.roleEnrollments'. The main area displays a table titled 'Rows (4)' with columns: EMPLOYEE_ID, DESCRIPTION, DEPARTMENT, EMPLOYEE_NAME, and association.EMPLOYEE. The data is as follows:

Top right corner “SQL” image “on click” will open an editor where you can execute SQL queries.

Congratulations!!, Now your HDI type DB is ready in Cloud Foundry. You can use this further in UI or any other modules with the help of Odata service if required.

Alert Moderator

Assigned tags

SAP Cloud Platform for the Cloud Foundry environment | Cloud | OData | SAP HANA | cloud foundry |

[View more...](#)

Related Blog Posts

[Deploying any UI5 application on Cloud Foundry](#)

By [Gopal Anand](#), Jan 04, 2019

[Instant realtime GraphQL engine on Cloud Foundry \(with PostgreSQL and Docker\)](#)

By [Pierre Dominique](#), Nov 06, 2019

[Develop, Deploy & Debug a Hana Node.js app \(xsjs\) using Hana DBaaS on cloud foundry.](#)

By [AKSHAY NAYAK](#), Aug 15, 2018

Related Questions

[Issues with cloud foundry trial and Add Container. No HDI containers available.](#)

By [bill steiner](#), Jun 10, 2019

[SAP Hana Service in SAP Cloud Platform using SAP Webide MultiTarget project](#)

By [DurgaPrasanth vemula](#), May 15, 2019

[SAP Fiori launchpad configuration issue of full stack app deployed in cloud foundry](#)

2 Comments

You must be [Logged on](#) to comment or reply to a post.



DurgaPrasanth vemula

May 15, 2019 at 2:22 pm

Hi,

In SAP Cloud Foundry i have two services i.e SAP HANA Schemas & HDI Container and SAP Hana Service as per your blog you are using the service SAP HANA Schemas & HDI Container but my question if already i have table and data in SAP Hana Service and how to get those data in Webide by creating Multi Target project.

In the Documentation of the Service SAP HANA schemas & HDI containers it is mentioned that The SAP HANA schemas & HDI containers service is part of the SAP HANA service.

But my question is that i already have table and data in SAP Hana Service and how to get those data in Webide by creating Multi Target project and we can data from the table in the SAP HANA service through SAP HANA schemas & HDI containers?

Like (0)



Sushma Kudum | Post author

May 17, 2019 at 4:25 am

Hi Durga,

FYR,

<https://developers.sap.com/tutorials/haas-dm-create-db-mta.html>

Regards,

Sushma

Like (0)

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support