Developer Guide

This guide helps you to start development in SAP Business Application Studio. SAP Business Application Studio is a service within SAP Business Technology Platform that provides a development environment for SAP Cloud Foundry. This is an important tool for any kind of development with SAP HANA Cloud, SAP HANA database, with PostgreSQL database.

SAP Business Application Studio provides a desktop-like experience similar to leading IDEs, with command line and optimized editors. This simplifies and saves time in setting up your development environment and allows you to efficiently develop, test, build, and run your solution locally or in the cloud.

You will learn

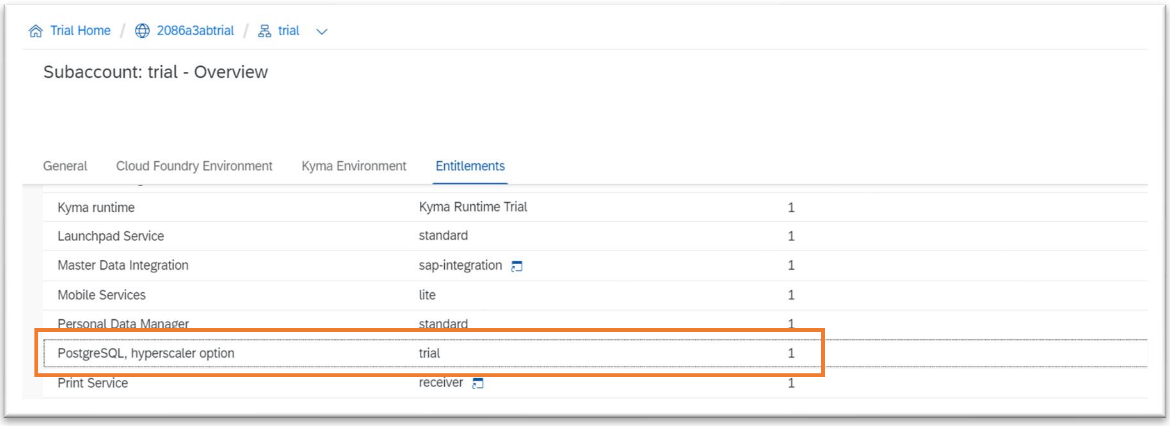
- How to set up SAP Business Application Studio
- How to create an application with the wizard
- How to deploy and test simple REST API application in the cloud
- How to deploy and test REST API application in SAP Business Application Studio
- How to see logs provided by the application

Prerequisites

If you plan to follow the development steps and deploy the project by yourself, you have to do some preparations:

1. Get a free account on SAP BTP Trial
   Follow the steps from this tutorial: https://developers.sap.com/tutorials/hcp-create-trial-account.html

2. Ensure, that the *PostgreSQL, hyperscaler option* is in the list of entitlements of your trial account. If not, then you need to add the service manually.

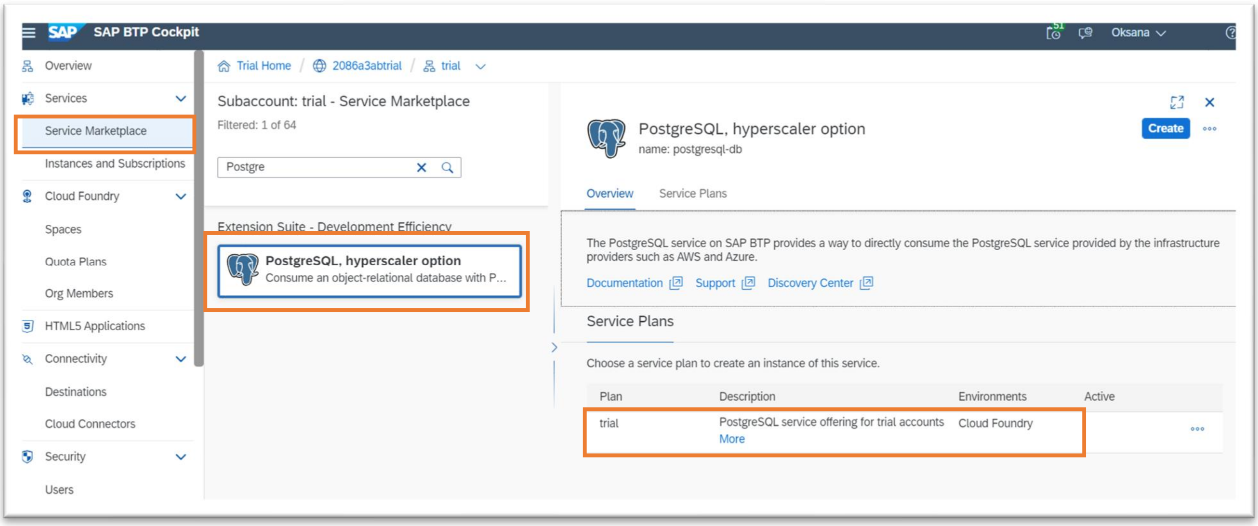


Subaccount: trial - Overview

Kyma runtime	Kyma Runtime Trial	1
Launchpad Service	standard	1
Master Data Integration	sap-integration	1
Mobile Services	lite	1
Personal Data Manager	standard	1
PostgreSQL, hyperscaler option	trial	1
Print Service	receiver	1

3. Ensure, that you have *PostgreSQL, hyperscaler* option in Service Marketplace.

Please NOTE: “trial” service plan will be used for the current example



Subaccount: trial - Service Marketplace

Filtered: 1 of 64

PostgreSQL, hyperscaler option
Consume an object-relational database with P...

PostgreSQL, hyperscaler option
name: postgresql-db

The PostgreSQL service on SAP BTP provides a way to directly consume the PostgreSQL service provided by the infrastructure providers such as AWS and Azure.

Service Plans

Choose a service plan to create an instance of this service.

Plan	Description	Environments	Active
trial	PostgreSQL service offering for trial accounts More	Cloud Foundry	

4. Set up SAP Business Application Studio for development

Follow the steps from this tutorial: https://developers.sap.com/tutorials/appstudio-onboarding.html

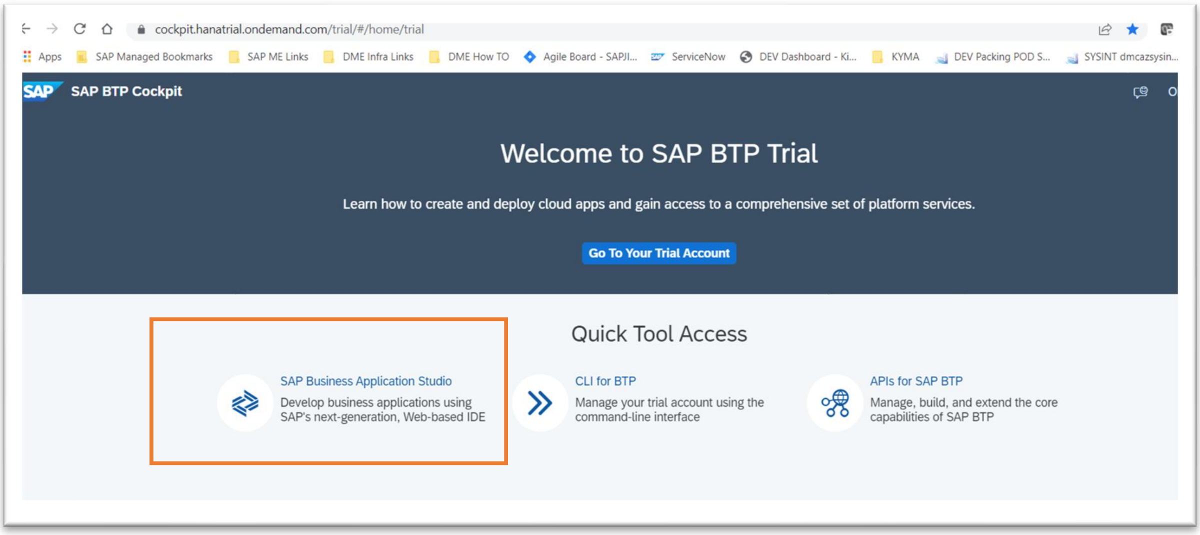
Development Steps

Step 1: Create dev space in SAP Business Application Studio

Dev spaces are like isolated virtual machines in the cloud that can be quickly spun-up. Each dev space type contains tailored tools and pre-installed run times for a target scenario such as SAP Fiori or mobile development. This simplifies and saves time in setting up the development environment as there’s no need to install anything or upgrade; letting developers focus on their business domain, anytime, anywhere.

In this step we will create a new Dev Space configured for SAP Cloud Foundry development. If you already have a dev space configured for Cloud Foundry development, you can skip this step as each dev space can hold multiple projects.

1. Open SAP Business Application Studio.



Welcome to SAP BTP Trial

Learn how to create and deploy cloud apps and gain access to a comprehensive set of platform services.

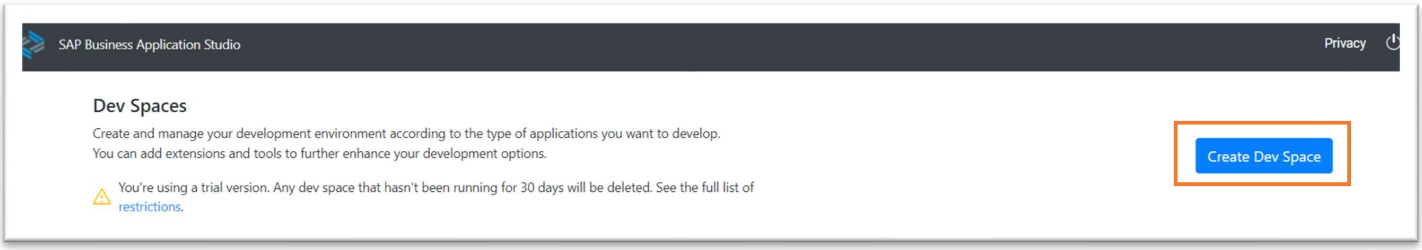
Quick Tool Access

SAP Business Application Studio
Develop business applications using SAP's next-generation, Web-based IDE

CLI for BTP
Manage your trial account using the command-line interface

APIs for SAP BTP
Manage, build, and extend the core capabilities of SAP BTP

2. Choose Create Dev Space.



SAP Business Application Studio

Privacy

Dev Spaces

Create and manage your development environment according to the type of applications you want to develop.
You can add extensions and tools to further enhance your development options.

⚠ You're using a trial version. Any dev space that hasn't been running for 30 days will be deleted. See the full list of restrictions.

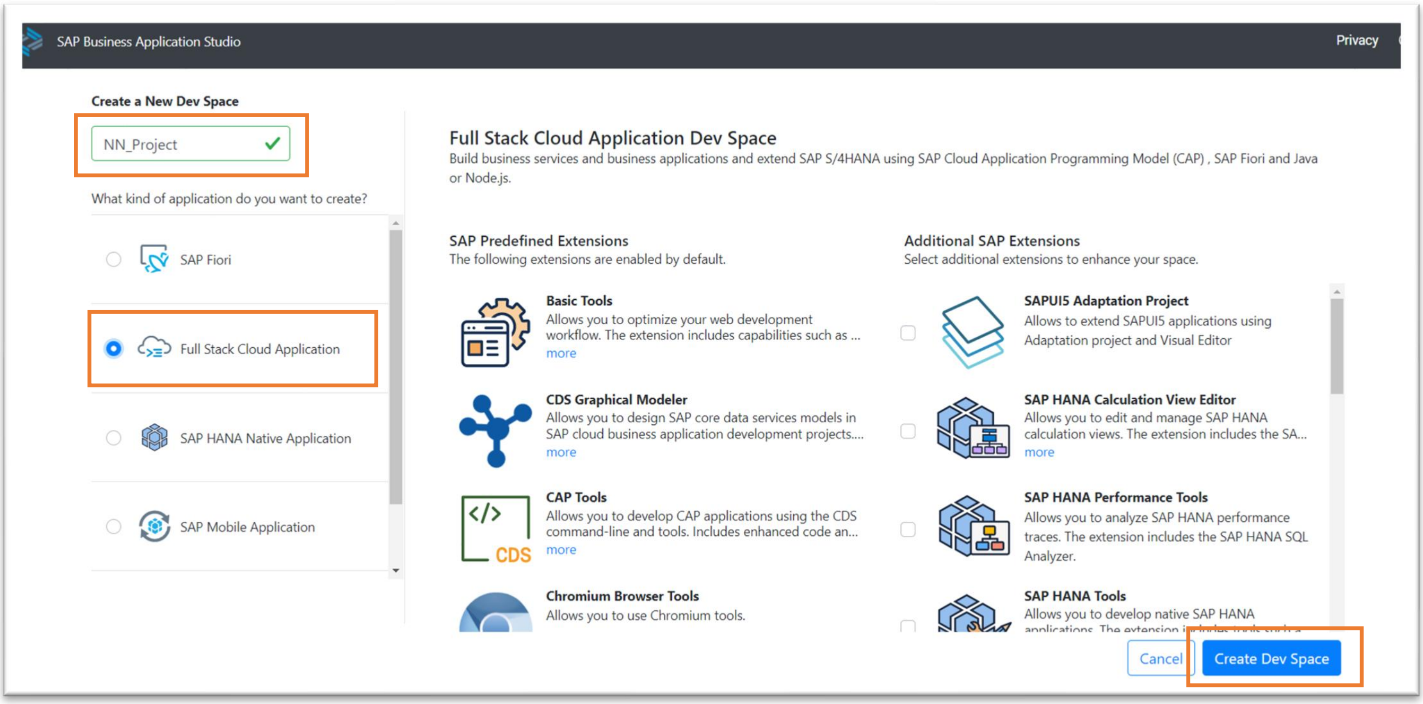
Create Dev Space

3. At the “Create New Dev Space” screen execute the following steps:
   - Enter the NN_Project name for your dev space.
   - Choose Full Stack Cloud Application as the application type.

   By selecting Full Stack Cloud Application your dev space comes with several extensions out-of-the-box that you need to develop applications.

   - Choose Create Dev Space.

   The Dev Space will then begin starting and the process will take a minute or so as your cloud environment is being created. You see that the status for your dev space will change from STARTING to RUNNING.



SAP Business Application Studio

Privacy

Create a New Dev Space

NN_Project ✓

What kind of application do you want to create?

○ SAP Fiori

● Full Stack Cloud Application

○ SAP HANA Native Application

○ SAP Mobile Application

Full Stack Cloud Application Dev Space

Build business services and business applications and extend SAP S/4HANA using SAP Cloud Application Programming Model (CAP) , SAP Fiori and Java or Node.js.

SAP Predefined Extensions

The following extensions are enabled by default.

Basic Tools
Allows you to optimize your web development workflow. The extension includes capabilities such as ...
more

CDS Graphical Modeler
Allows you to design SAP core data services models in SAP cloud business application development projects....
more

CAP Tools
Allows you to develop CAP applications using the CDS command-line and tools. Includes enhanced code an...
more

Chromium Browser Tools
Allows you to use Chromium tools.

Additional SAP Extensions

Select additional extensions to enhance your space.

□ SAPUI5 Adaptation Project
Allows to extend SAPUI5 applications using Adaptation project and Visual Editor

□ SAP HANA Calculation View Editor
Allows you to edit and manage SAP HANA calculation views. The extension includes the SA...
more

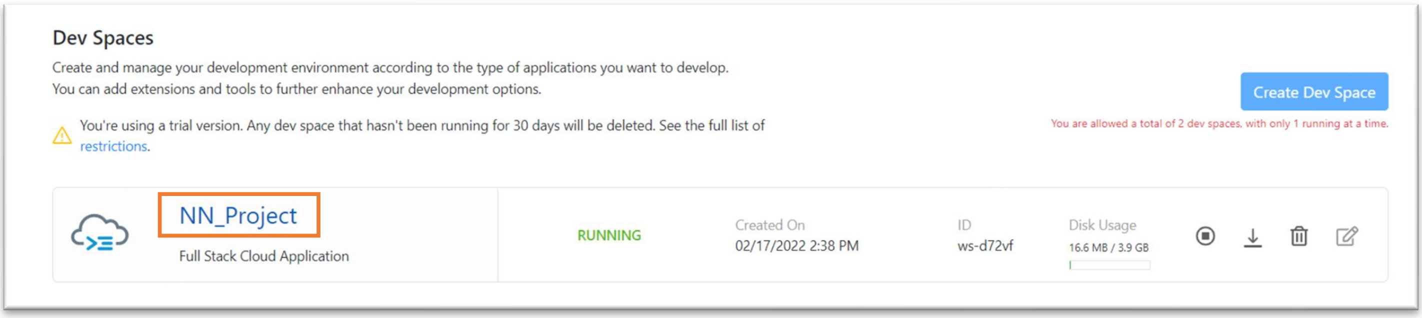
□ SAP HANA Performance Tools
Allows you to analyze SAP HANA performance traces. The extension includes the SAP HANA SQL Analyzer.

□ SAP HANA Tools
Allows you to develop native SAP HANA applications. The extension includes tools such a

Cancel | Create Dev Space

Please NOTE: In the SAP BTP trial you are limited to only two Dev Spaces and only one can be active at a time.

4. Once the Dev Space reaches the green status of RUNNING, you can click on the name of the Dev Space and it will load into the editor within your browser.



Dev Spaces

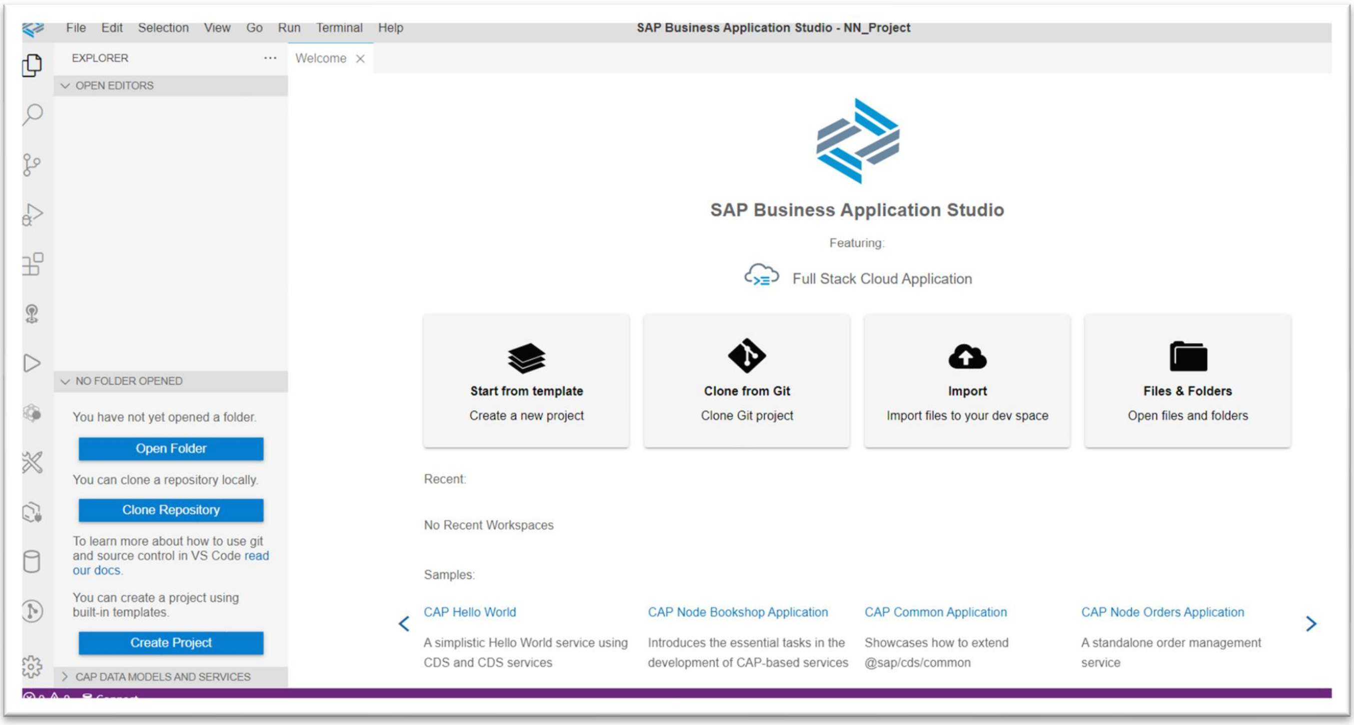
Create and manage your development environment according to the type of applications you want to develop.
You can add extensions and tools to further enhance your development options.

⚠ You're using a trial version. Any dev space that hasn't been running for 30 days will be deleted. See the full list of restrictions.

You are allowed a total of 2 dev spaces, with only 1 running at a time.

		Created On	ID	Disk Usage
NN_Project Full Stack Cloud Application	RUNNING	02/17/2022 2:38 PM	ws-d72vf	16.6 MB / 3.9 GB

5. You'll be redirected to your newly created SAP Business Application Studio Dev Space. Recommend you bookmark this URL so it's easier for you to access this dev space of your SAP Business Application Studio in the future.



SAP Business Application Studio - NN_Project

You have not yet opened a folder.

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

You can create a project using built-in templates.

SAP Business Application Studio

Featuring:

Full Stack Cloud Application

Start from template – Create a new project

Clone from Git – Clone Git project

Import – Import files to your dev space

Files & Folders – Open files and folders

Recent:

No Recent Workspaces

Samples:

CAP Hello World – A simplistic Hello World service using CDS and CDS services

CAP Node Bookshop Application – Introduces the essential tasks in the development of CAP-based services

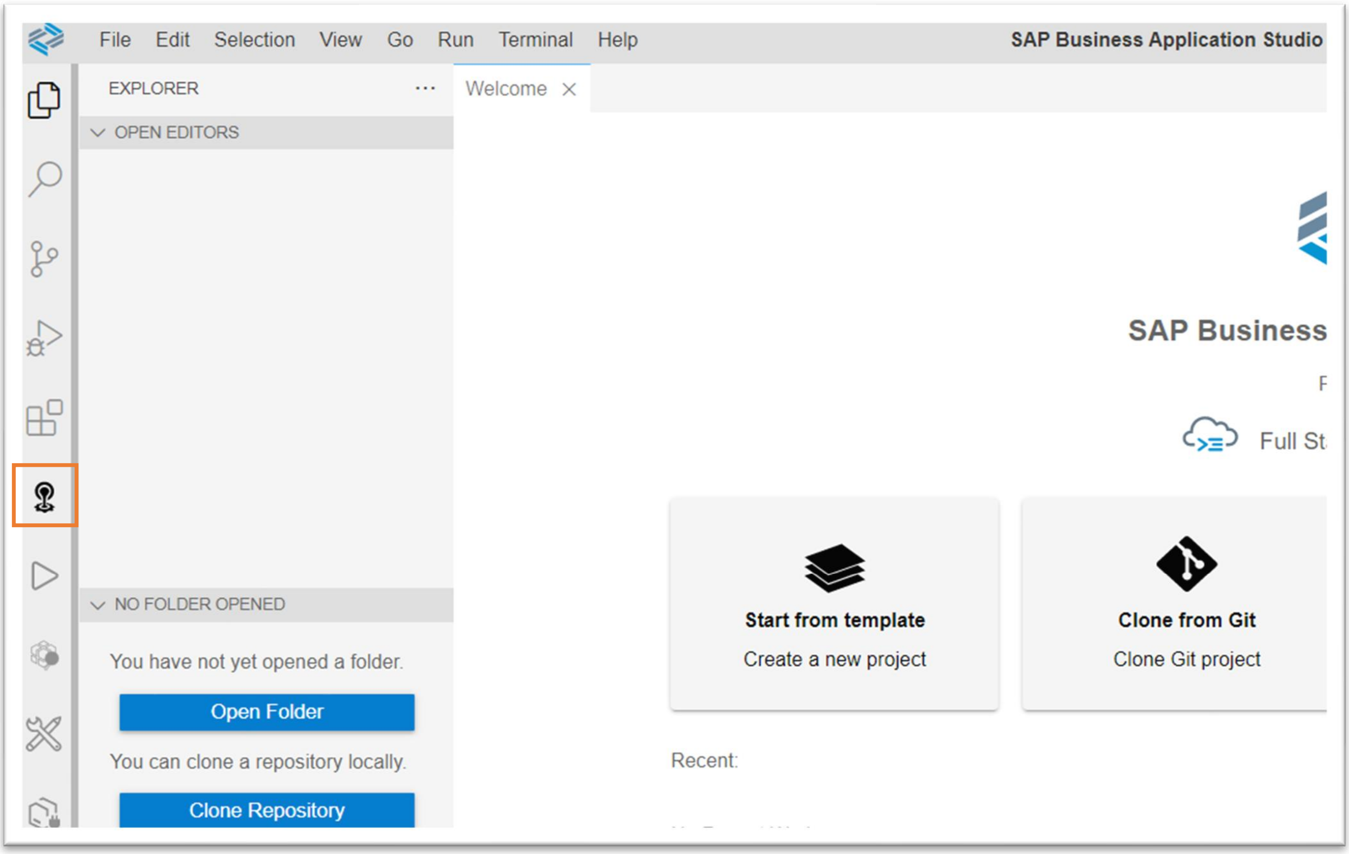
CAP Common Application – Showcases how to extend @sap/cds/common

CAP Node Orders Application – A standalone order management service

Step 2: Configure dev space

Before we create our project, we want to do a few more one-time configuration steps to prepare the Dev Space

1. On the left side of the Business Application Studio click on the Cloud Foundry targets icon



You have not yet opened a folder.

Open Folder

You can clone a repository locally.

Clone Repository

Start from template

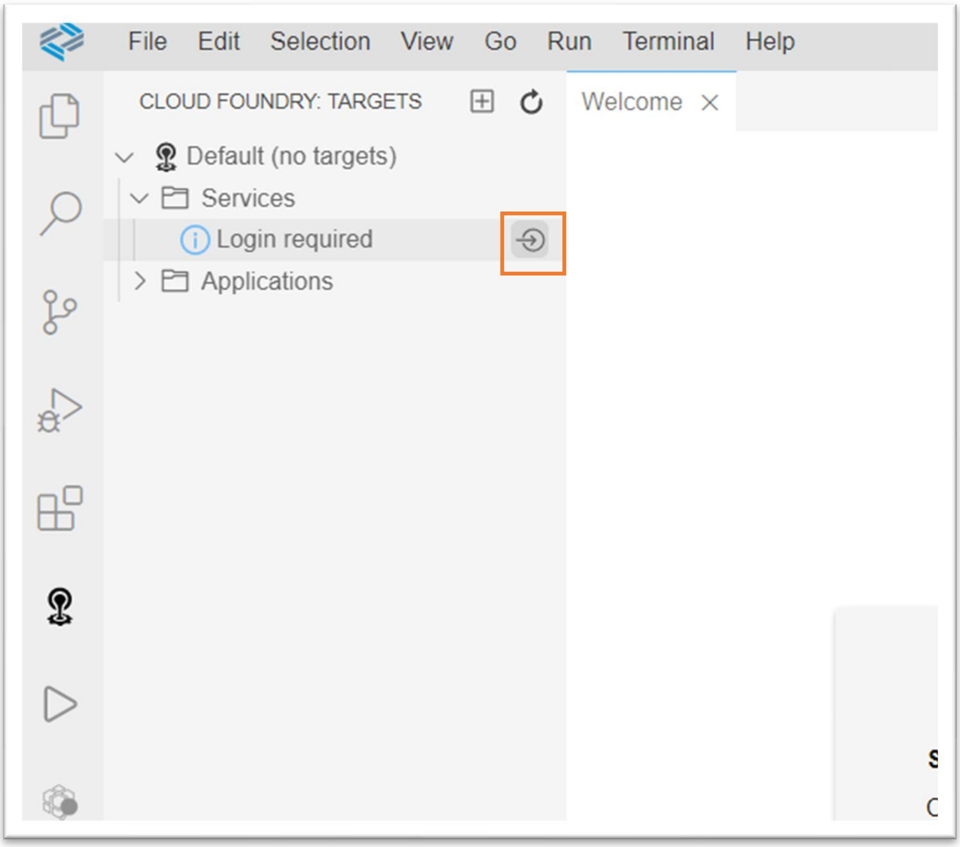
Create a new project

Clone from Git

Clone Git project

Recent:

2. In the Cloud Foundry Targets window, you can expand either Service or Applications and then click on the Logon icon to continue the configuration process



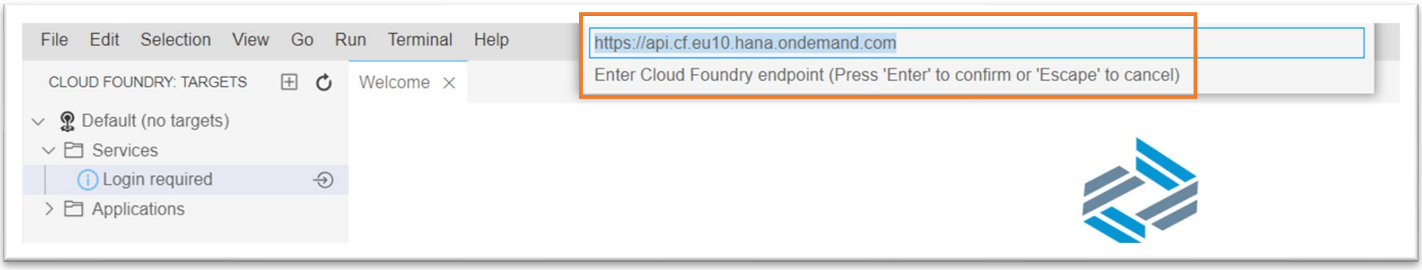
Default (no targets)

Services

Login required

Applications

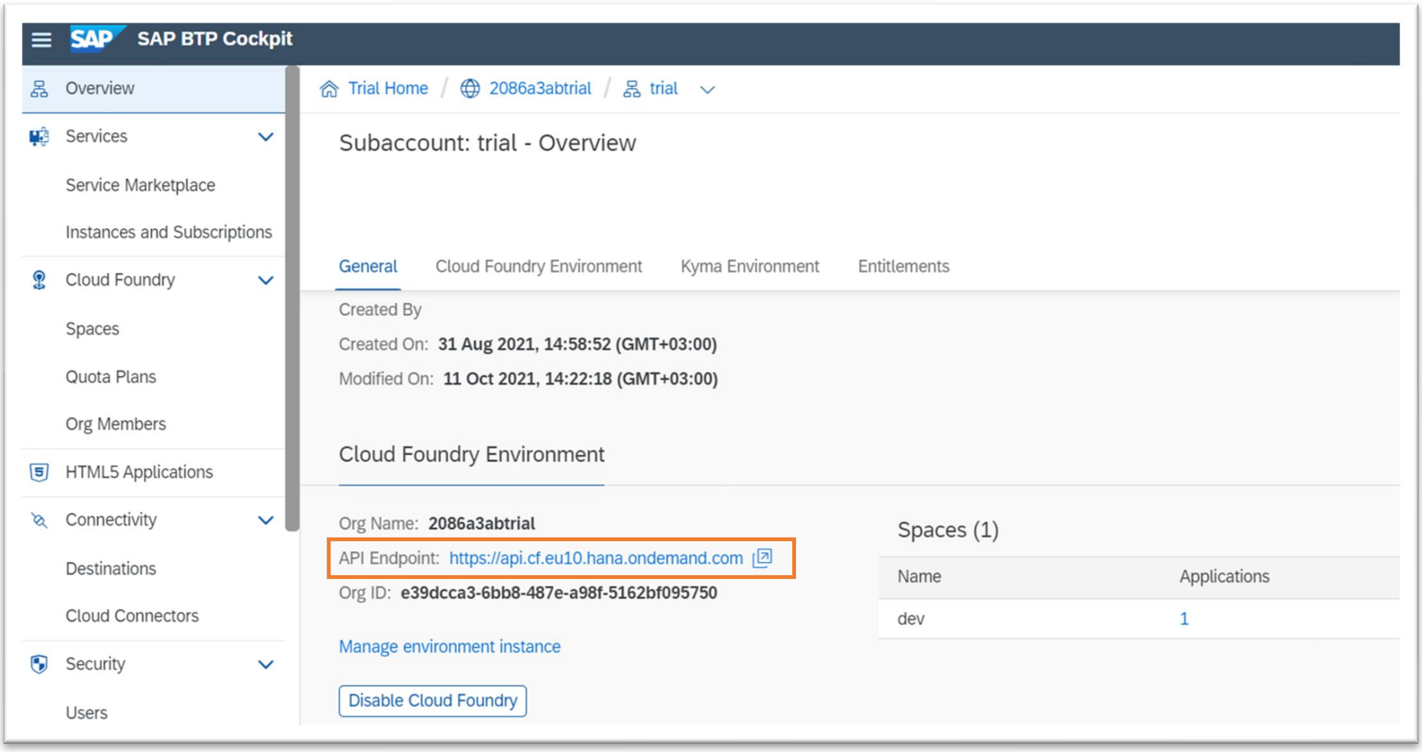
3. The command window will then open at the top of the SAP Business Application Studio. The first input will prompt you for the Cloud Foundry endpoint



https://api.cf.eu10.hana.ondemand.com

Enter Cloud Foundry endpoint (Press 'Enter' to confirm or 'Escape' to cancel)

The default value proposed is likely the correct value, but if you need to confirm; the value can be found in the SAP BTP cockpit at the Subaccount level.



Subaccount: trial - Overview

Created By

Created On: **31 Aug 2021, 14:58:52 (GMT+03:00)**

Modified On: **11 Oct 2021, 14:22:18 (GMT+03:00)**

Cloud Foundry Environment

Org Name: **2086a3abtrial**

API Endpoint: https://api.cf.eu10.hana.ondemand.com

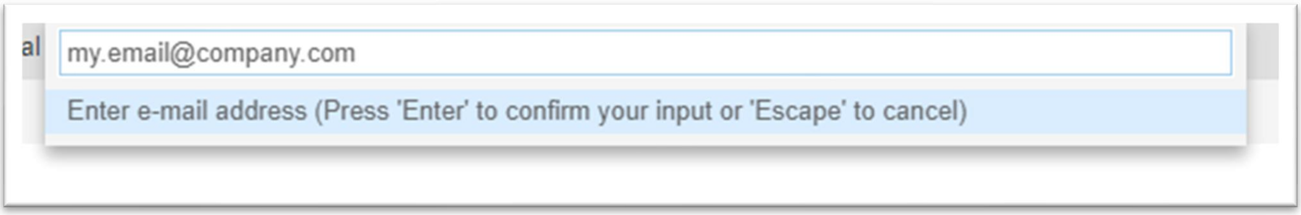
Org ID: **e39dcca3-6bb8-487e-a98f-5162bf095750**

Spaces (1)

Name	Applications
dev	1

Press Enter to confirm your input of the Cloud Foundry endpoint.

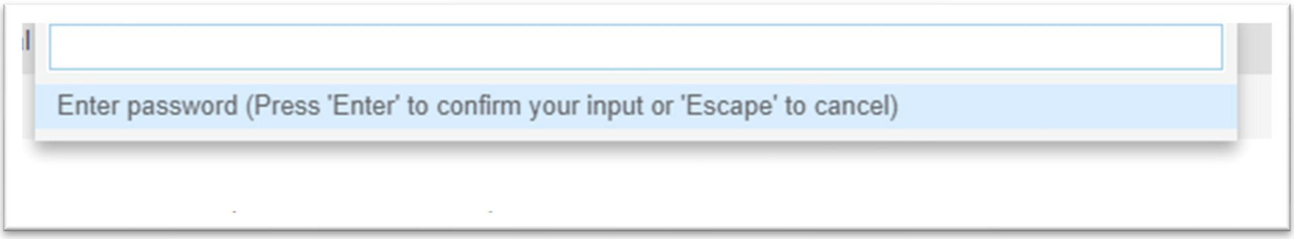
4. The next input field will ask you for the email address you used to create your SAP BTP trial account



my.email@company.com

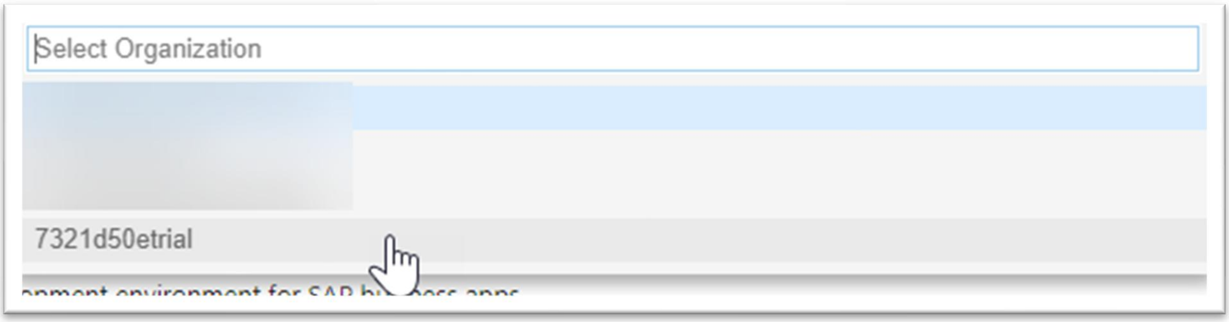
Enter e-mail address (Press 'Enter' to confirm your input or 'Escape' to cancel)

5. The next input will ask you for your SAP BTP trial account password



Enter password (Press 'Enter' to confirm your input or 'Escape' to cancel)

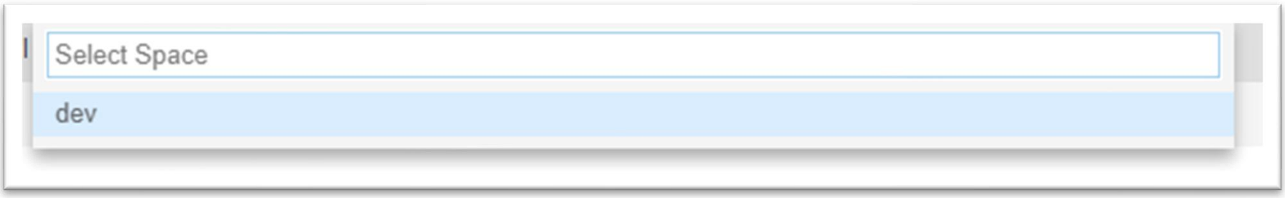
6. The next input will ask you for your Organization. In most situations, you will have a single choice. But like the API endpoint earlier, if you need to confirm the correct value it will be displayed in the top navigation of the SAP BTP cockpit



Select Organization

7321d50etrial

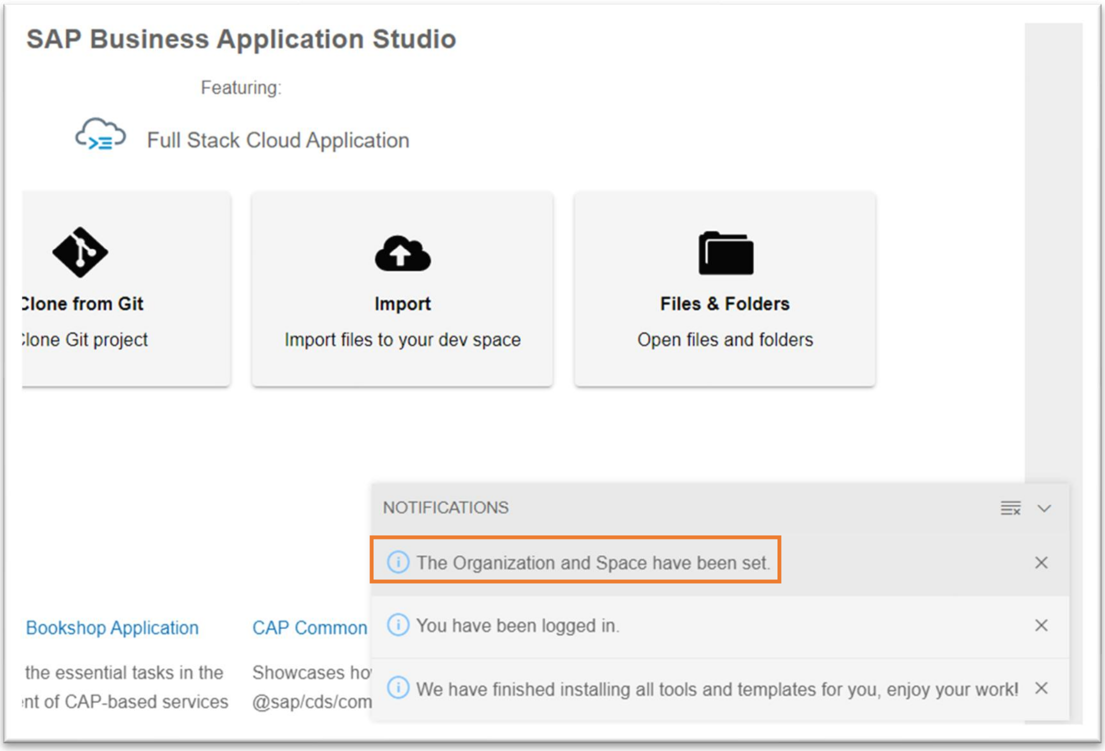
7. The final input will ask you for your Space. If you choose the endpoint API and Organization correctly, then you should have a single option of dev



Select Space

dev

8. Upon completion of all the inputs, you should see that the Organization and Space have been set.



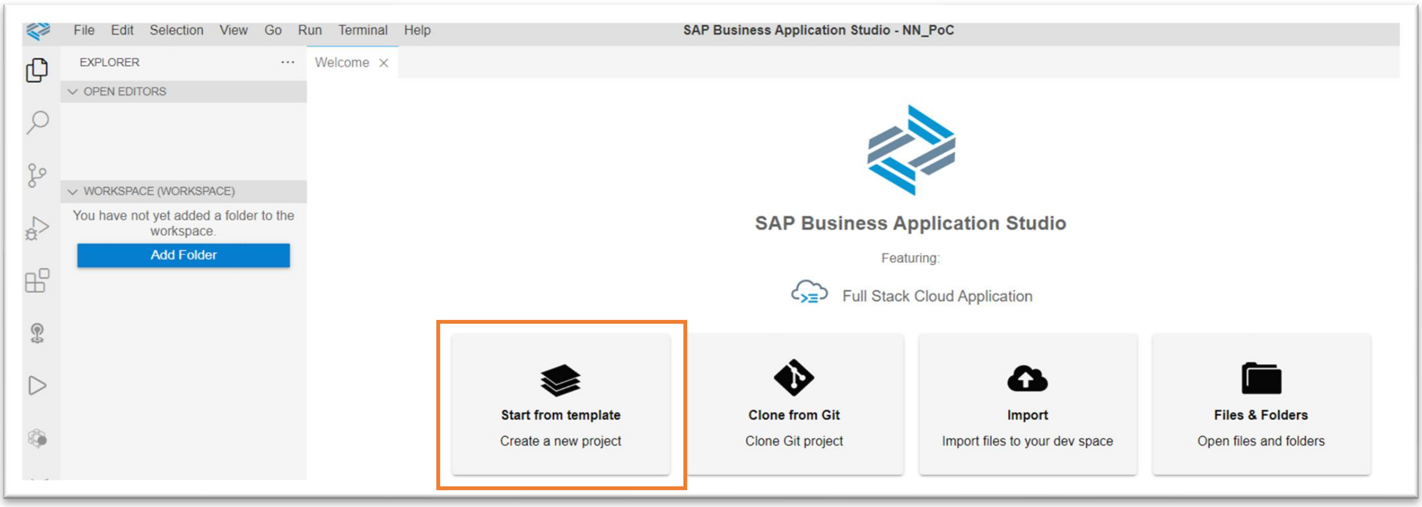
The Organization and Space have been set.

You have been logged in.

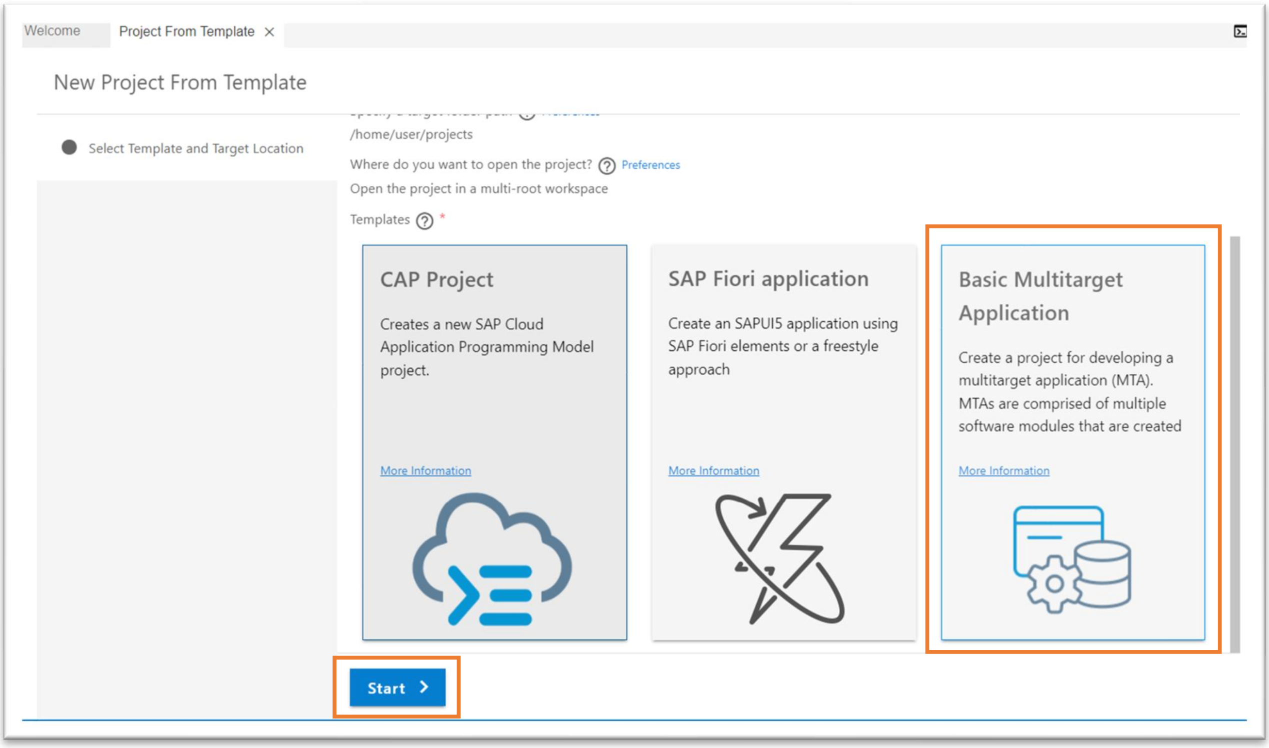
We have finished installing all tools and templates for you, enjoy your work!

Step 3: Create a new project

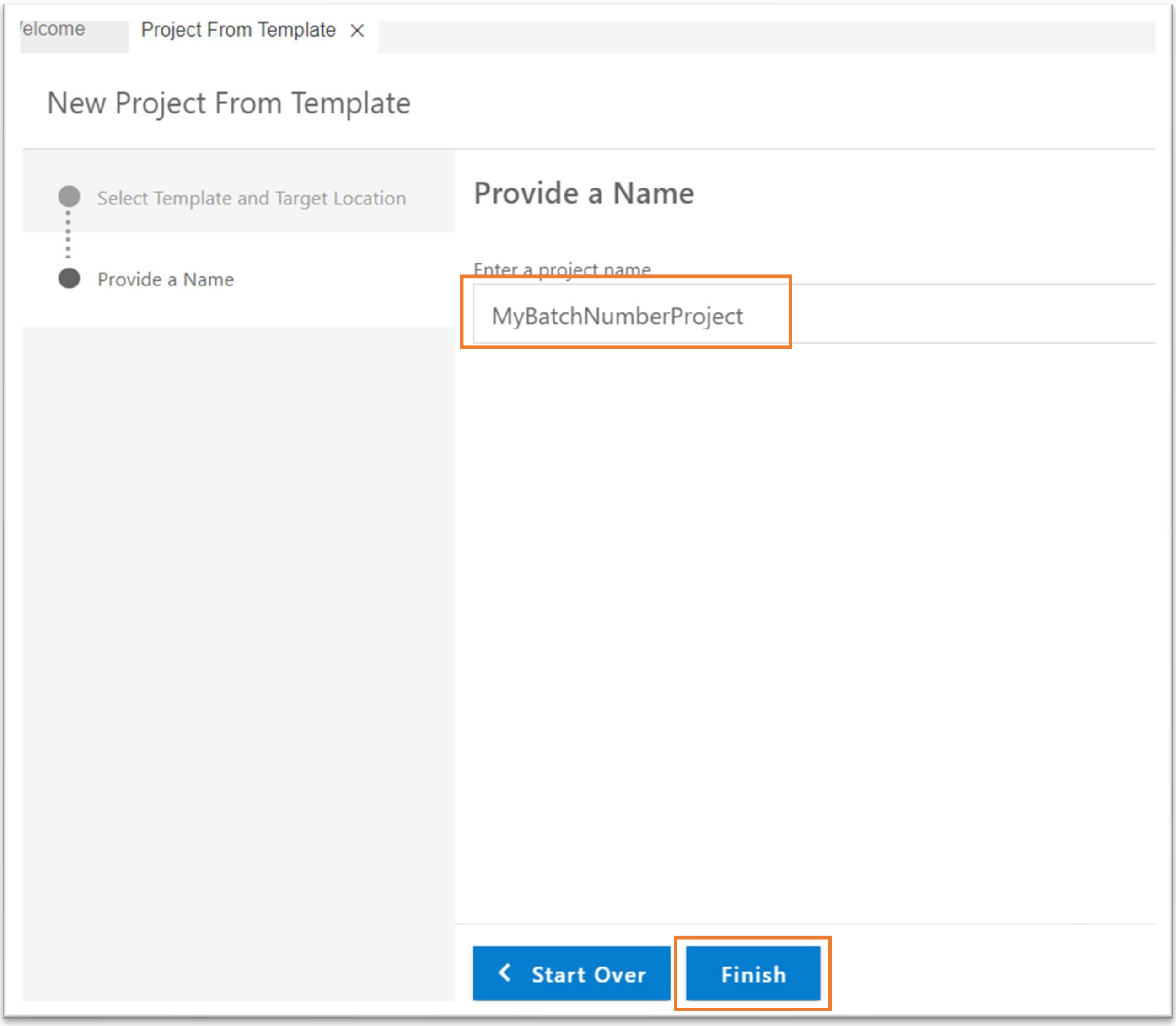
1. Return to the Explorer view. From the SAP Business Application Studio Welcome tab, click Start from template Create a new project


File Edit Selection View Go Run Terminal Help
SAP Business Application Studio - NN_PoC
EXPLORER
OPEN EDITORS
WORKSPACE (WORKSPACE)
You have not yet added a folder to the workspace.
Add Folder
Welcome
SAP Business Application Studio
Featuring:
Full Stack Cloud Application
Start from template
Create a new project
Clone from Git
Clone Git project
Import
Import files to your dev space
Files & Folders
Open files and folders

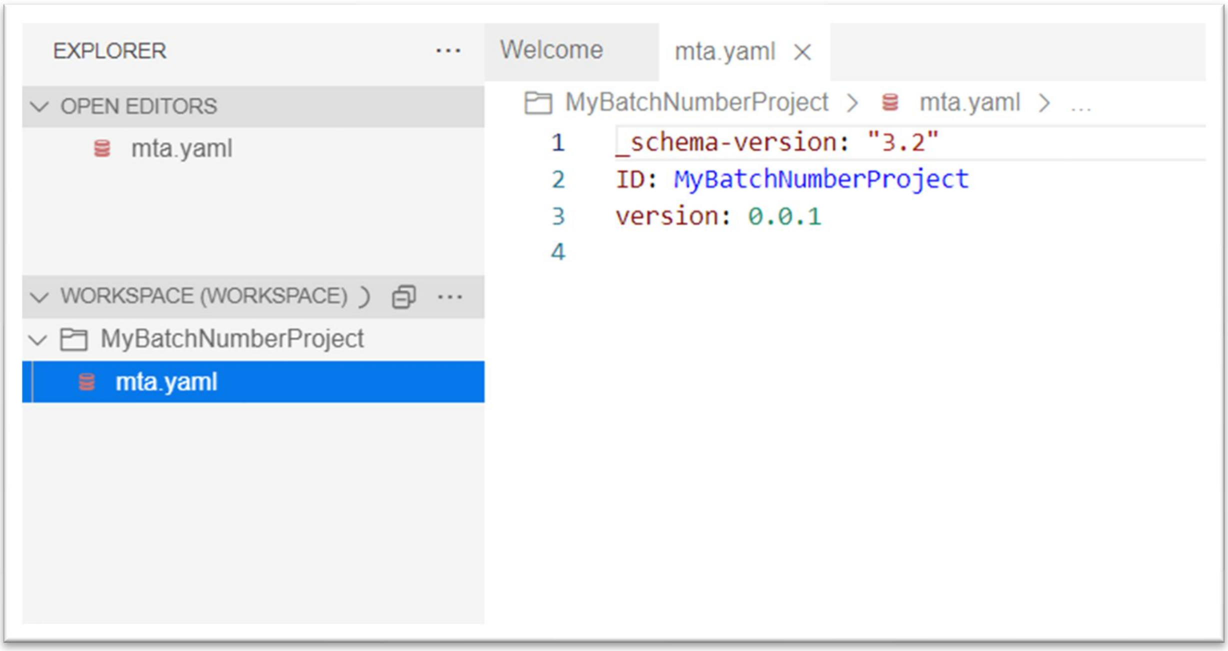
2. Choose **Basic Multitarget Application** and click Start


Welcome
Project From Template
New Project From Template
Select Template and Target Location
/home/user/projects
Where do you want to open the project?
Preferences
Open the project in a multi-root workspace
Templates *
CAP Project
Creates a new SAP Cloud Application Programming Model project.
More Information
SAP Fiori application
Create an SAPUI5 application using SAP Fiori elements or a freestyle approach
More Information
Basic Multitarget Application
Create a project for developing a multitarget application (MTA). MTAs are comprised of multiple software modules that are created
More Information
Start

3. Enter MyBatchNumberProject project name and click Finish



The wizard has generated an empty multi-target application with development descriptor mta.yaml.

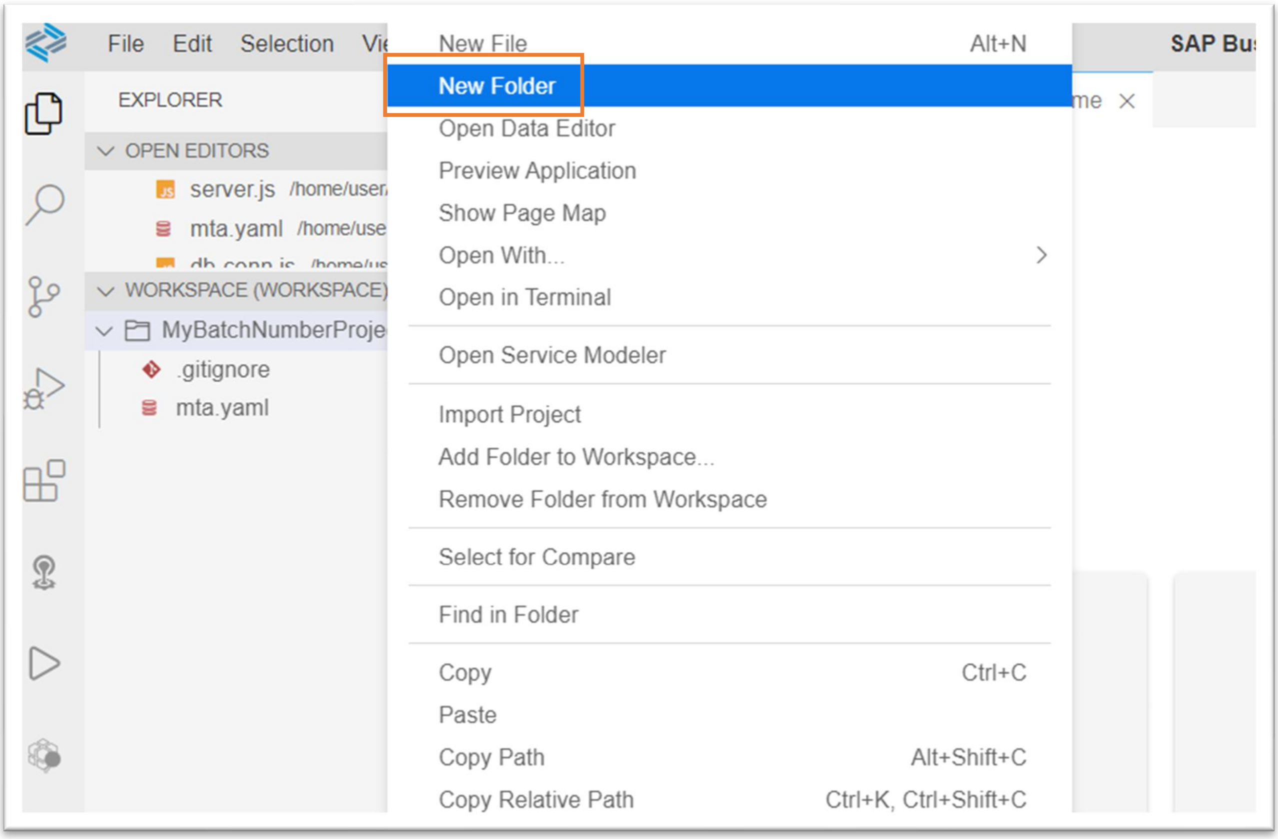


```yaml
_schema-version: "3.2"
ID: MyBatchNumberProject
version: 0.0.1

```

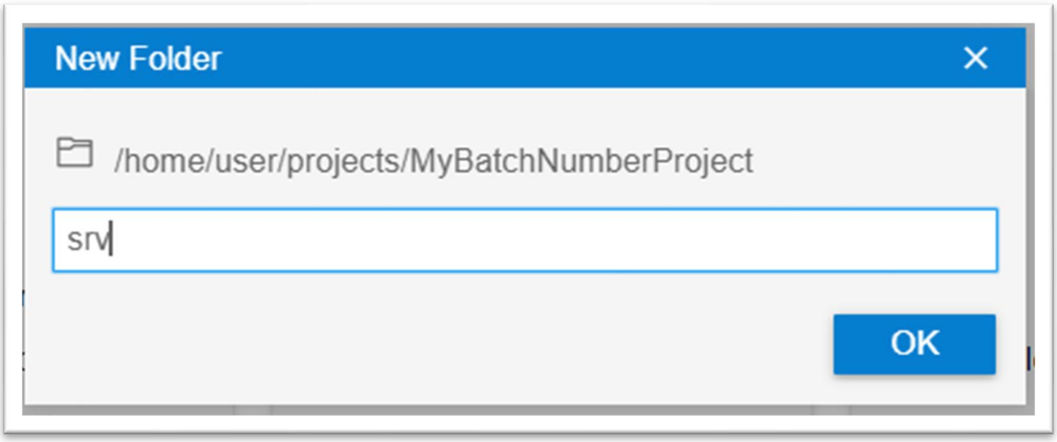
The next our steps to add content to the MyBatchNumberProject project.

Step 4: Adjust MTA project for new development

1. Choose MyBatchNumberProject project and click the right mouse button to create New Folder



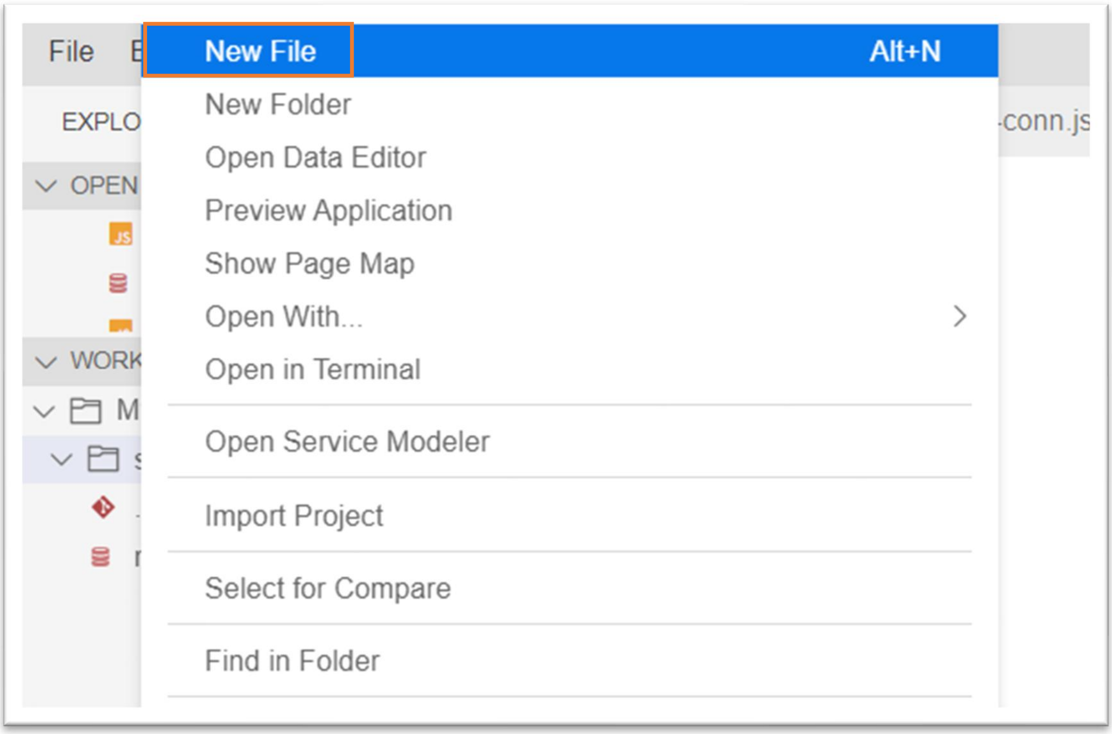
2. Specify srv name and click OK



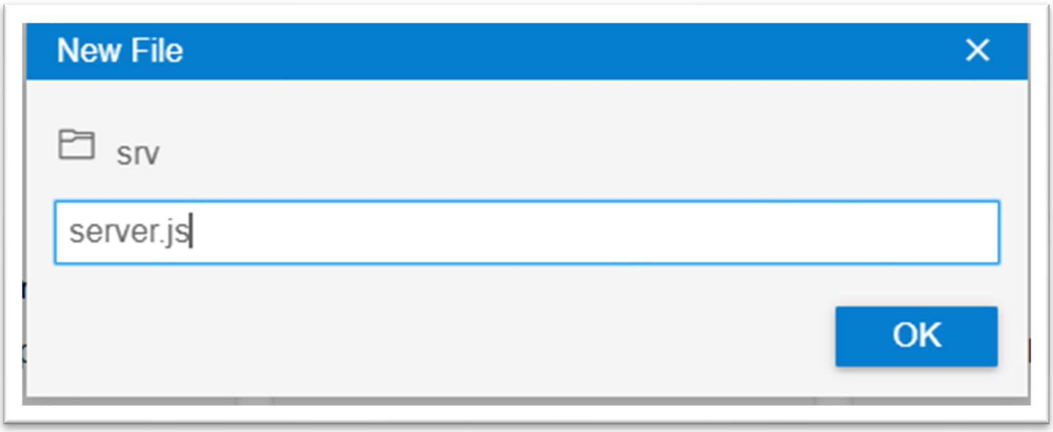
/home/user/projects/MyBatchNumberProject

srv

3. Add New File to srv folder.



4. Enter server.js name and click OK

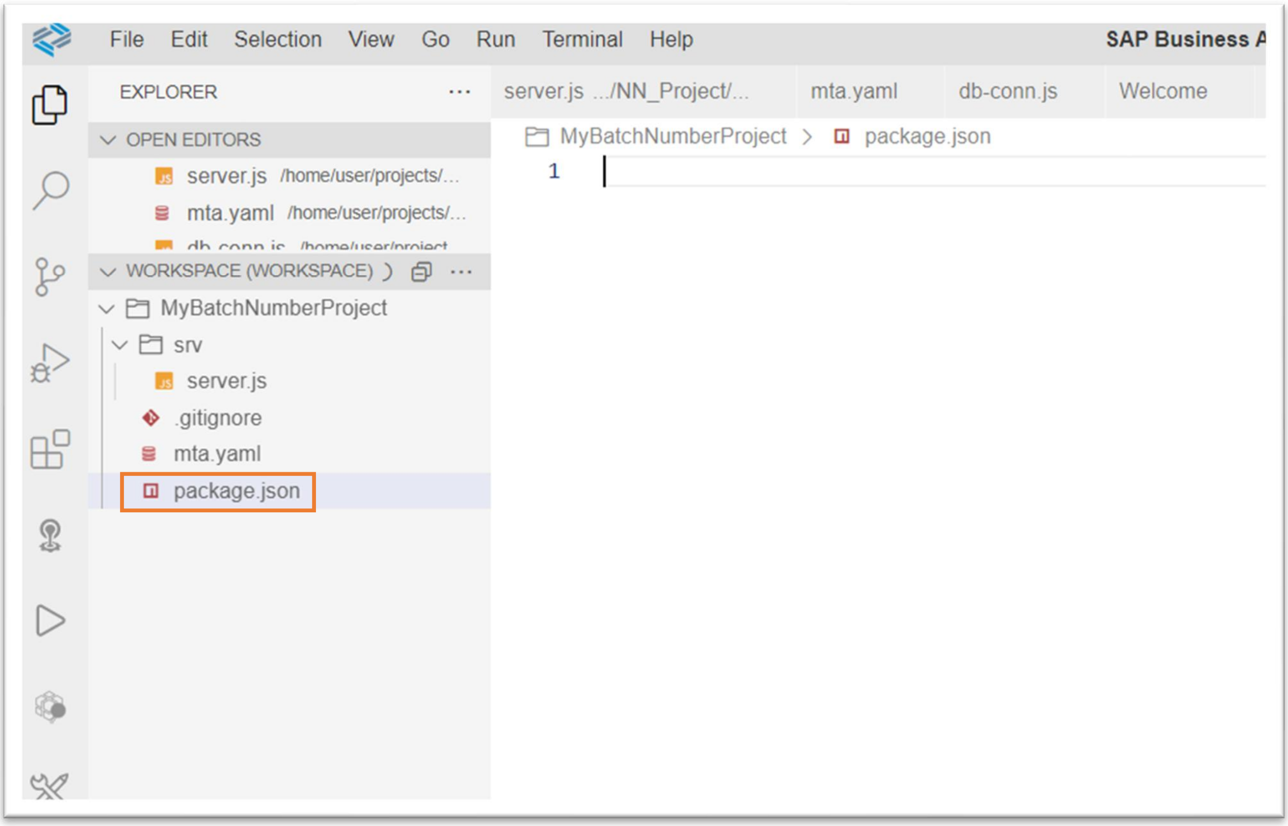


server.js

5. Open created server.js file and add the following Node.js content. It is a very simple REST API implementation.

```
'use strict';
const express = require('express');
const bodyParser = require('body-parser');

const app = express();
app.use(bodyParser.json());

// GET method to check that REST service deployed to Cloud Foundry
app.get('/', function (request, res) {
    res.send('Simple Rest API method GET - deployed!');
});

// POST method
app.post('/', async function (req, res) {
    res.send('Simple Rest API method POST - deployed!');
});

// PUT method
app.put('/', function (req, res) {
    res.send('Simple Rest API method PUT - deployed!');
});

// DELETE method
app.delete('/', function (req, res) {
    res.send('Simple Rest API method DELETE - deployed!');
});

var port = process.env.PORT || 3003;

app.listen(port, function () {
    console.log(' listening on port ' + port);
});
```

6. Choose MyBatchNumberProject project and click the right mouse button to create New File, name it - package.json



7. Add the following content that defines project dependencies and scripts to start, build and deploy the application.

```
{
    "name": "rest_api_example",
    "description": "simple REST API example",
    "version": "1.0.0",
    "private": true,
    "dependencies": {
      "express": "4.17.2",
      "body-parser": "1.19.1"
    },
    "scripts": {
      "start": "node srv/server.js",
      "build:cf": "mbt build",
      "deploy:cf": "cf deploy mta_archives/MyBatchNumberProject_1.0.0.mtar"
    }
  }
```

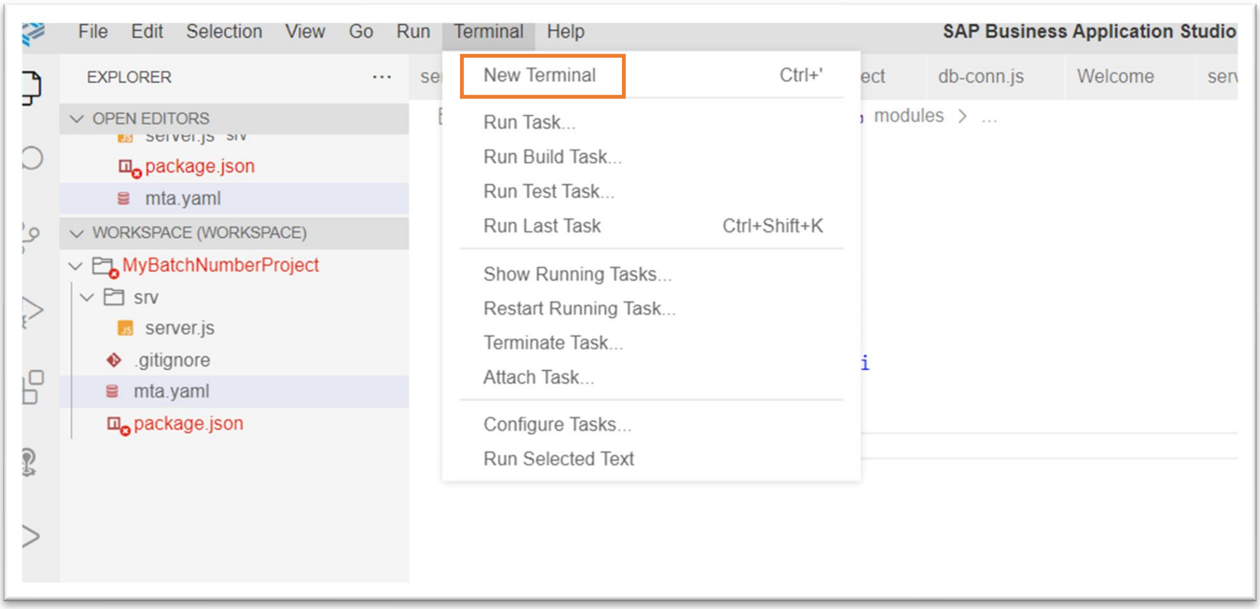
8. Open mta.yaml file and replace existing content on the following content that defines service module name and type.

```
_schema-version: "3.2"
ID: MyBatchNumberProject
version: 1.0.0
modules:
 - name: next-number-example
   type: nodejs
   path: .
   provides:
    - name: next-number-example_api
      properties:
         url: ${default-url}
```

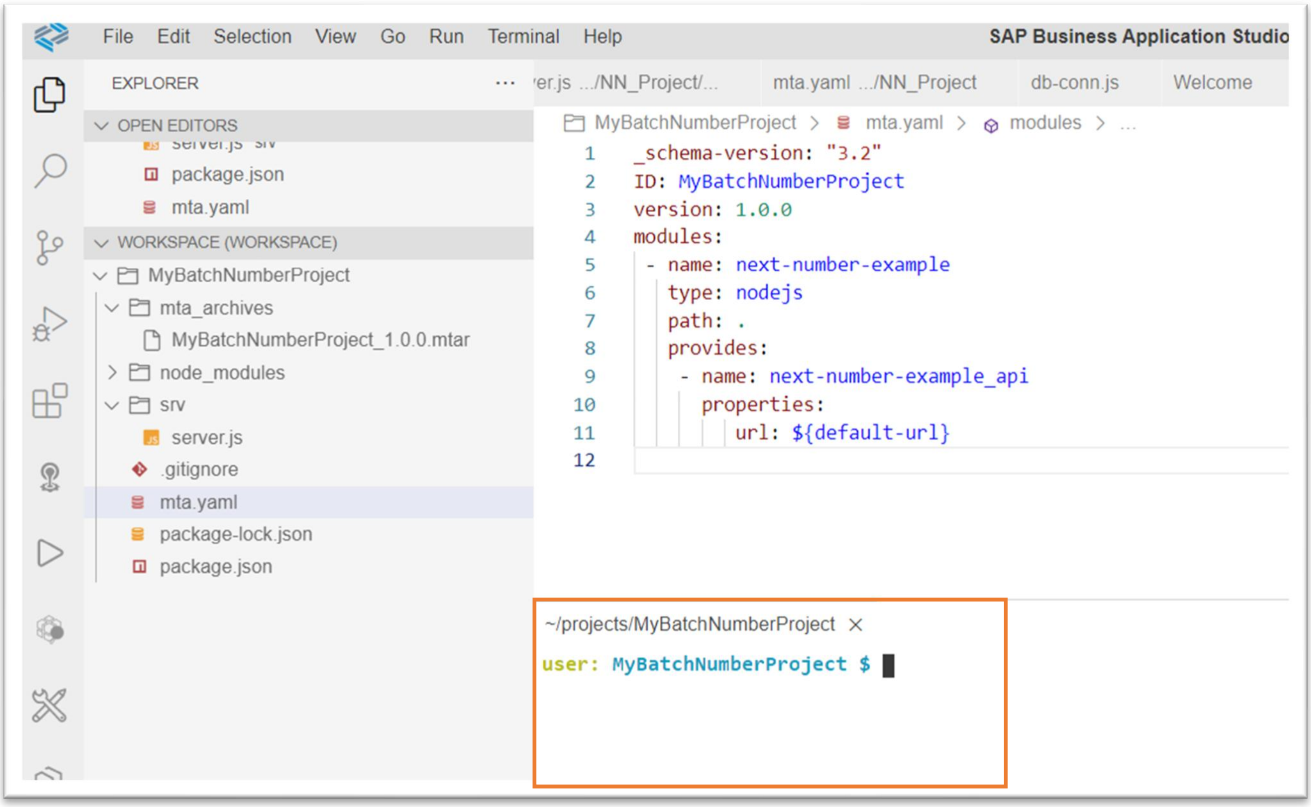
Step 5: Install dependencies

We are using several dependent Node.js modules in our project. They are referenced in the dependencies section of the package.json file we were just editing. These dependencies need to be installed into your project before we can do any testing.

1. Open a New Terminal



2. The terminal will open in the bottom right of the SAP Business Application Studio

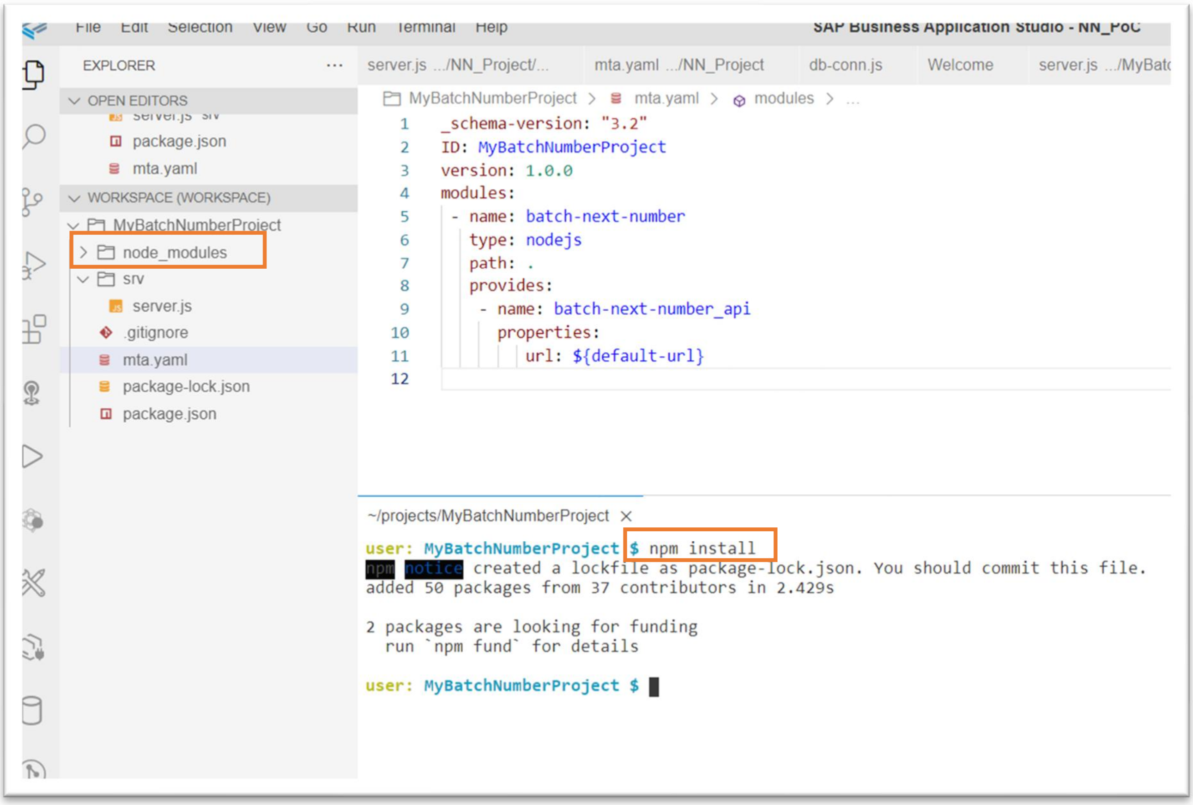


```
_schema-version: "3.2"
ID: MyBatchNumberProject
version: 1.0.0
modules:
  - name: next-number-example
    type: nodejs
    path: .
    provides:
      - name: next-number-example_api
        properties:
          url: ${default-url}
```

```
user: MyBatchNumberProject $
```

3. From the terminal run the command: npm install

Expected result that node_modules folder was created that contains needed dependencies.

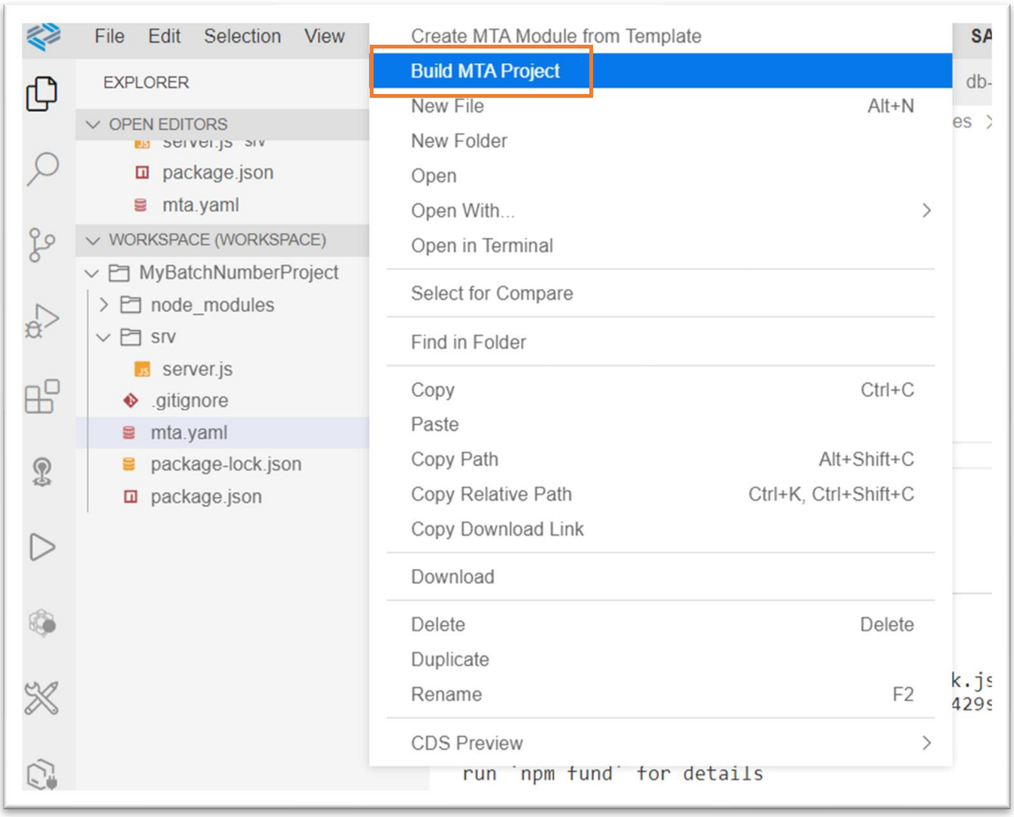


```
_schema-version: "3.2"
ID: MyBatchNumberProject
version: 1.0.0
modules:
  - name: batch-next-number
    type: nodejs
    path: .
    provides:
      - name: batch-next-number_api
        properties:
          url: ${default-url}
```

```
user: MyBatchNumberProject $ npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 50 packages from 37 contributors in 2.429s

2 packages are looking for funding
  run `npm fund` for details

user: MyBatchNumberProject $
```

Step 6: Build the project

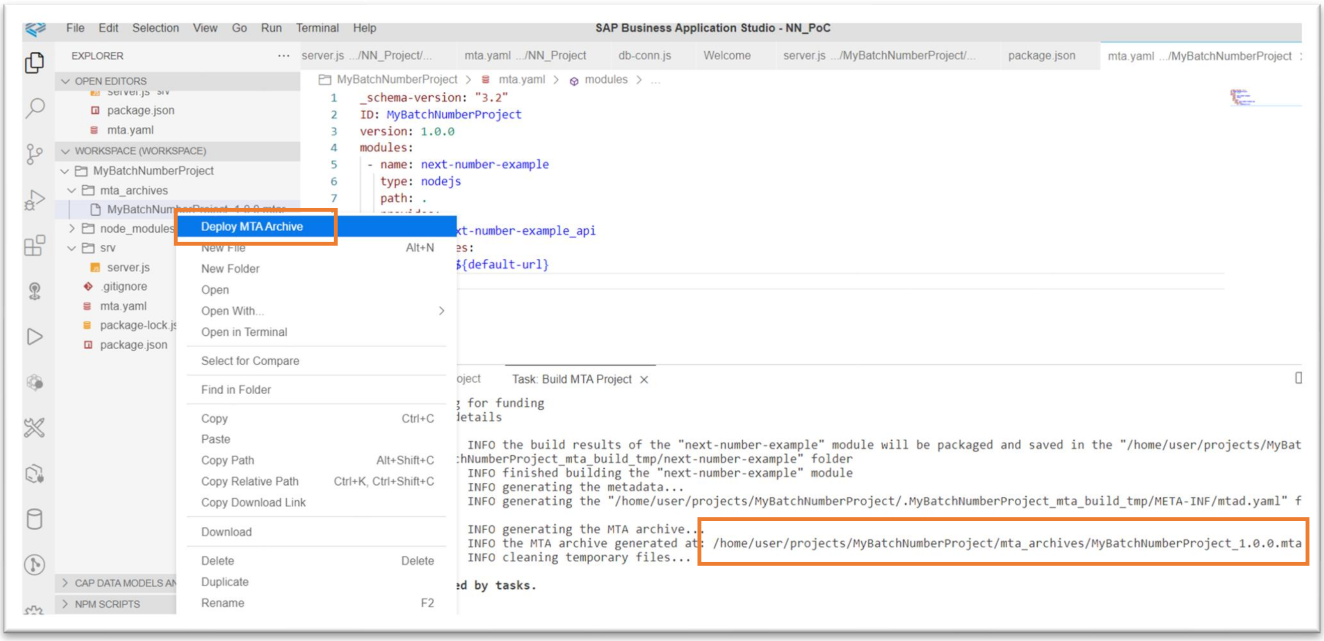
Right-click on the mta.yaml file and select Build MTA Project. This will trigger a process that generates the .mtar deployment artifact.



Step 7: Deploy the project

Once the build process has been completed, look for the newly generated file in the project tree, you'll find it in the mta_archives folder.

1. Right-click on this file and select Deploy MTA Archive.

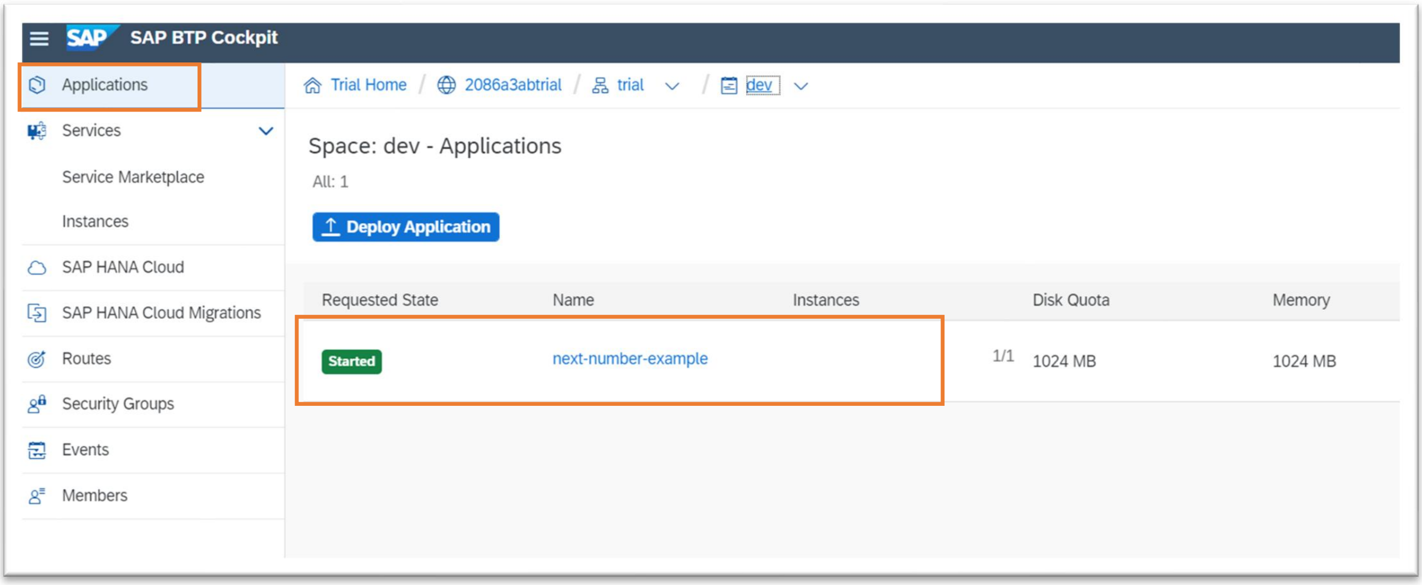


```
_schema-version: "3.2"
ID: MyBatchNumberProject
version: 1.0.0
modules:
  - name: next-number-example
    type: nodejs
    path: .
```

```
  INFO the build results of the "next-number-example" module will be packaged and saved in the "/home/user/projects/MyBat
chNumberProject_mta_build_tmp/next-number-example" folder
  INFO finished building the "next-number-example" module
  INFO generating the metadata...
  INFO generating the "/home/user/projects/MyBatchNumberProject/.MyBatchNumberProject_mta_build_tmp/META-INF/mtad.yaml" f

  INFO generating the MTA archive...
  INFO the MTA archive generated at: /home/user/projects/MyBatchNumberProject/mta_archives/MyBatchNumberProject_1.0.0.mta
  INFO cleaning temporary files...
```

The result of deployment can be checked at SAP BTP Cockpit in Cloud Foundry space in Applications left side menu.

2. Verify that next-number-example application is Started

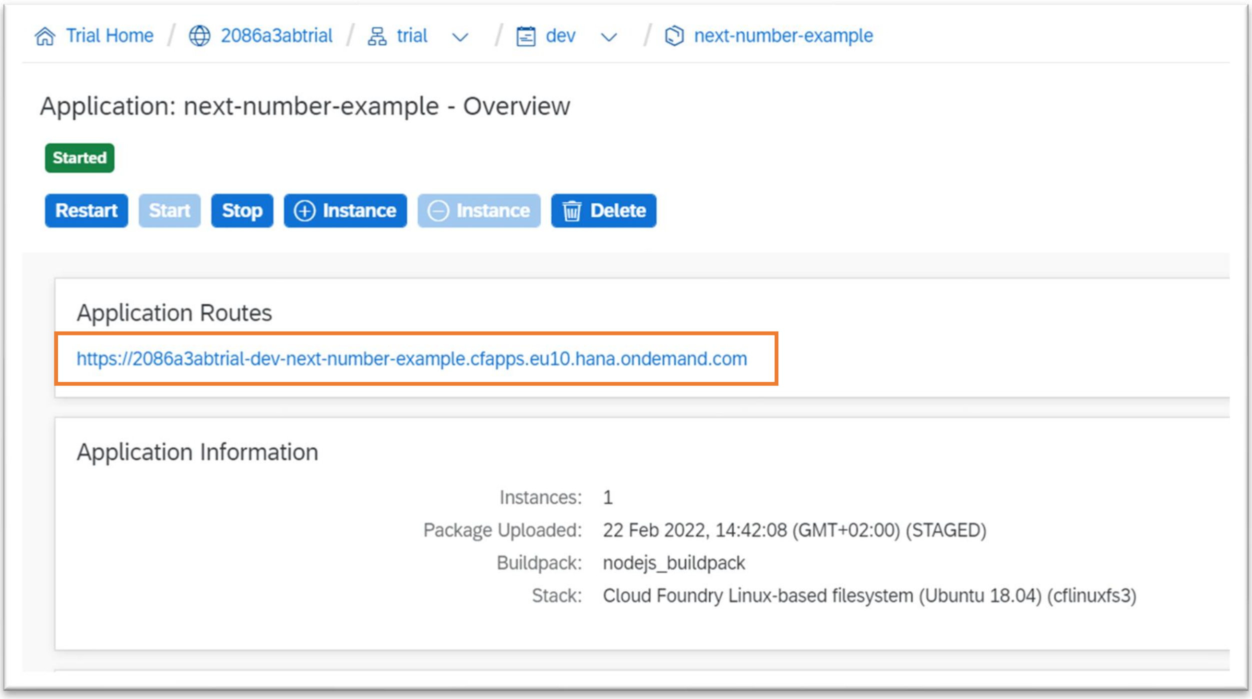


Space: dev - Applications

All: 1

Requested State	Name	Instances	Disk Quota	Memory
Started	next-number-example	1/1	1024 MB	1024 MB

3. Click on the link in Application Routes section


Trial Home / 2086a3abtrial / trial / dev / next-number-example
Started
Restart
Start
Stop
Instance
Instance
Delete

Application: next-number-example - Overview

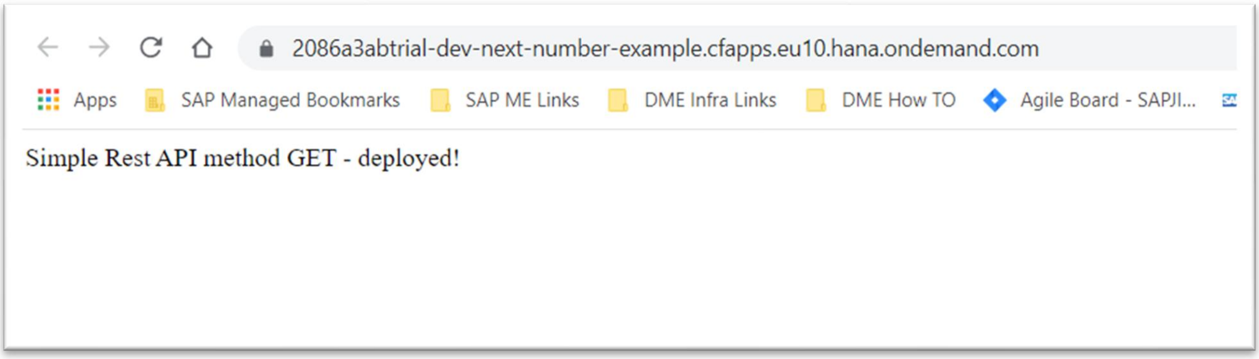
Application Routes

https://2086a3abtrial-dev-next-number-example.cfapps.eu10.hana.ondemand.com

Application Information

Instances: 1
Package Uploaded: 22 Feb 2022, 14:42:08 (GMT+02:00) (STAGED)
Buildpack: nodejs_buildpack
Stack: Cloud Foundry Linux-based filesystem (Ubuntu 18.04) (cflinuxfs3)

4. The result should be the following. REST API is successfully deployed to Cloud Foundry!



Simple Rest API method GET - deployed!

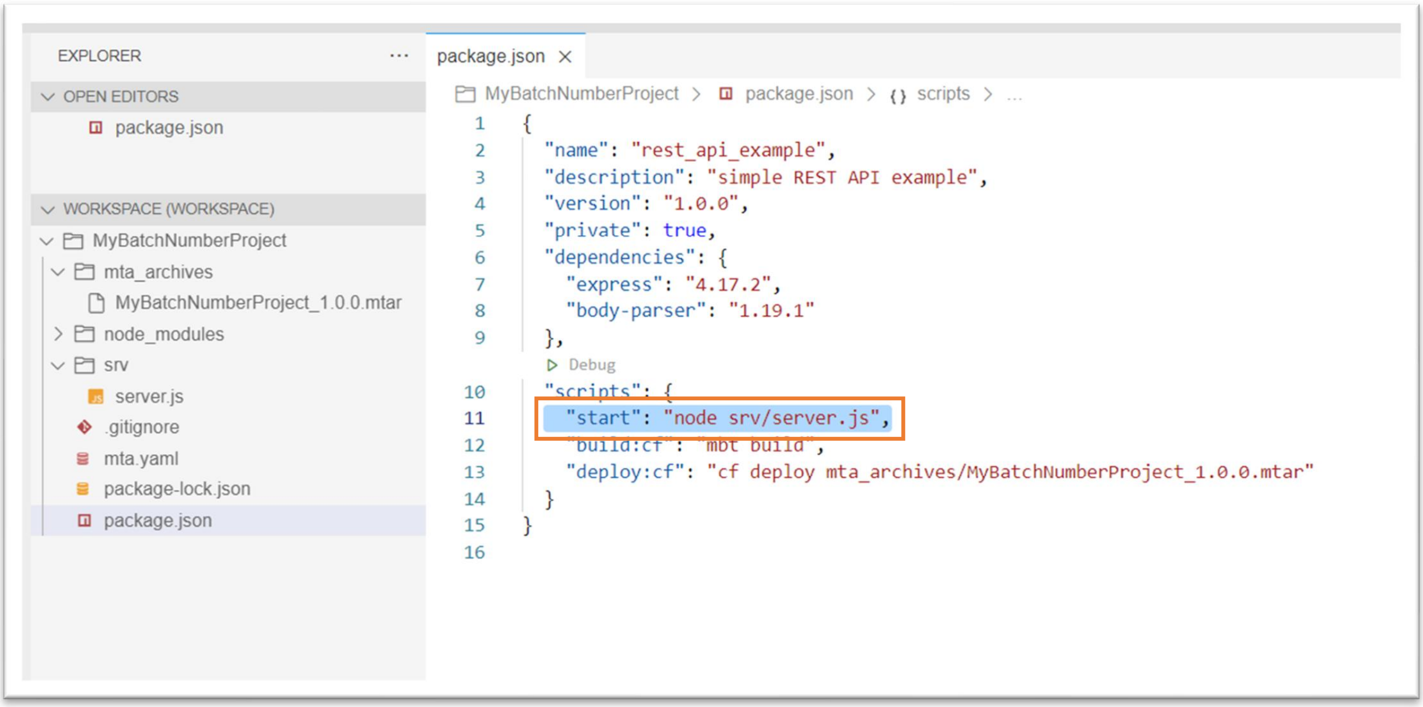
How to run the application locally in SAP Business Application Studio without deployment to Cloud Foundry?

If you want to test application locally without deploying to Cloud Foundry then execute the following steps:

1. Open package.json file, adjust it with “start” script as shown below

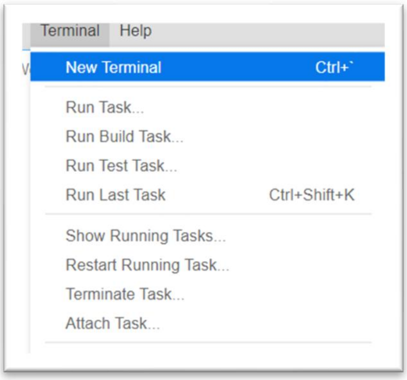
```
"start": "node srv/server.js"
```

This script means that server.js file from srv folder will be used to start application.

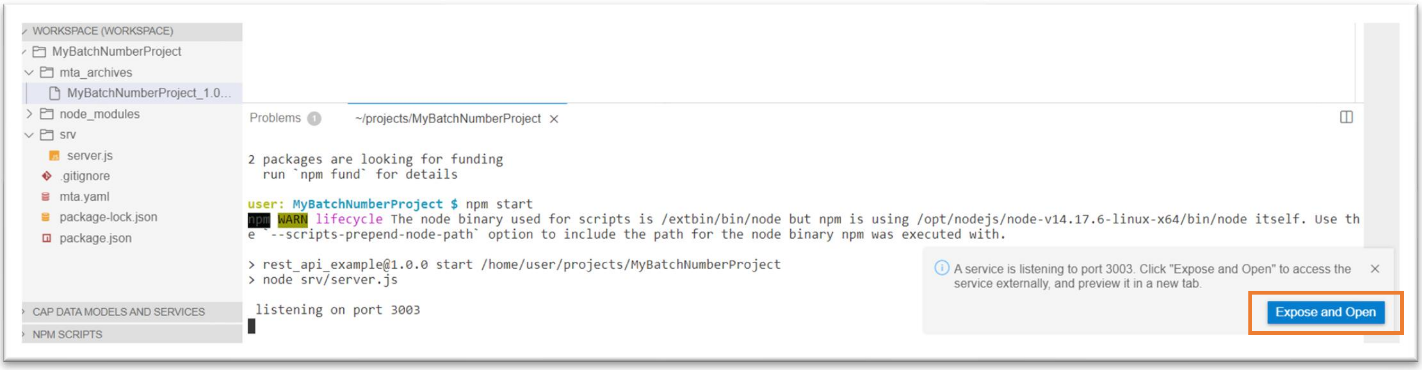


```
{
  "name": "rest_api_example",
  "description": "simple REST API example",
  "version": "1.0.0",
  "private": true,
  "dependencies": {
    "express": "4.17.2",
    "body-parser": "1.19.1"
  },
  "scripts": {
    "start": "node srv/server.js",
    "build:cf": "mbt build",
    "deploy:cf": "cf deploy mta_archives/MyBatchNumberProject_1.0.0.mtar"
  }
}
```

2. Open New Terminal window



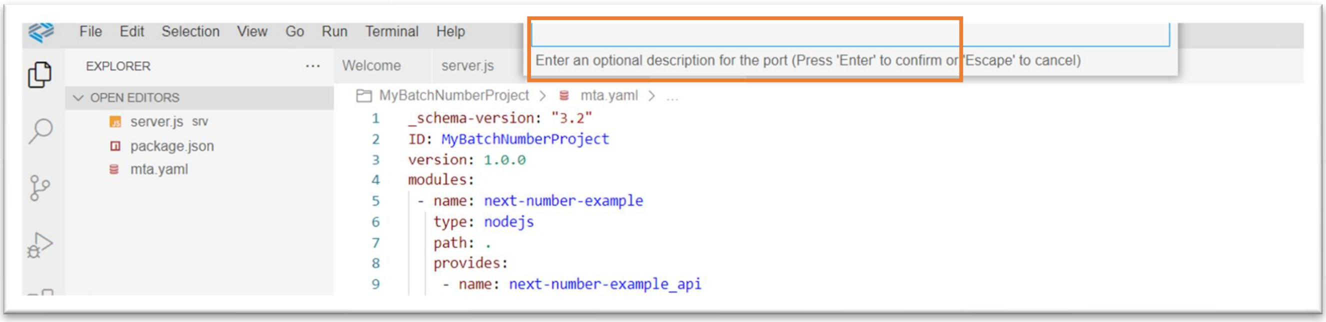
3. From the terminal run the command: npm start
4. Click on Expose and Open



```
2 packages are looking for funding
  run `npm fund` for details

user: MyBatchNumberProject $ npm start
npm WARN lifecycle The node binary used for scripts is /extbin/bin/node but npm is using /opt/nodejs/node-v14.17.6-linux-x64/bin/node itself. Use th
e `--scripts-prepend-node-path` option to include the path for the node binary npm was executed with.

> rest_api_example@1.0.0 start /home/user/projects/MyBatchNumberProject
> node srv/server.js

 listening on port 3003
```

A service is listening to port 3003. Click "Expose and Open" to access the service externally, and preview it in a new tab.

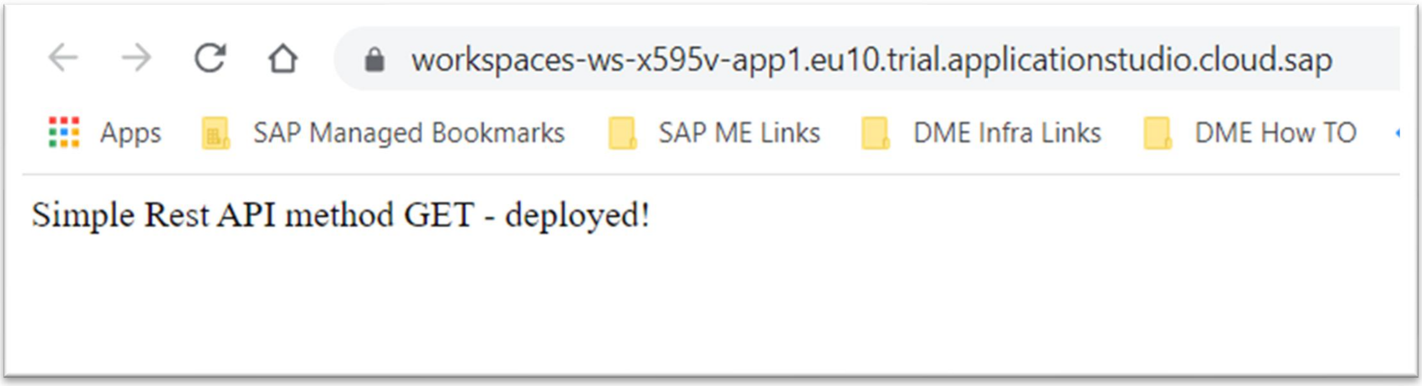
5. Enter a description for the port and press Enter to confirm.



Enter an optional description for the port (Press 'Enter' to confirm or 'Escape' to cancel)

```
_schema-version: "3.2"
ID: MyBatchNumberProject
version: 1.0.0
modules:
- name: next-number-example
  type: nodejs
  path: .
  provides:
   - name: next-number-example_api
```

As result, you will get a browser open with REST API running locally in the workspace


workspaces-ws-x595v-app1.eu10.trial.applicationstudio.cloud.sap
Apps
SAP Managed Bookmarks
SAP ME Links
DME Infra Links
DME How TO

Simple Rest API method GET - deployed!

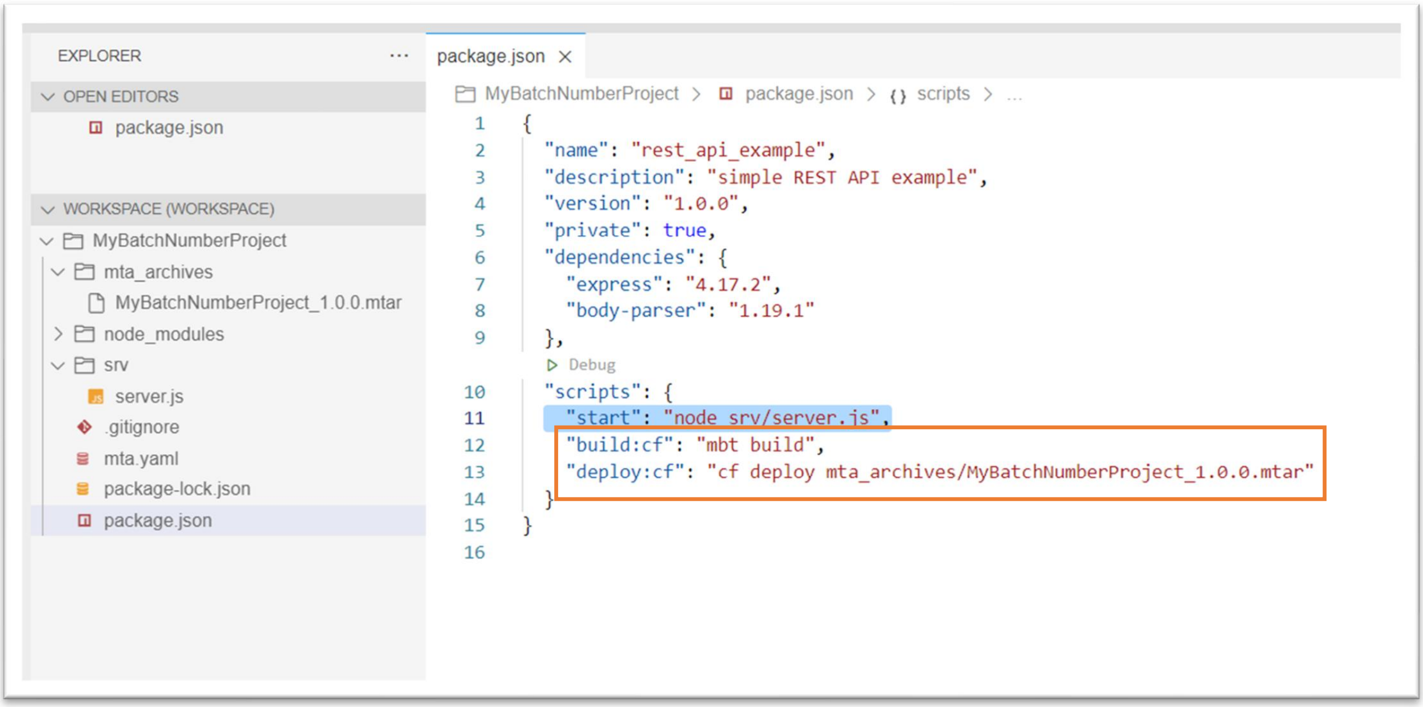
Note PLEASE: Stop running REST API locally you should return to Terminal window and press Ctrl + C.

How to build and deploy the application from the command line in the Terminal window?

If you want to build and deploy application from the command line, then execute the following steps:

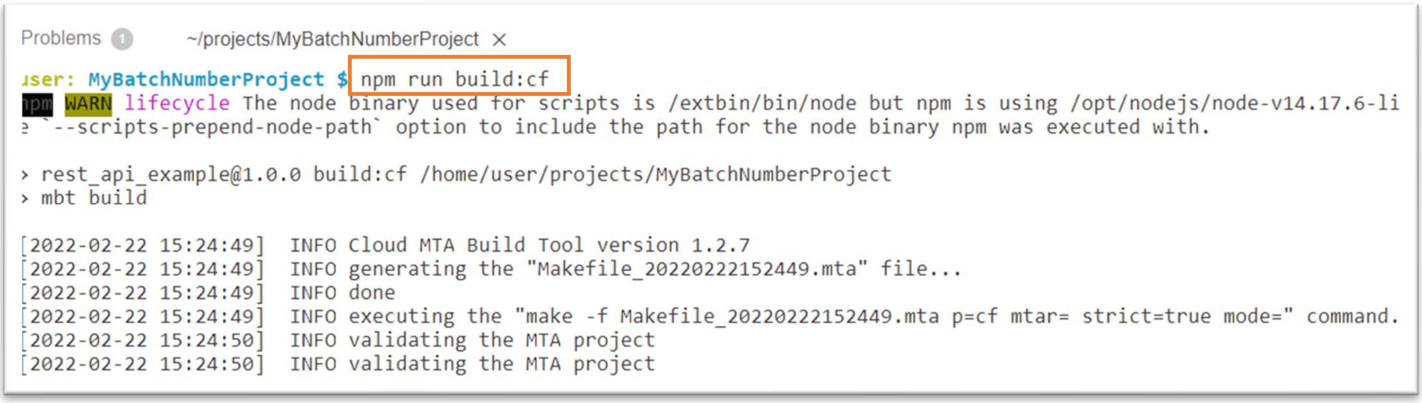
1. Add build and deploy scripts to package.json file content

```
"build:cf": "mbt build",
"deploy:cf": "cf deploy mta_archives/MyBatchNumberProject_1.0.0.mtar"
```



```json
{
  "name": "rest_api_example",
  "description": "simple REST API example",
  "version": "1.0.0",
  "private": true,
  "dependencies": {
    "express": "4.17.2",
    "body-parser": "1.19.1"
  },
  "scripts": {
    "start": "node srv/server.js",
    "build:cf": "mbt build",
    "deploy:cf": "cf deploy mta_archives/MyBatchNumberProject_1.0.0.mtar"
  }
}
```

2. Open New Terminal window and run the command there: npm run build:cf

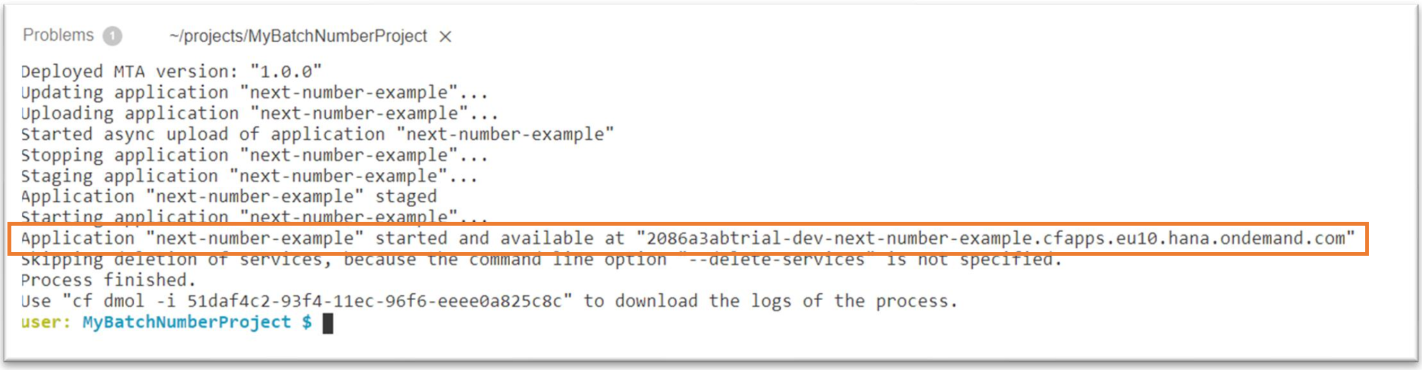


```
ıser: MyBatchNumberProject $ npm run build:cf
npm WARN lifecycle The node binary used for scripts is /extbin/bin/node but npm is using /opt/nodejs/node-v14.17.6-li
e `--scripts-prepend-node-path` option to include the path for the node binary npm was executed with.

> rest_api_example@1.0.0 build:cf /home/user/projects/MyBatchNumberProject
> mbt build

[2022-02-22 15:24:49]  INFO Cloud MTA Build Tool version 1.2.7
[2022-02-22 15:24:49]  INFO generating the "Makefile_20220222152449.mta" file...
[2022-02-22 15:24:49]  INFO done
[2022-02-22 15:24:49]  INFO executing the "make -f Makefile_20220222152449.mta p=cf mtar= strict=true mode=" command.
[2022-02-22 15:24:50]  INFO validating the MTA project
[2022-02-22 15:24:50]  INFO validating the MTA project
```

3. As a result of this script execution the .mtar was generated in mta_archives folder

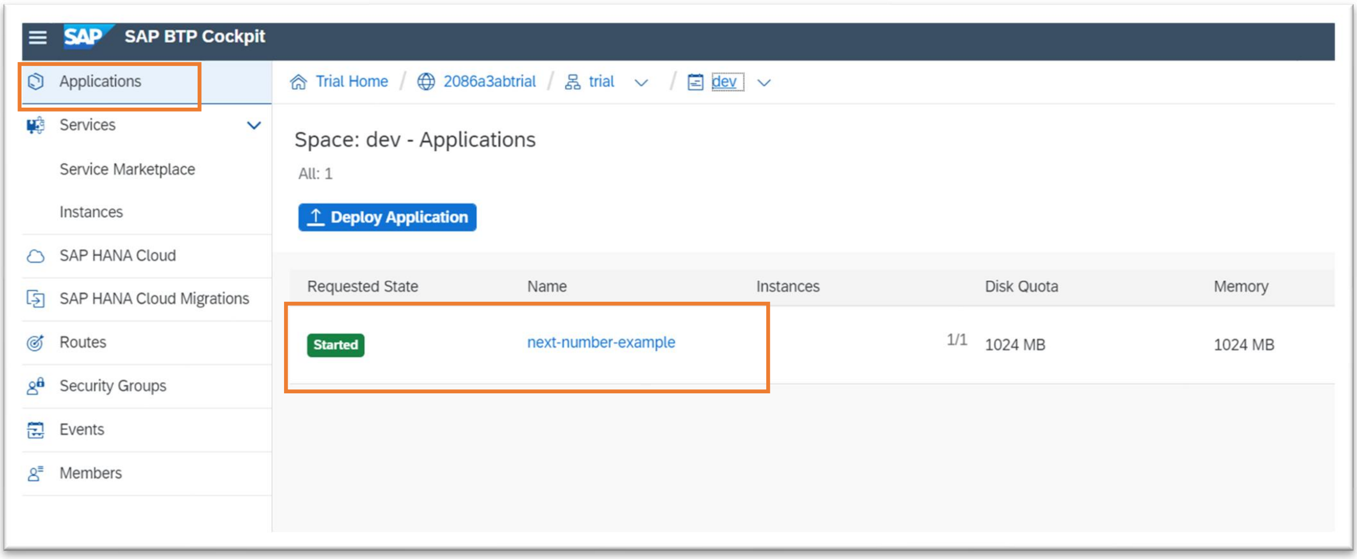
INFO the MTA archive generated at: /home/user/projects/MyBatchNumberProject/mta_archives/MyBatchNumberProject_1.0.0.mtar

4. Run the deployment command in terminal window: npm run deploy:cf



```
Deployed MTA version: "1.0.0"
Updating application "next-number-example"...
Uploading application "next-number-example"...
Started async upload of application "next-number-example"
Stopping application "next-number-example"...
Staging application "next-number-example"...
Application "next-number-example" staged
Starting application "next-number-example"...
Application "next-number-example" started and available at "2086a3abtrial-dev-next-number-example.cfapps.eu10.hana.ondemand.com"
Skipping deletion of services, because the command line option "--delete-services" is not specified.
Process finished.
Use "cf dmol -i 51daf4c2-93f4-11ec-96f6-eeee0a825c8c" to download the logs of the process.
user: MyBatchNumberProject $
```

5. The result of deployment can be checked at SAP BTP Cockpit in Cloud Foundry space in Applications left side menu. Verify that next-number-example application is Started



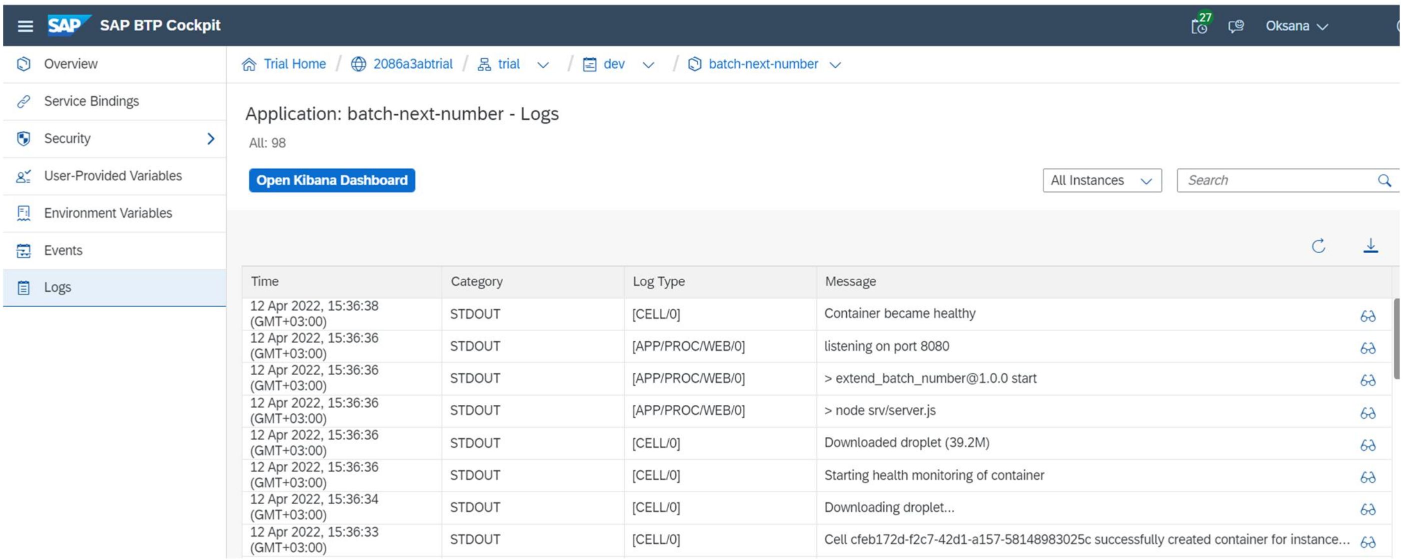
Space: dev - Applications

All: 1

Requested State	Name	Instances	Disk Quota	Memory
Started	next-number-example	1/1	1024 MB	1024 MB

Where to find logs added by application?

If you added logs in JS code, for example: `console.log(' listening on port ' + port);`
Then you should navigate to deployed application and open Logs.



Application: batch-next-number - Logs

All: 98

Time	Category	Log Type	Message
12 Apr 2022, 15:36:38 (GMT+03:00)	STDOUT	[CELL/0]	Container became healthy
12 Apr 2022, 15:36:36 (GMT+03:00)	STDOUT	[APP/PROC/WEB/0]	listening on port 8080
12 Apr 2022, 15:36:36 (GMT+03:00)	STDOUT	[APP/PROC/WEB/0]	> extend_batch_number@1.0.0 start
12 Apr 2022, 15:36:36 (GMT+03:00)	STDOUT	[APP/PROC/WEB/0]	> node srv/server.js
12 Apr 2022, 15:36:36 (GMT+03:00)	STDOUT	[CELL/0]	Downloaded droplet (39.2M)
12 Apr 2022, 15:36:36 (GMT+03:00)	STDOUT	[CELL/0]	Starting health monitoring of container
12 Apr 2022, 15:36:34 (GMT+03:00)	STDOUT	[CELL/0]	Downloading droplet...
12 Apr 2022, 15:36:33 (GMT+03:00)	STDOUT	[CELL/0]	Cell cfeb172d-f2c7-42d1-a157-58148983025c successfully created container for instance...