Deliver SAP Fiori User Experience
On the SAP Cloud Platform, on the Cloud Foundry environment

August 2019, Version 1
Legal Disclaimer

The information in this document is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. This presentation is not subject to your license agreement or any other service or subscription agreement with SAP.

SAP has no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP’s strategy and possible future developments, products, and platforms, directions, and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. This document is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP’s willful misconduct or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>4</td>
</tr>
<tr>
<td>Terminology</td>
<td>4</td>
</tr>
<tr>
<td>About SAP Cloud Platform on the Cloud Foundry environment</td>
<td>5</td>
</tr>
<tr>
<td>a. The domain model</td>
<td>5</td>
</tr>
<tr>
<td>b. Tools and services</td>
<td>6</td>
</tr>
<tr>
<td>ARCHITECTURE</td>
<td>8</td>
</tr>
<tr>
<td>Design time</td>
<td>9</td>
</tr>
<tr>
<td>Runtime</td>
<td>9</td>
</tr>
<tr>
<td>Application architecture</td>
<td>9</td>
</tr>
<tr>
<td>SETTING UP YOUR SAP CLOUD PLATFORM ENVIRONMENT</td>
<td>12</td>
</tr>
<tr>
<td>Setting up your SAP Cloud Platform subaccount for development</td>
<td>12</td>
</tr>
<tr>
<td>Setting connectivity to your on-premise backend system (optional)</td>
<td>14</td>
</tr>
<tr>
<td>Configuring the Portal service</td>
<td>15</td>
</tr>
<tr>
<td>Setting up your dev-test-prod environment</td>
<td>15</td>
</tr>
<tr>
<td>DESIGN AND DELIVER YOUR PORTAL SITE</td>
<td>17</td>
</tr>
<tr>
<td>Deliver Portal sites using the Portal service</td>
<td>17</td>
</tr>
<tr>
<td>a. Create your Portal site</td>
<td>17</td>
</tr>
<tr>
<td>b. Designing your site pages (optional)</td>
<td>17</td>
</tr>
<tr>
<td>c. Designing your site menu (optional)</td>
<td>17</td>
</tr>
<tr>
<td>d. Integrate apps and manage their data model</td>
<td>17</td>
</tr>
<tr>
<td>Getting started with delivering Portal sites</td>
<td>19</td>
</tr>
<tr>
<td>DEVELOP AND DEPLOY YOUR SAP FIORI CONTENT AND BEST PRACTICES</td>
<td>20</td>
</tr>
<tr>
<td>Develop Fiori apps and expose them to be consumed by the Portal service</td>
<td>20</td>
</tr>
<tr>
<td>a. Develop SAP Fiori apps in SAP Web IDE</td>
<td>20</td>
</tr>
<tr>
<td>b. Expose your SAP Fiori apps, to be consumed by the Portal service</td>
<td>21</td>
</tr>
<tr>
<td>c. Build and Deploy</td>
<td>22</td>
</tr>
<tr>
<td>Getting started with developing and deploying your SAP Fiori content</td>
<td>22</td>
</tr>
<tr>
<td>Best practices for developing UIs on SAP Web IDE</td>
<td>22</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>23</td>
</tr>
<tr>
<td>Efficient development cycle with deployment pipeline or CI/CD pipeline</td>
<td>23</td>
</tr>
<tr>
<td>TIPS FOR MIGRATING YOUR SAP FIORI SITES FROM THE SAP CLOUD PLATFORM NEO</td>
<td>24</td>
</tr>
<tr>
<td>ENVIRONMENT TO THE CLOUD FOUNDRY ENVIRONMENT</td>
<td>24</td>
</tr>
<tr>
<td>Edit an MTA application to use the HTML5 Application Repository</td>
<td>24</td>
</tr>
<tr>
<td>Convert your Neo application to run on the Cloud Foundry environment</td>
<td>24</td>
</tr>
<tr>
<td>FURTHER READING</td>
<td>24</td>
</tr>
</tbody>
</table>
INTRODUCTION
This document provides you with guidelines and best practices for delivering an SAP Fiori user experience on the SAP Cloud Platform on the Cloud Foundry environment.

The document is mainly targeted for SAP Cloud Platform administrators, Portal administrators, and developers.

At the end of this document, you can find tips for migrating an existing SAP Fiori user experience that currently run on the Neo environment, to the Cloud Foundry environment.

Terminology
Before you continue reading this document, you may find it helpful to become familiar with these concepts in SAP Cloud Platform on the Cloud Foundry environment.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Cloud Platform</td>
<td>SAP Cloud Platform is a Platform-as-a-Service for creating new applications or extending existing applications in a secure cloud computing environment managed by SAP. Using a set of end-to-end services, capabilities, and tools, you can build, extend, and integrate business applications in the cloud.</td>
</tr>
<tr>
<td>Cloud Foundry</td>
<td>Cloud Foundry is an open source, industry standard Platform as a Service (PaaS) technology for developing and deploying cloud applications. It is designed to run on a variety of infrastructures (Infrastructure as a Service - IaaS), such as Amazon Web Services, Google Cloud Platform, and Microsoft Azure. You can read more about Cloud Foundry here: <a href="https://docs.cloudfoundry.org/">https://docs.cloudfoundry.org/</a></td>
</tr>
<tr>
<td>Cloud Foundry Command Line Interface (CLI)</td>
<td>A tool you can use to deploy and manage your applications in the Cloud Foundry environment.</td>
</tr>
<tr>
<td>Entitlement</td>
<td>Your right to provision and consume a resource.</td>
</tr>
<tr>
<td></td>
<td>When you purchase an enterprise account, you are entitled to use a specific set of resources, such as the amount of memory that can be allocated to your applications.</td>
</tr>
<tr>
<td>Quota</td>
<td>A numeric quantity that defines the maximum allowed consumption of a specific technical asset/resource.</td>
</tr>
<tr>
<td></td>
<td>When you subscribe to an SAP Cloud Platform contract, you will receive quotas of dedicated resources.</td>
</tr>
<tr>
<td>XSUAA (User Account and Authentication)</td>
<td>A Cloud Foundry service that is used for SAP Cloud Platform user management (like cockpit users). It owns the user accounts and authentication sources and supports standard protocols such as SAML and LDAP to provide SSO and delegated authorization to Web applications.</td>
</tr>
<tr>
<td>Multi Target Application (MTA)</td>
<td>An application that is comprised of different modules representing the data, business logic, and the UI of the application. An MTA is logically a single application, consisting of multiple related and interdependent modules, that can be written using different technologies or programming paradigms and designed to run on different target runtime environments, with a single, consistent lifecycle.</td>
</tr>
<tr>
<td></td>
<td>When developing SAP Fiori apps for the Cloud Foundry environment, developers create an MTA in SAP Web IDE. Each SAP Fiori app is developed as a UI module of the MTA.</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTML5 Application Repository</td>
<td>A Cloud Platform service that provides a central storage for HTML5 applications static content and enables storing and serving this content in an optimized manner. You must use the HTML5 application repository service when you develop Fiori apps that you want to integrate with the Portal service.</td>
</tr>
<tr>
<td>Application Router (appRouter)</td>
<td>The entry point for an application at runtime. The App Router serves static content by calling the HTML5 application repository, routes between the different components of an application, handles authentication using the XSUAA service, and is responsible for session management.</td>
</tr>
</tbody>
</table>

About SAP Cloud Platform on the Cloud Foundry environment

a. The domain model

The SAP Cloud Platform on Cloud Foundry model follows the domain model of the Cloud Foundry technology.

SAP Cloud Platform has the following generic entities:

- **Global Account** - the realization of the commercial contract with SAP. It represents the scope of the functionality and the level of support based on a customer or partner’s entitlement to resources and services. A global account serves as an entry point for managing the resources, landscape, and entitlements for your departments and projects.
- **Region** – represents the physical location of a data center, where applications, data, or services are hosted. A global account is associated with one datacenter.

A global account can consist of multiple subaccounts. You can have Cloud Foundry subaccounts or Neo subaccounts

- **Subaccount** – Lets you structure a global account according to the requirements of your organization and projects, with regards to members, authorizations, and entitlements. Each subaccount resides in a region.

  For example, if you want to set up different environments for development, testing, and productive usage, you can create a subaccount for each of these scenarios in your global account. You can also create subaccounts for different development teams or departments in your organizations.

A Cloud Foundry subaccount can be used in a pure subscription model. The subscription model allows you to subscribe to available content (delivered by SAP or partners) and consume it, using the application subscriptions. You can learn more about subscribing to available content in [this help page](https://help.sap.com). You can also enable development of your own apps in your subaccount (by setting Enable-Cloud-Foundry flag on your subaccount). When you enable development, additional components are created in your subaccount to provide a shared location for your application development and deployment.
The diagram below shows a Cloud Foundry subaccount structure and relationships

Here is an explanation of each entity:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Subscriptions</td>
<td>Enables you to subscribe to business applications provided by SAP or by third party providers who have registered their apps. As this document focus on Fiori development, applications subscriptions are only mentioned here. You can read more about this capability <a href="#">here</a>.</td>
</tr>
<tr>
<td>Organization (org)</td>
<td>This is a development account with a resource quota plan, applications, services availability, and custom domains. An org is associated with one subaccount and is further subdivided into spaces.</td>
</tr>
<tr>
<td>Space</td>
<td>Provides developers with access to a shared location for application development, deployment, and maintenance.</td>
</tr>
</tbody>
</table>

**b. Tools and services**

SAP Cloud Platform provides you with a rich set of capabilities and tools.

Below you can find the main SAP Cloud Platform tools and services to implement Fiori user experience in the cloud:

SAP Cloud Platform cockpit – An administration interface that provides access to a number of functions for configuring and managing applications, services, and subaccounts. SAP Cloud Platform cockpit is the central point of entry to manage your accounts, and applications.

SAP Web IDE – A development environment that enables you to easily develop, test, build, deploy, and extend apps for business users. Use SAP Web IDE to create applications rapidly and deliver an outstanding user experience. Developers can extend or build SAP Fiori apps, create new SaaS solutions, extend 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 node.js and Java programming language and technologies. Since SAP Web IDE runs on SAP Cloud Platform, it needs no installation and allows you to easily leverage other services that run on the platform—such as Git version control, mobile services, IoT services, and more.

**SAP Cloud Platform Portal** – Allows you to deliver role based, business sites with rich user experience including an SAP Fiori launchpad page, for users.

The Portal service on the Cloud Foundry environment differs from the Portal service on the Neo environment in several ways.

What’s new in the Cloud Foundry environment?

- **A unified way to consume content**: Applications are planned to be consumed in a uniform way by the Portal service, whether it be from SAP business applications, your own developed apps, or a third party. This means that all integrated applications (including those developed in your company) should include the common data model (CDM) configuration. The CDM is a configuration file and is a new concept available only on the Cloud Foundry environment.

- **A new Launchpad module**: A built-in launchpad module in SAP Web IDE allows developers to add a Common Data Model configuration to their developed Fiori apps and expose them to the Portal service. This launchpad site is configured on a Space level (see the [architecture diagram](#) for a graphical representation of the solution). If your portal site does not require integration of additional SAP Cloud Platform services and consumption of content from other SAP solutions and SAP backend systems, the launchpad site created using SAP Web IDE can also be used directly by end users. This is not the common use case but it may be beneficial for specific scenarios, where developers create the launchpad site themselves, and would like to use it as part of the CI/CD process.

- **New admin tool**: The Portal service includes a new tool called the **Site Editor**. The Site Editor is where you design your site by creating portal pages (including a launchpad page), integrate apps (including assigning groups, catalogs, and roles to these apps), adding web content and creating a site menu. The Site Editor replaces the SAP Fiori Configuration Cockpit (that was used to create SAP Fiori launchpad sites), and the Site Designer (that was used to create freestyle sites with or without an SAP Fiori launchpad page) from Neo, and provides an improved user experience with richer capabilities.
ARCHITECTURE

SAP Cloud Platform is where you develop and run SAP Fiori apps in the cloud. These apps can run fully on the cloud or they can consume data from a backend system such as SAP Business Suite or SAP S/4HANA.

The following diagram illustrates a combined architectural view of the design time and runtime environments for delivering SAP Fiori UX on SAP Cloud Platform on the Cloud Foundry environment.

To obtain data from an SAP backend system, you should configure a secured connection for both design time and runtime environments.

The following components can be used to achieve this:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Cloud Platform Connectivity Service</td>
<td>Allows you to establish secure, reliable, and easy-to-consume access between your cloud applications and on-premise systems running on isolated networks. The service provides a connectivity proxy that you can use to access on-premise resources.</td>
</tr>
<tr>
<td>SAP Cloud Platform Destination Service</td>
<td>Allows cloud applications running on SAP Cloud Platform to retrieve technical information that is required to consume existing APIs and data from remote services that run on the internet or on-premise. Using the Destination Service, you can retrieve and use the technical information about the target resource (destination) that you need to connect your application to a remote service or system.</td>
</tr>
<tr>
<td>SAP Cloud Connector</td>
<td>Can be installed in an on-premise environment to establish a secure connection to an SAP Cloud Platform account and a protected on-premise network. Cloud Connector is used to enable cloud applications to access and extend on-premise systems.</td>
</tr>
<tr>
<td>Identity Provider</td>
<td>Use your corporate identity provider or SAP Identity Provider service, to allow end users to sign in to the Portal as authenticated users and provide single sign-on to backend business systems.</td>
</tr>
</tbody>
</table>
Design time

Design time is comprised of two parts: developing your own apps with the Common Data Model configuration and designing a Portal site using the admin tools.

a. Developing your own apps and Common Data Model:

SAP Web IDE is the recommended environment for developing SAP Fiori apps. Since SAP Web IDE service is available on the SAP Cloud Platform Neo environment, to enable developing apps that consume data from an SAP business backend, you should configure a secured connection between the SAP Cloud Platform on Neo environment, to your on-premise/Internet backend (including user authentication). Use the SAP Cloud Platform connectivity service on the Neo environment to access backend systems from SAP Web IDE. Use your corporate identity provider or the Identity Authentication service to configure user authentication.

To expose your developed apps to the Portal service, add a launchpad module to your development project. The launchpad module is the Common Data Model (CDM) of your developed apps. The CDM is a uniform configuration that allows content to be consumed by the Portal service.

You should also configure a connection between SAP Web IDE and your SAP Cloud Platform on Cloud Foundry environment for deployment.

b. Design your Portal site using the admin tools:

To design your Portal site, access the Portal service on your SAP Cloud Platform on the Cloud Foundry environment subaccount. The Portal service enables portal administrators to create portal sites with apps and web content, and enrich your Portal site experience with additional SAP Cloud Platform services. In addition, you can subscribe to available content that is delivered by SAP, or an SAP partner, and add such apps to your portal site.

Runtime

To allow end users to access Portal sites and enjoy a single sign-on experience, you need to define a secured connection between SAP Cloud Platform on the Cloud Foundry environment and your on-premise/Internet backend. Use SAP Cloud Platform on the Cloud Foundry environment connectivity service, destination service, and the SAP Cloud Connector to access on-premise backend systems from the Portal service. Use your corporate identity provider or the Identity Provider service to configure user authentication.

End users can then access your Portal site through their mobile or desktop devices. Once they are authenticated through the identity provider, they can securely access apps and content.

Note:

a. Apps that are added to a Portal site (on the subaccount level), run in an isolated mode. This means that these apps are launched with the configuration that is defined in the Launchpad module (on the app/space level). An advantage of this architecture is that it allows you to add apps with multiple UI5 versions to a single Portal Site.

b. Apps can get data from backend systems using the OData protocol, that can bring data from cloud, on-premise, or third-party solutions.

Application architecture

In this section we will drill down from the application level, to its building blocks, and describe the design time and runtime of a single application in more details.

As SAP Web IDE service is available on the Neo environment, developers should access it from a Neo subaccount, and configure to deploy applications to the required SAP Cloud Platform on Cloud Foundry Space.
In SAP Cloud Platform on Cloud Foundry, developers create Multi Target Applications (MTA) and deploy them to SAP Cloud Platform (instead of creating individual apps as was done in SAP Cloud Platform on the Neo environment). In the following diagram, you can see the main application components of a Multi Target Application in SAP Web IDE.

A typical MTA contains one or more UI modules, a launchpad module, and in some cases a Java module. In addition, an MTA contains an App Router and UI Deployer modules, that are automatically created with the MTA template. An application can use services provided by SAP Cloud Platform such as the connectivity service, the ABAP environment service, and the HTML5 Application Repository service.

During the deployment of the MTA application, an App Router application is created in the customer's Cloud Foundry space, along with instances of all the SAP services used by the application. After the application is deployed, you can see it in the SAP Cloud Platform cockpit by selecting Applications in the Space. Each MTA has a URL property that points to the App Router.
As seen in the diagram below, during runtime, the App Router serves as the entry point of an application and is responsible for routing end user calls to the relevant service instances like destination service, XSUAA, HTML5 app repository and more.

The App Router calls the XSUAA service to verify authorizations, collects static HTML5 data from the HTML5 application repository, and then calls the SAP Fiori app. The SAP Fiori app can then call the destination service and connectivity service to retrieve data from backend systems.
SETTING UP YOUR SAP CLOUD PLATFORM ENVIRONMENT

This topic is mainly targeted at SAP Cloud Platform administrators who wish to setup their company's subaccount.

By default, your subaccount is configured to subscribe to content. If you want to develop your own content, you need to configure the subaccount for development.

Setting up your SAP Cloud Platform subaccount for development

Note: If you are using a trial account, you only need to perform step 5 below.

For a step by step guide with screenshots on the initial SAP Cloud Platform on Cloud Foundry environment setup for development (with steps 1 and 2 described below) see this tutorial

1. Enable Cloud Foundry in your subaccount

   To enable development of your own apps, you first need to enable Cloud Foundry in your subaccount.

   Once you enable Cloud Foundry, the following occurs:

   • An organization is created for you.
   • A Spaces menu item is added to your subaccount, enabling you to create different spaces for your developers to develop applications and deploy them.

2. Create a space for development

   After you've enabled Cloud Foundry, create a space from the Spaces menu that has been added to the side menu of the cockpit.

   As explained above, a space provides developers with access to a shared location for application development and deployment. Since each space is a shared workspace for developers, you should ensure the right permissions are granted for developers to access the relevant space.

3. Set entitlements

   When you purchase an enterprise account (without the consumption-based commercial model), you are entitled to use a specific set of resources (on the level of your global account). For each resource, you have a quota (a numeric quantity that defines the maximum allowed consumption of a resource).

   Entitlements and quotas purchased are reflected in your global account. You must distribute quotas from your global account to your subaccounts by adding entitlements and quotas to your subaccounts. To view and edit entitlements and assigned quotas of resources to your subaccounts, go to your global account and select Entitlements from the left navigation panel. Learn more about this from the following documentation Assign Entitlements to Subaccounts. Note: You will have some quota for HTML5 Applications in your global account when you purchase Application Runtime quota.

   To prepare your Subaccount for development, assign entitlements to the following services:

   • **Application Runtime** – provides VM Memory GBs runtime capacity in the Cloud Foundry environment.

     Application Runtime is used to run Cloud Foundry applications (backend apps, Application Routers).
You can use the following information to decide what Application Runtime quota you need:

1. The required memory to run your App Router application. For each App Router application, you can see the assigned quota in the applications page of the Space in the SAP Cloud Platform Cockpit. By default, this quota is set to 256MB.

2. The planned number of sessions (and their size) your App Router should support.

3. The required VM Memory to run your developed backend applications.

For example, you can assign 256MB Application Runtime for a small session (with mainly HTML5 applications) to run up to 5000 concurrent login sessions.

- **HTML5 Applications with the app-host plan** – provides storage in the HTML5 Application Repository.

  The HTML5 Application Repository is used for central storage of HTML5 applications and enables storing and serving this content in an optimized manner.

  You can use the following information to decide what HTML5 Application with the app-host plan quota you need:

  1. The size of the HTML5 applications.
  2. The number of HTML5 applications.

  For example, if you have 10 HTML5 applications, and the size of each application is 500KB, you will need to assign about 5MB of HTML5 Applications with the app-host plan quota. A best practice would include additional quota for future development.

- **Portal** – provides an elastic quota for developing SAP Fiori launchpad sites using SAP Web IDE (or exposing your developed apps to the Portal service).

  You should assign the Portal entitlement per subaccount and assign the standard service plan to it.

  Note: this entitlement is related to creating a Launchpad module using SAP Web IDE only, and is not related to the ability to create Portal sites using the Portal service.

You may also assign the following entitlements according to your application requirements:

- **Connectivity** – to establish connections to on-premise systems through the Cloud Connector.
- **Destination** – to provide your cloud applications with access to destination and certificate configurations in a secure and reliable way.
- **XSUAA** – to manage your business user authorization model consisting of roles, groups and role collections, and assigning them to users.

To avoid overuse of service resources in the space of a subaccount, you can create space quota plans. To create a space quota plan, navigate to your subaccount, and select Quota Plans from the left navigation panel.

Read more on Assigning Quotas to Spaces from the SAP Cloud Platform onboarding documentation. For more information on space quota plans, you can view the following page from the Cloud Foundry documentation https://docs.cloudfoundry.org/adminguide/quota-plans.html. Make sure your Space can consume sufficient quota, as your apps will be deployed to your Space.
4. Define authorizations with members and roles

Follow this step only if you need to manage other users.

A member is a user who is assigned to an SAP Cloud Platform entity like global account or subaccount. Roles determine which functions users can view and access, and which actions they can initiate.

A member automatically has the permissions required to use the SAP Cloud Platform functionality within the scope of the respective entity and as permitted by their account member roles.

You can manage users at the global account level by adding them as members to your global account. You can also manage users at subaccount, org, and space level. For more information, see Security Administrators in Your Subaccount, Adding Organization Members, and Adding Space Members on the SAP Cloud Platform documentation, to learn about defining members and roles on the subaccount, org, and space level.

You should add developers as members to the relevant Org or Space with the relevant roles.

5. Enable SAP Web IDE Full Stack service on the Neo environment

SAP Web IDE is the recommended development environment by SAP. The SAP Web IDE service is available on the Neo environment and hence, you should enable this service on that environment. Developers can later configure SAP Web IDE to deploy apps to the Cloud Foundry environment.

In this document, we assume you already have a Neo subaccount. If you don’t, follow this tutorial to get started with SAP Web IDE Full-Stack.

Note: To consume data from an on-premise backend system in SAP Web IDE, you need to configure a secured connectivity from the SAP Cloud Platform for the Neo environment subaccount (that you use to work with the SAP Web IDE service), to your backend system. For more information see the SAP Cloud Platform Connectivity documentation.

Setting connectivity to your on-premise backend system (optional)

Only do this section if you wish to consume data from an on-premise system.

1. Set a secured connectivity from your SAP Cloud Platform, Cloud Foundry subaccount to your on-premise system

You can follow this blog post for a step by step guide for setting connectivity between SAP Cloud Platform on the Cloud Foundry environment and your on-premise backend system. In this way you can consume data coming from an on-premise system in a Cloud Foundry based application. For more details, see SAP Cloud Platform Connectivity documentation.

2. Set a secured connectivity from your SAP Cloud Platform, Neo subaccount to your on-premise system

If you want to consume content from your backend system during development, you should configure a secured connection between your SAP Cloud Platform subaccount on the Neo environment, where SAP Web IDE service is available, and your on-premise backend system.
Configuring the Portal service

To enable portal administrators to create portal sites, you should subscribe to the Portal service from your subaccount and define your portal administration users.

1. **Subscribe to the Portal service** – do this from the Subscriptions section in your subaccount.
2. **Set your Portal admin users** – to define your Portal admin users, configure a Role Collection and add the Super_Admin role to it. The Super_Admin role enables users to perform all the administrative tasks for a Portal site. You should assign the created Role Collection to your Portal admin users.

For a step by step guide of how to configure the Portal service follow [this tutorial](#). You can also follow [this short video](#). For more information, visit the [Onboarding](#) section of the Portal administrator guide.

Note: The Super_Admin role needs to be explicitly assigned to any user who wishes to perform Portal administrative tasks.

Setting up your dev-test-prod environment

The following steps describe how to set up a three-tier system landscape in the SAP Cloud Platform on the Cloud Foundry environment to develop, test, and host productive applications.

This process is performed by system administrators and technical consultants.

In general, we recommend that you create different subaccounts for a staged development environment, as shown below. This allows for dedicated user management between the different stages. You can then create dedicated spaces for applications or projects within these subaccounts (if you do not need a dedicated user management for these applications and projects).

![Diagram](#)

To achieve the landscape described in the illustration, go to your global account and create three subaccounts for Dev, Test and Prod. In each of the new subaccounts, enable the Cloud Foundry environment, as explained in [Item 1 of the setting-your-subaccount-for-development section above](#). You will be prompted to create a Cloud Foundry organization.

Once you're done, your cockpit should look like this:
Now you can create new spaces in your subaccounts and assign the relevant roles for them.

You can create multiple spaces in each subaccount. For example, one for each application, business unit, development team etc.
DESIGN AND DELIVER YOUR PORTAL SITE

Using the SAP Cloud Platform cockpit and the Portal service on SAP Cloud Platform on the Cloud Foundry environment, Portal administrators can design and deliver Portal sites for users.

You can integrate available SAP content to your Portal site. You can also integrate your own developed content or Fiori apps to your Portal site.

Note: In the section Getting started with delivering Portal sites below, you can find links to tutorials that provide step-by-step instructions on how to deliver your first Portal site on SAP Cloud Platform on the Cloud Foundry environment.

Deliver Portal sites using the Portal service

This document focuses on delivering portal sites with your own developed SAP Fiori apps and hence contains the following topics:

a. Create your Portal site
b. Design your site pages
c. Design your site menu
d. Integrate your own developed Fiori apps

a. Create your Portal site

To create and design your Portal site, you need to access the Portal service. Navigate to the Subscriptions tab of your subaccount, locate the Portal service, and open it.

In the Portal service, you can use the following tools:

1. Site Directory – where you create and manage your business sites
2. Site Editor – where you design the pages of your site, add apps to the launchpad page, add web content to other pages, and design your site menu

Create your Portal site using the Site Directory. Use the Site Editor to design its pages, integrate content, and designing your site menu, as described below.

b. Designing your site pages (optional)

By default, your new site contains an SAP Fiori launchpad page. A site can also contain various other pages with your own design that combine web content and apps.

- To add more pages to your site, navigate to the Pages editor (access from the left navigation panel of the Portal's Site Editor).
- Select each page you created to add content to it and define how it looks.
  - Design the launchpad page using the Content Manager.
  - Design other pages of your site by selecting the page you want to edit, adding web content widgets to the different page sections, and editing the web content widgets to include your own content.

c. Designing your site menu (optional)

Select the Menu Editor from the side navigation panel of the Site Editor to define your site mega menu. Your site menu can have various levels including titles and sub menus. Each menu item can point to a Portal page, an app, or a link.

d. Integrate apps and manage their data model

Before you can integrate app technologies to your Portal site, you first need to define a destination to the developed content you would like to consume from the Portal service.
Defining a destination to your developed content

To add apps to your Portal site, you first need to define a destination for each MTA App Router that contains apps you would like to consume. To obtain the URL of the MTA App Router of your own developed content, navigate to the Applications tab, in the Space where the developer deployed the required content. See image below:

Locate your MTA _appRounter application, that contains the SAP Fiori apps you would like to integrate. Click on it to view its details and copy the displayed URL. The displayed URL is the URL of the MTA application App Router.

To add the appRouter as a destination, follow this documentation link. Make sure you put the relevant properties in the newly defined destination, and also include an additional property: sap-platform with the value CF.

Now you can configure the portal to integrate with these apps

Add apps to your portal site

To integrate apps to your portal site or design your launchpad page, select the content section from the Portal service, and access the Content Manager. You can integrate apps from different app types. View the documentation for more information.

To integrate your own developed Fiori apps, click to add a new application and configure the following app settings for it:

- **Title**: your selected title
- **System**: the destination of the MTA appRounter you configured in step a.
- **SAPUI5 Component Name**: The SAPUI5 component name of the app you want to integrate. Ask a developer for this information - they can get it by accessing the code of the relevant SAPUI5 app and view its component.js file. The SAPUI5 component name is the name that appears in this file, without the .component suffix

In addition, add intent configuration for your app in the Navigation tab, and configure your app visualization in the Visualization tab. You can also translate your tile, using the Translation tab.

The launchpad page in your Portal site is the access point to the apps that you’ve added to your site. The apps are managed by assigning the following entities to them in the Portal Content Manager:

- **Catalogs**
  Assign your app to a catalog to display them in the App Finder – the App Finder is a convenient tool for end users to find all their apps in one place, and possibly customize their launchpad page to include additional apps. End users can access the App Finder from the user action menu at the top left of the launchpad screen.

- **Groups**
  Assign your app to a group so that an end user can see the app in the launchpad page.
• Roles
In addition, assign your app to a role to enable end users with this role to view and access these apps. A user can be granted permissions to access the applications of a specific role. To assign roles to end users, select the Trust Configuration under Security from the side navigation panel in the SAP Cloud Platform Cockpit. Click the SAP ID Service or your own Identity Provider, and assign the relevant Role Collection to end users. Note: when you create a role using the Portal Content Manager, a role collection with that name is created on SAP Cloud Platform.

For more information about creating a Portal site with content, read this documentation.

Getting started with delivering Portal sites

To create your first Portal site, you can follow these tutorials:

1. Prepare the Portal Environment for Creating Sites
2. Create a Portal Site
3. Add Web Content to Your Portal Site and Design its Layout
4. Integrate a URL App to Your Portal Site
5. Integrate Your Own SAP Fiori App into Your Portal Site

Note: In the next chapter you will learn how to develop SAPUI5 apps and expose them to the Portal service. You may want to follow the tutorials mentioned in the next chapter and only then implement tutorial 5 above to integrate your developed app to your Portal site.

For a complete explanation about how to design your Portal site, go to the Portal Administrator Guide.
DEVELOP AND DEPLOY YOUR SAP FIORI CONTENT AND BEST PRACTICES

This section explains how to develop SAP Fiori apps and expose them as content for the Portal service. It includes highlights for developing these apps using SAP Web IDE, creating a common data model to expose the apps to the Portal service, deployment to SAP Cloud Platform on Cloud Foundry, as well as best practices.

Note: In the section Getting started with developing and delivering your SAP Fiori content below, you can find links to tutorials that provide step-by-step instructions on how to develop your first SAP Fiori app, expose it to the Portal service, and integrate it to your Portal site.

Develop Fiori apps and expose them to be consumed by the Portal service

a. Develop SAP Fiori apps in SAP Web IDE

Using SAP Web IDE, developers can develop SAP Fiori applications for SAP Cloud Platform on the Cloud Foundry environment.

An application developed for SAP Cloud Platform on the Cloud Foundry environment, involves some concepts that are different from the SAP Cloud Platform Neo environment:

- MTA (Multi-Target Application) – an MTA application is comprised of different modules representing the data, business logic and the UI of the application. The MTA concept aims at orchestrating the deployment of all its modules to share the same development lifecycle so that all runtime dependencies are properly resolved and the application functions as expected. SAP Web IDE offers a Multi-Target application template which enables easy and fast creation of an MTA project structure in SAP Web IDE. The MTA project contains an MTA descriptor file (mta.yaml), which contains the metadata of all entities comprising an application or used by it during deployment or runtime, as well as the dependencies between them. The MTA descriptor content is updated when the project properties change or when a module is added or removed.

- App Router – The app router is a module in the MTA application and is the single point of entry to the business application. It provides the applications routing functionality.

  SAP Web IDE creates an app router module during the project creation of an MTA project that uses the HTML5 Application Repository. The App Router is used to handle the interaction with the HTML5 Application Repository and all the related technical and backend services such as authentication, destinations, and HTML5 static content requests.

  Each MTA project will contain a single App Router that serves all the HTML5 modules in the project.

- HTML5 application repository – the HTML5 application repository service is an SAP Cloud Platform service that enables central storage of HTML5 applications on the SAP Cloud Platform on the Cloud Foundry environment and allows application developers to manage the lifecycle of their HTML5 applications.

  At runtime, it enables applications to access static content in a secure and efficient manner.

  When creating a new MTA project from SAP Web IDE, developers should select the option to use the HTML5 application repository in their application.

The basic development process to create an SAP Fiori application for SAP Cloud Platform on the Cloud Foundry environment, in SAP Web IDE is as follows:

- Create an MTA Project - the "Multi Target Application" project template, helps you to quickly create a project structure for developing an SAP Fiori application for the Cloud Foundry environment.
- Create an HTML5 module – an HTML5 module is a collection of related HTML5 files that implement the UI of the application.
• Run on the Cloud Foundry environment – you can then preview the UI of your application on the Cloud Foundry environment to test it during development.

Note that you must configure the associated Cloud Foundry space, from SAP Web IDE to deploy your apps to the relevant location. You can configure your Cloud Foundry space from the SAP Web IDE general settings, Cloud Foundry tab, or from the preferences of your MTA app.

In the Cloud Foundry development model, the SAP Fiori application settings is part of the Fiori app development. Edit the app "manifest.json" file located under the app's webapp folder and configure intents, app title, icon, and additional settings.

Follow this step by step tutorial to develop your first "Hello World" SAP Fiori app, including SAP Fiori application settings.

b. Expose your SAP Fiori apps, to be consumed by the Portal service

To expose your SAP Fiori apps to become available to the Portal service, you must configure a common Data Model (CDM) for your apps. This common data model is the Portal protocol to consume content.

This is done by adding a launchpad module to your Multi Target Application (MTA) project. After you add the launchpad module, you can edit its CommonDataModel.json file and add your apps to a group and/or catalog.

To add the launchpad module your MTA project, you should follow these steps:
1. Enable the SAP Fiori Launchpad Site extension in your SAP Web IDE environment.
2. Add the launchpad module to your MTA project.
3. Configure your launchpad module by editing the CommonDataModel.json file and adding any app you would like to expose to the Portal service to groups and/or catalogs. Note that you can edit this file in either the Launchpad UI editor or the JSON code editor.

You can follow this tutorial for a step by step guide on how to expose your apps to the Portal service.

Note: The Launchpad module created using SAP Web IDE, can also be used directly by end users for simple scenarios that include your own developed apps only. In this case, configure the app authorizations on the app level by configuring app scopes and role templates. Read the Developing Security Artifacts help page for more information.
c. **Build and Deploy**

Once you’ve completed the development of your MTA application, you need to build it. The build process packages all the project modules for deployment in a file called MTAR. This file will be created in your project folder.

To deploy the application, simply select the relevant MTAR file from your project folder and deploy it to the desired Cloud Foundry space.

See how to build and deploy your application in step 6 and 7 of this tutorial.

When looking in the Cloud Platform cockpit in the applications tab, you’ll find the URL of your application’s AppRouter. As explained in previous chapters, the AppRouter is the entry point to your application, in order to launch a specific HTML5 application you need to add its name to the URL, as explained in step 8 of this tutorial.

Note: SAP Web IDE uses Grunt for building your project and supports the best practice configurations for SAPUI5 applications using grunt-openui5, which is a set of Grunt plugins provided by SAP that improves application performance considerably in the productive environment.

**Getting started with developing and deploying your SAP Fiori content**

To develop your first SAP Fiori app, expose it to the Portal service and add it to your Portal site, you can follow these tutorials:

1. **Develop Your First SAP Fiori App including SAP Fiori Application Settings**
2. **Add an SAP Fiori Launchpad Module to Your Development Project and Expose Your App to the Portal Service**
3. **Integrate Your Own SAP Fiori App to Your Portal Site**

**Best practices for developing UIs on SAP Web IDE**

- When developing a new application using an MTA template, you must choose whether to use the HTML5 application repository or not. Since the Portal service requires the HTML5 application repository, we recommend using it in all your SAP Fiori applications.

- In some cases, during the deployment of your application, new service instances (e.g. destination, XSUAA) are created automatically in your space. These service instances should be deleted to release resources when the application is deleted or not in use. you can delete service instances from the resource manager in SAP Web IDE

- Since SAP Web IDE is on the Neo environment and your application is deployed to SAP Cloud Platform on the Cloud Foundry environment, when using a data source that is external to SAP Cloud Platform such as S/4 cloud or an internet service, you should create destinations both in the Neo and in the Cloud Foundry environments. The destination in the Neo environment will be used in design time for the discovery of the data source, and the destination in the Cloud Foundry environment will be used to retrieve the data at runtime.
Troubleshooting

UI-development with SAP Web IDE supports debugging with the standard browser-based development tools. You can create a run configuration to include the Support Assistant tool, which checks whether your application is built according to the best practices for building SAPUI5 apps. You may open the SAPUI5 Diagnostics to help in debugging by pressing <Ctrl>+<Alt>+<Shift>+<S>

See also the UI5 Inspector Chrome extension.

Efficient development cycle with deployment pipeline or CI/CD pipeline

To automate the deployment, you must download the Cloud Foundry CLI. You will then deploy your apps through an API. In addition, Install the CF MTA plugin relevant to your operating system, for performing operations on MTAs, such as deploying, removing, viewing, and more.

Link to download: tools.hana.ondemand.com

You can apply this to your CI/CD pipeline.

You can learn more about CI/CD practices in this link. You can also follow this tutorial.
TIPS FOR MIGRATING YOUR SAP FIORI SITES FROM THE SAP CLOUD PLATFORM NEO ENVIRONMENT TO THE CLOUD FOUNDRY ENVIRONMENT

This section contains some tips that will help you to migrate apps that were developed on the SAP Cloud Platform on the Neo environment to the Cloud Foundry environment. After you migrate your apps, you should recreate your portal site on the Cloud Foundry environment.

**Edit an MTA application to use the HTML5 Application Repository**

SAP Fiori launchpad and SAP Cloud Platform Portal sites running on SAP Cloud Platform on the Cloud Foundry environment store static resources in the HTML5 application repository. Applications that were previously created need to be adjusted to use the HTML5 application repository for storing the static content of the HTML5 applications.

Follow this step by step guide to update a Multi-Target Application (MTA) that is not using the HTML5 application repository to an MTA application that stores the HTML5 application static content on the HTML5 application repository in the SAP Cloud Platform on the Cloud Foundry environment.

**Convert your Neo application to run on the Cloud Foundry environment**

As already described in this document, the structure of an application developed for Cloud Foundry is different from an application developed for the Neo environment.

In the Neo environment Fiori apps have a flat structure whereas in Cloud Foundry the Fiori application is a module inside an MTA structure.

If you want your Fiori Neo application to run on Cloud Foundry, you need to create a new MTA project, put your Neo Fiori project as a UI5 module in your new MTA project and configure some files in the new project to reference the new UI5 module.

**FURTHER READING**

Portal:
- Fiori launchpad - Deployment Options and Recommendations
- SAP Cloud Platform Portal roadmap
- SAP Cloud Portal on Cloud Foundry help

SAPUI5:
- SAPUI5 Documentation

SAP Web IDE:
- SAP Web IDE Site
- SAP Web IDE Documentation

Cloud Foundry:
- Cloud Foundry documentation

Tutorials:
This document contains a list of tutorials, you can follow this mission to complete them