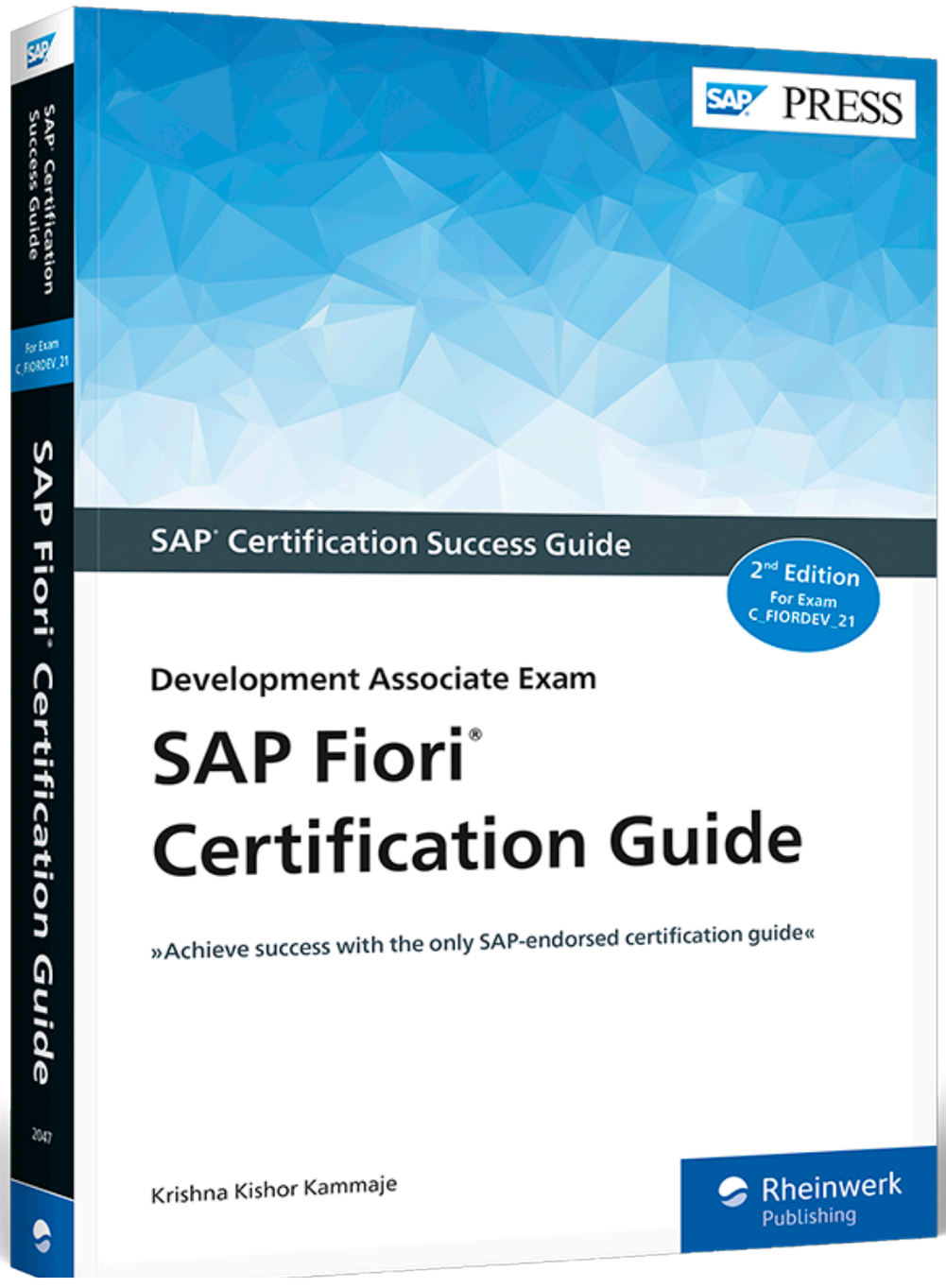


Browse the Book

This chapter discusses the architecture of SAP Fiori apps, the SAP Fiori launchpad, and how to change the look and feel of SAP Fiori based on a corporate theme.

Krishna Kishor Kammaje

SAP Fiori Certification Guide: Development Associate Exam

429 Pages, 2020, $79.95
ISBN 978-1-4932-2047-2

www.sap-press.com/5221

Chapter 2

SAP Fiori Architecture Overview

2

Techniques You'll Master:

- Understand the architecture of SAP Fiori
- Explain the data flow in SAP Fiori app types
- Choose the right deployment option for SAP Gateway
- Configure SAP Fiori launchpad
- Brand SAP Fiori with your corporate theme

In this chapter, we'll start by discussing the architectures of various types of SAP Fiori apps. We'll deep dive into these architectures by exploring the data flow in each of these types. Next, we'll see how to choose the right deployment option for your landscape by discussing possible options and the advantages of each option. We'll explore how tiles are configured within SAP Fiori launchpad while covering catalogs, groups, and roles. We'll end the chapter by showing you how to change colors and logos and come up with a custom theme for your SAP Fiori launchpad that represents your enterprise's corporate theme.

Real-World Scenarios

You want to debug an issue in your SAP Fiori app. To pinpoint the issue, you need to find the data flow in various parts of the app so that you can set breakpoints and verify that the app behaves as expected.

You're setting up an SAP Fiori landscape in your company. You want to know the possible options and compare them so that you can come up with an optimal configuration for the SAP Gateway.

You have a set of corporate colors and logos that represents your brand and company and is used by all internal and external tools. You want to use these in a theme from SAP Fiori so that SAP Fiori is also consistent with rest of the applications in your enterprise.

Objectives of This Portion of the Test

The objectives of this portion of the SAP Fiori Certification Test are as follows:

- Understand SAP Fiori architecture and data flow for all three application types.
- Understand deployment options for SAP Gateway and your skills in choosing the right deployment option.
- Demonstrate skills in creating a custom theme per your corporate standards and how to apply it to SAP Fiori launchpad.

Key Concepts Refresher

In this section, we'll cover the architecture of SAP Fiori apps for various application types discussed in Chapter 1. A typical SAP Fiori landscape has many components, which results in multiple deployment options. We'll discuss the common options and architectures here.

Generic Architecture

Figure 2.1 shows the general architecture components involved with SAP Fiori.

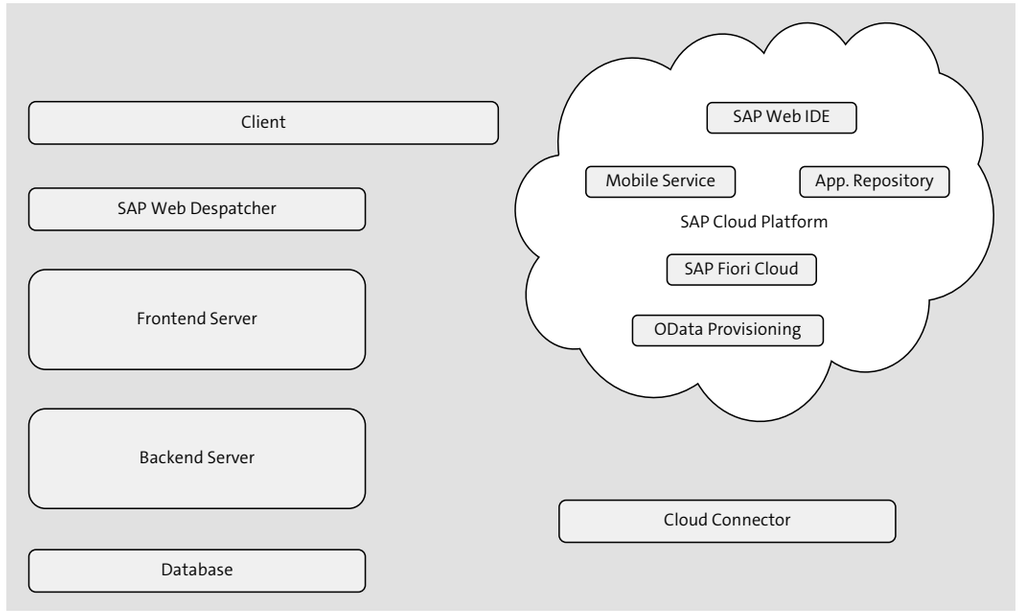


Figure 2.1 Components in the SAP Fiori Architecture

Client

The client here represents a desktop browser, a mobile browser, or an SAP Fiori Client. In an on-premise scenario, clients access SAP Fiori using SAP Fiori launchpad by clicking the link pointing to an SAP Web Dispatcher or to a frontend server.

SAP Web Dispatcher

SAP Web Dispatcher is a reverse proxy product, but it provides more features than a simple reverse proxy. It helps SAP Fiori apps fetch data from more than one source. Without a reverse proxy, such calls from the browser would get blocked due to the same-origin policy security concept implemented by browsers. In addition, it can also be a switch that blocks or enables access to internal resources.

Another important feature of SAP Web Dispatcher is load balancing in a web scenario. If there is more than one SAP NetWeaver server (ABAP or Java), SAP Web Dispatcher can act as an effective load balancer. Due to these features, SAP recommends SAP Web Dispatcher for all web scenarios.

Frontend Server

This is the SAP Gateway server that hosts the OData service, which is required for the SAP Fiori apps to communicate with the business data. In addition, the frontend server hosts the SAP Fiori launchpad as well as all the SAP Fiori apps as Business Server Pages (BSP) applications.

Backend Server

This is the main system where business data and business logic lies. This is also the server where an OData service is implemhented.

Database

This is the database that is connected to the backend server and stores all the important business-critical data. This database can be any of the SAP-supported databases or even SAP HANA. In SAP HANA, this layer can also work as an application server.

SAP Cloud Platform

SAP Cloud Platform is SAP's platform-as-a-service (PaaS) offering providing tools and technologies to fast-track application development.

SAP Fiori Cloud

This is a service that allows you to host your SAP Fiori launchpad and applications on the cloud. For business data, it connects to your on-premise backend system. Along with OData provisioning, this can replace the frontend server from your SAP Fiori architecture.

OData Provisioning

This is one of the services of SAP Cloud Platform, where you can register and expose OData services that were developed on your backend system.

SAP Mobile Service

This is a cloud version of the SAP Mobile Platform on-premise solution.

SAP Web IDE

This is a cloud-based (SAP Cloud Platform) development environment for developing and extending the SAP Fiori apps. Chapter 4 provides a deep dive into SAP Web IDE.

App Repository

This is a repository available on the SAP Cloud Platform that stores the user interface (UI) resources of each SAP Fiori app that is exposed to SAP Fiori Cloud.

SAP Cloud Platform Cloud Connector

This is a lightweight server that is part of the on-premise server infrastructure and exposes the on-premise system to the SAP Cloud Platform.

SAP Fiori On-Premise

SAP Fiori was initially released as an on-premise product on the SAP Business Suite to improve the user experience (UX) of selected scenarios by creating targeted SAP Fiori apps. As discussed in Chapter 1, there are three types of SAP Fiori apps, and each of them slightly differs in architecture. This architecture gradually evolved in SAP S/4HANA and follows a single architecture for all three types of SAP Fiori apps. Let's review these scenarios.

SAP Business Suite

The three types of SAP Fiori apps, namely transactional apps, fact sheet apps, and analytical apps. Transactional apps run on any database, whereas fact sheet and analytical apps require SAP HANA as the database.

Transactional Applications

Let's consider the architecture of transactional apps as shown in Figure 2.2.

At the top level, there is a box representing clients. These clients can be a browser on a desktop and mobile phone or an SAP Fiori Client app. These clients will open the SAP Fiori launchpad and use it to launch the SAP Fiori apps.

The next level from the top is for the SAP Web Dispatcher. For a transactional app, all the calls will end up at the frontend server. Because of this, SAP Web Dispatcher isn't a necessity, but SAP recommends you use one for the security and load balancing features.

In the next level, we have a frontend server. The SAP Fiori launchpad and the UI resources for all the SAP Fiori apps are hosted on this server. The SAP Fiori launchpad administration requires data related to catalogs, groups, and roles to be stored here as well. SAP Fiori launchpad doesn't store any data in the backend. All the communication to load and perform administrative operations for SAP Fiori launchpad will end at the frontend server.

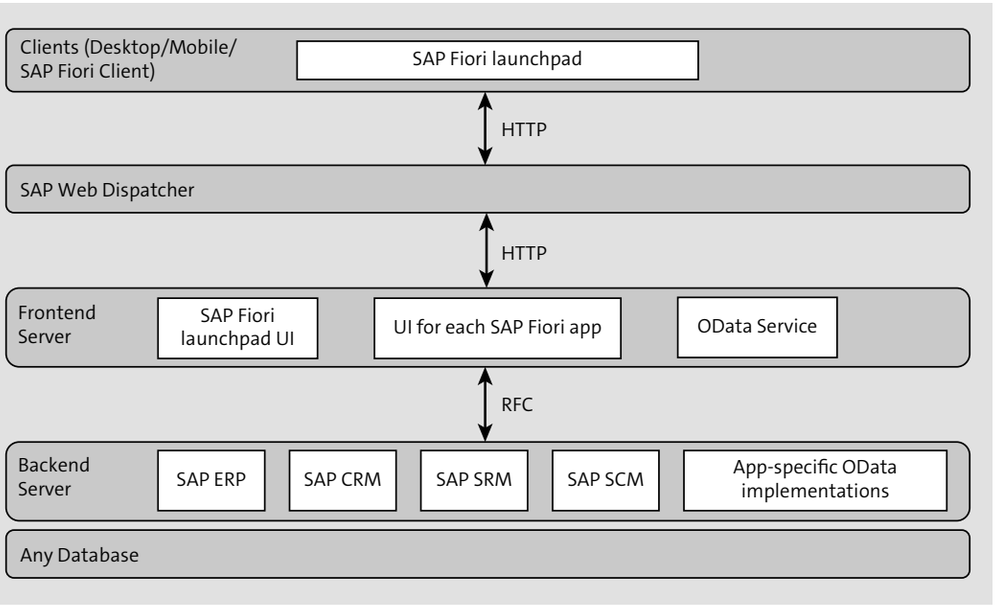


Figure 2.2 Architecture of an SAP Fiori Transactional App

UI resources for each SAP Fiori app are packaged as BSP applications. These app-specific BSPs are hosted on the frontend server as well. When a tile pointing to an SAP Fiori app is clicked on in the SAP Fiori launchpad, it will start loading the app by contacting the frontend server.

The frontend server also exposes the OData services for consumption. Even though services are implemented in the backend servers, service exposure occurs through the frontend server. The frontend server can also connect to more than one backend server for fetching the data.

Next, we have the backend server, which is nothing but the application server where all the business logic resides. This can be an SAP ERP, SAP Customer Relationship Management (SAP CRM), SAP Supplier Relationship Management (SAP SRM), or SAP Supply Chain Management (SAP SCM) systems. OData service implementations also reside here.

The last layer is the database layer. For transactional apps, this can be any of the SAP-supported databases.

Fact Sheet Applications

Fact sheets are read-only applications that give information about a business object or a transaction. They are usually opened from search results on SAP Fiori launchpad or by clicking on links on other factsheet and transactional apps.

Configuring SAP Fiori Search is a prerequisite for fact sheet apps. SAP Fiori Search can be configured to work against multiple backend systems as well. In such cases, you can also configure fact sheet apps to open data from multiple systems. However, when opened, each fact sheet app will provide data from only one backend system.

Figure 2.3 shows the architecture of SAP Fiori fact sheet apps. One big difference when compared with transactional apps is that there is direct communication from SAP Web Dispatcher to the backend server for fetching search results. The UIs for fact sheets are still hosted in the frontend server. Upon receiving search requests, SAP Web Dispatcher routes the calls directly to the backend server. Routing rules to affect these routing changes are required in SAP Web Dispatcher. This communication uses the SAP proprietary INA search protocol to access the search models built in the backend system using the SAP HANA database. This is why fact sheets have SAP HANA as the database prerequisite.

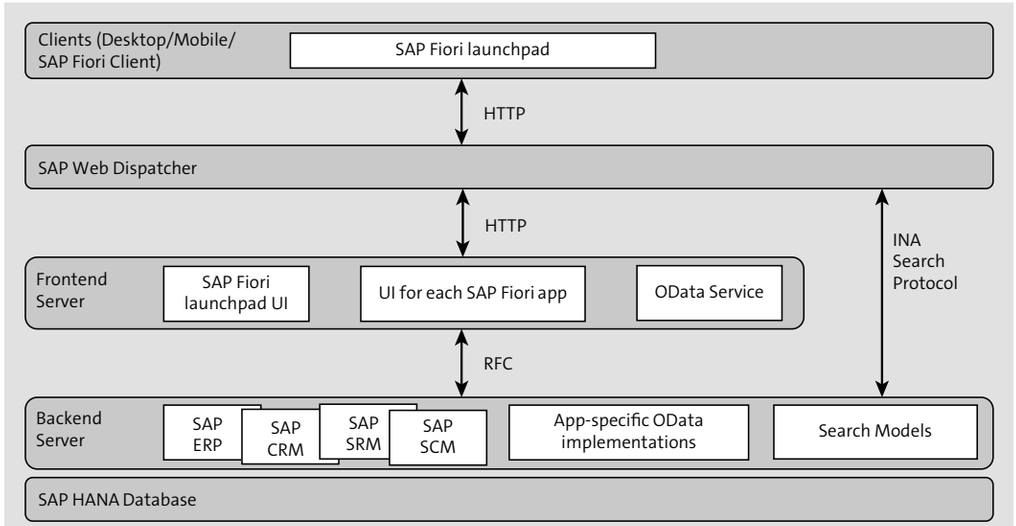


Figure 2.3 Architecture of an SAP Fiori Fact Sheet App

Note

Because fact sheet apps must connect to two different hosts (frontend server and backend server), SAP Web Dispatcher is a mandatory component in the architecture.

Analytical Applications

As we saw in Chapter 1, analytical apps provide capabilities to report by slicing and dicing the business data in the SAP HANA system. Because of this, it makes use of the application server available in the SAP HANA system instead of using the backend server. OData services for analytics are exposed using the SAP HANA XS

engine and provide data for charts and figures in analytical apps. The backend server is bypassed because the business logic in the backend server is of little significance to the analytical apps, and directly contacting the SAP HANA layer will improve performance.

However, for the UI, analytical apps still use the frontend server to host the required UI resources. Because of this, SAP Web Dispatcher is a mandatory requirement for analytical apps as well. Figure 2.4 shows the architecture of an SAP Fiori analytical app.

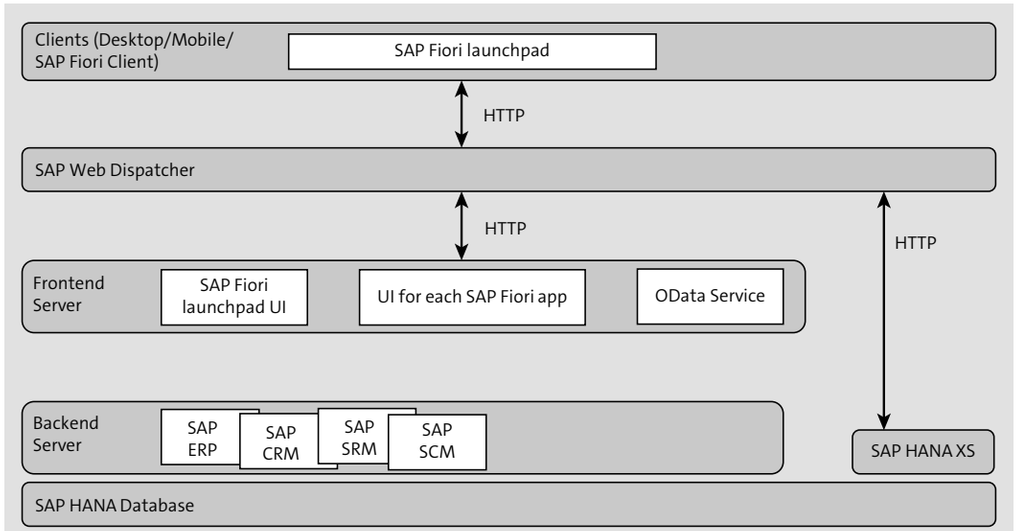


Figure 2.4 Architecture of an SAP Fiori Analytical App

SAP S/4HANA

Because simplification is one of the focuses in SAP S/4HANA, SAP Fiori architecture also aims for simplification at various levels. Now, let's explore various simplifications achieved by SAP S/4HANA in regard to architecture, authorization management, business object models, and technical levels:

- **One archetype**
  The SAP Fiori architecture for SAP S/4HANA provides only one archetype across all three application types (transactional, analytical, and object view/search), unlike three archetypes that existed with SAP Fiori for SAP Business Suite.
  Figure 2.5 shows the one-archetype architecture of SAP Fiori in SAP S/4HANA.
- **One user/authorization management**
  In the earlier architecture for SAP Business Suite for analytical apps, there was a direct connection to the SAP HANA XS layer. This required maintaining users and their authorizations separately. However, in the current architecture, as shown in Figure 2.5, all the connections to SAP HANA occur through the ABAP

layer (i.e., through the SAP S/4HANA layer), and there is no direct access to the SAP HANA system. Thus, there is no need to maintain a separate list of users and handle their authorizations.

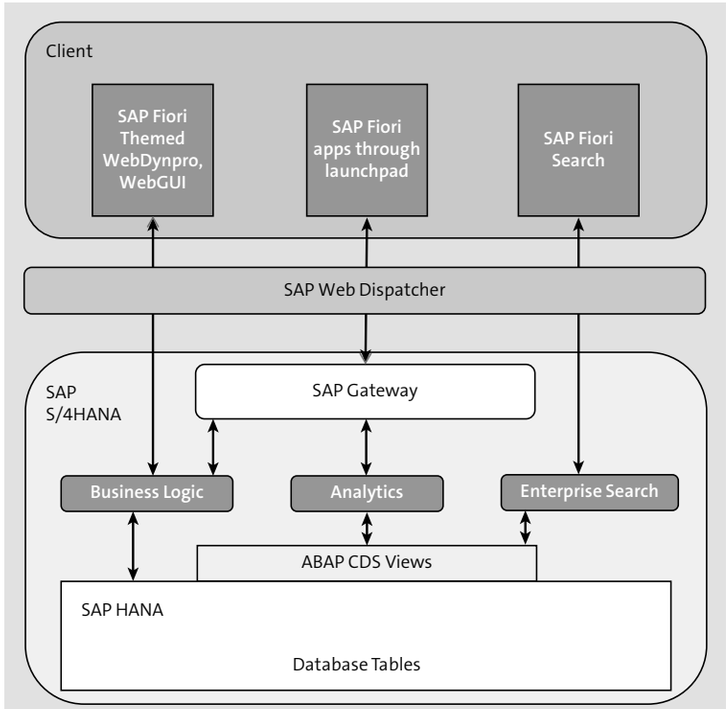


Figure 2.5 SAP Fiori for SAP S/4HANA's Single Archetype Architecture

- **One business model**
  In the SAP Fiori architecture for SAP Business Suite, data modeling was done on the ABAP layer for transactional apps, whereas modeling was done on the SAP HANA XS layer as well with calculation/analytical views. With the current architecture, ABAP Core Data Services (CDS) views are considered as the central part of all data modeling needs. This is true for all the query scenarios in transactional apps as well as for analytical apps, which will need CDS views with analytical annotations.
- **One lifecycle to manage**
  In the earlier architectures, development artifacts were present both in the ABAP layer and the SAP HANA layer. Thus, their lifecycles needed to be handled separately, increasing the effort involved as well as the possibility of mismatches and failures. But in the SAP Fiori for SAP S/4HANA architecture, all the objects, including CDS views, are created in the ABAP layer, thus providing the ability to maintain all the artifacts together and hugely reducing the efforts involved.

- **One protocol (OData) and one implementation layer (SAP Gateway)**
  Both transactional and analytical apps use the OData protocol for the required data. In analytical apps for SAP Business Suite, OData services were delivered by the SAP HANA XS layer; however, in the current architecture, all the required OData services are delivered through SAP Gateway.

Data Flow in SAP Fiori for SAP S/4HANA Transactional Applications

SAP Fiori transactional apps contain two types of data interactions (as seen in Figure 2.6):

- **Data query**
  ABAP CDS views are modeled for all the required data from the business objects. These CDS views will be annotated as well so that they can be used in controls belonging to SAP Fiori elements. CDS views allow you to make use of the code pushdown technique to improve query performance. All queries go through CDS views.

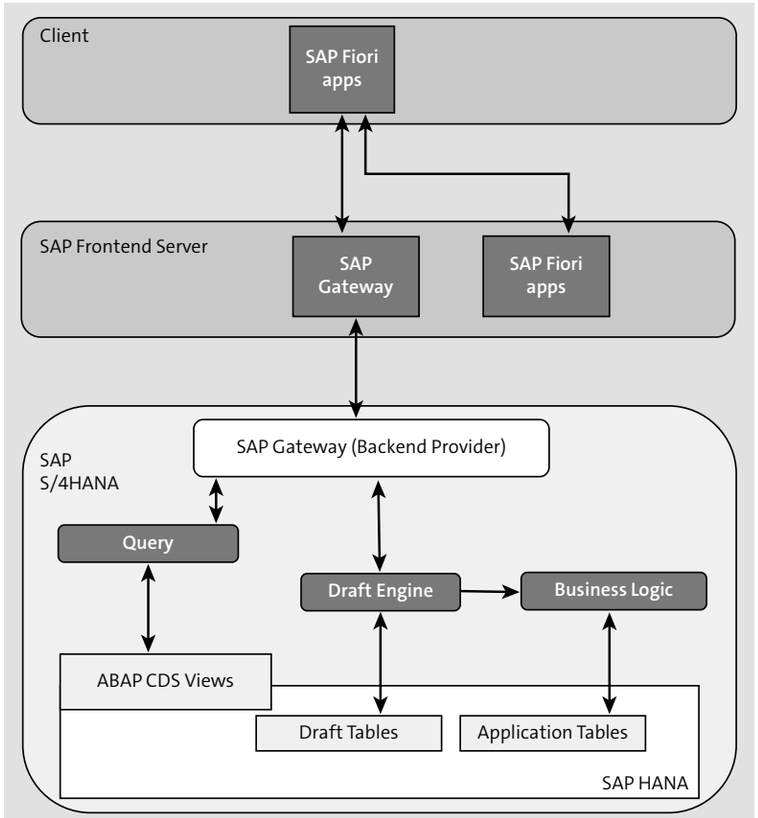


Figure 2.6 Data Flow in an SAP S/4HANA Transactional App

- **Draft handling**
  While making changes to business objects, many times users need to save the business document/object halfway through and then continue at a later time. To facilitate this, a draft handling feature is available.

The Business Object Processing Framework (BOPF) is used to model the business objects and provide all aspects of application development, including draft handling. The moment a business document is edited or created, the draft version gets created. The changes only get saved to the actual business object when saved explicitly.

Draft handling is also used for concurrency handling for a single business object. Only one draft of an existing business object is allowed at a time. When a user opens an object in edit mode, a draft version of that object gets created and assigned to the user. The changes in the draft get copied to the original object, and the draft gets destroyed when the user saves all the changes. Unless the user explicitly saves the changes, no other user will be able to create a new draft and or edit the same object.

Note
For draft handling capabilities, SAP NetWeaver AS ABAP 7.51 SP02 or higher is required.

SAP Fiori Cloud

SAP Cloud Platform provides the SAP Fiori Cloud for quickly mobilizing applications in SAP Business Suite and SAP S/4HANA. SAP Fiori Cloud enables you to renew your UX with minimum cost and effort. SAP Fiori Cloud connects to on-premise systems as well as SAP S/4HANA Cloud systems for integrating with the business data. SAP Fiori Cloud allows users to access SAP Fiori apps using the SAP Fiori launchpad, which is the single point of access for all the SAP Fiori apps on SAP Cloud Platform.

Content of SAP Fiori Cloud

SAP Fiori Cloud offers three types of services.

- **Runtime and configuration services**
  This is a set of services required to configure and run SAP Fiori apps and SAP Fiori launchpad. These services allow you to define catalogs, groups, and roles, as well as assign SAP Fiori apps to catalogs and groups. You can also create multiple SAP Fiori launchpads using these services. These services enable navigation and personalization as well.
- **Lifecycle management services**
  As the name suggests, these services provide ways to package, upgrade, and transport cloud-ready SAP Fiori content.

- **Development services**
  These services allow developers to build and extend SAP Fiori apps. SAP Web IDE is the most important tool as part of these services.

Note

Not all SAP Fiori apps are cloud-ready and available on SAP Fiori Cloud. SAP releases standard SAP Fiori apps selectively based on demand and technical feasibility. However, you can push any of your custom SAP Fiori apps on SAP Fiori Cloud.

Architecture and Landscape

In this chapter, we'll go through various architectural options while using a cloud/on-premise hybrid option for SAP Fiori Cloud. The two important architectural styles can be classified based on how the user accesses the SAP Fiori launchpad: from inside the network or from the Internet.

Based on this, the architectural types are as follows:

- Internal access point
- External access point

Let's consider each of these types in more detail.

Internal Access Point

Many customers are apprehensive about exposing their business data to the cloud but are keen to reap the benefits of a cloud-based SAP Fiori infrastructure. This architectural pattern takes care of these requirements by ensuring that data never reaches the cloud, while SAP Fiori launchpad and other SAP Fiori resources are accessed from the SAP Fiori Cloud.

Figure 2.7 gives a high-level overview of the internal access point architecture.

SAP Web Dispatcher is a very important component of this architecture style because it determines where each call from the client should be routed. When the user/client makes an initial call to SAP Fiori launchpad, SAP Web Dispatcher sees that it's a call to SAP Fiori launchpad-related sources and delegates the call to SAP Cloud Platform. All the calls to load the SAPUI5 library and to load each SAP Fiori app-related resource are also directed to SAP Cloud Platform.

When these resources run on the user's browser, they need business data to be displayed and thus make calls to OData services. These calls will be delegated to the on-premise SAP Gateway system. Upon data retrieval, the data will be shown on the user's browser. So far, the data has never entered SAP Cloud Platform and is retained within the customer's network. This architecture style can be used from mobile phones as well; however, this will only work if the mobile phone is connected to the internal network.

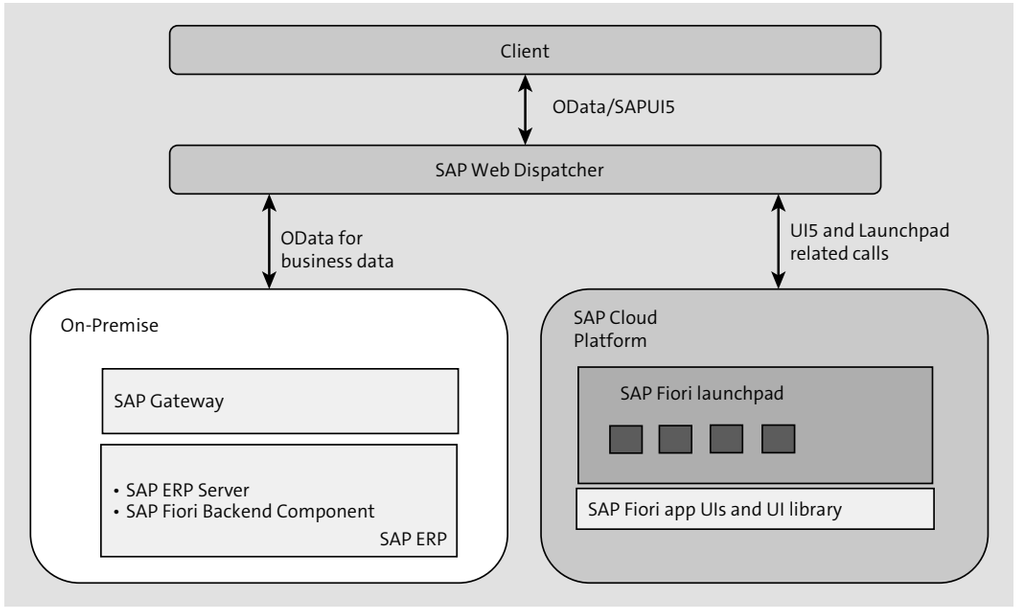


Figure 2.7 SAP Fiori Cloud Architectural Option: Internal Access Point

This is a unique scenario where a customer makes use of SAP Fiori Cloud, and still, none of the business data passes through the cloud, thus meeting complex data flow requirements.

Next, we'll see access from the external access point, which can be from anywhere on the Internet.

External Access Point

As the name suggests, this architectural style exposes both the business data and the SAP Fiori apps to the Internet. Both internal and external users access the SAP Fiori launchpad by connecting to the SAP Cloud Platform. The SAP Fiori launchpad, the SAPUI5 library, and the accessed UI content of SAP Fiori apps are all fetched from the SAP Fiori app repository.

OData services are also routed through SAP Cloud Platform, thus there is no requirement for SAP Web Dispatcher. SAP Cloud Platform connects to on-premise systems via the Cloud Connector, which is a small server hosted by and part of the on-premise network. OData service calls are fetched to the SAP Gateway system through Cloud Connector, and data is fetched or updated.

Figure 2.8 shows the external access point architecture.

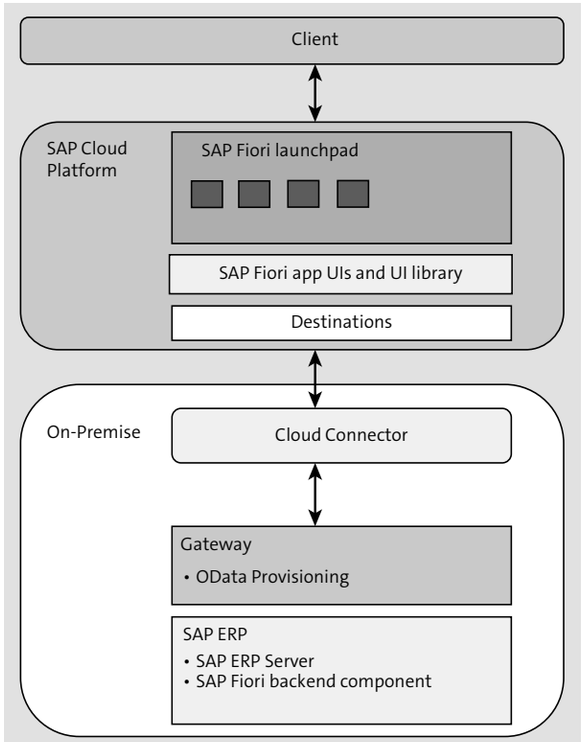


Figure 2.8 SAP Fiori Cloud Architecture: External Access Point

Authentication to SAP Cloud Platform is based on Security Assertion Markup Language (SAML) 2.0 for browser-based single sign-on (SSO), which is a standard feature of SAP Cloud Platform. For connecting to backend systems such as SAP Gateway, SAP Cloud Platform creates a token based on the logged-in user's identity and passes it along with the connection. The backend server may confirm the identity using a SAML 2.0-compliant identity provider.

Using the OData Provisioning Service

As you've seen in the previous architectures, the landscape involves having an SAP Gateway server for provisioning the OData services. However, SAP Cloud Platform provides an additional service for OData provisioning, which can be used to replace the SAP Gateway server.

Figure 2.9 shows such an architecture, where the SAP Gateway server has been replaced by the SAP Cloud Platform OData provisioning service.

The data flow is like the previous architecture with the SAP Gateway server; the only difference is that data is extracted by the OData provisioning service, which connects to the backend again using the Cloud Connector.

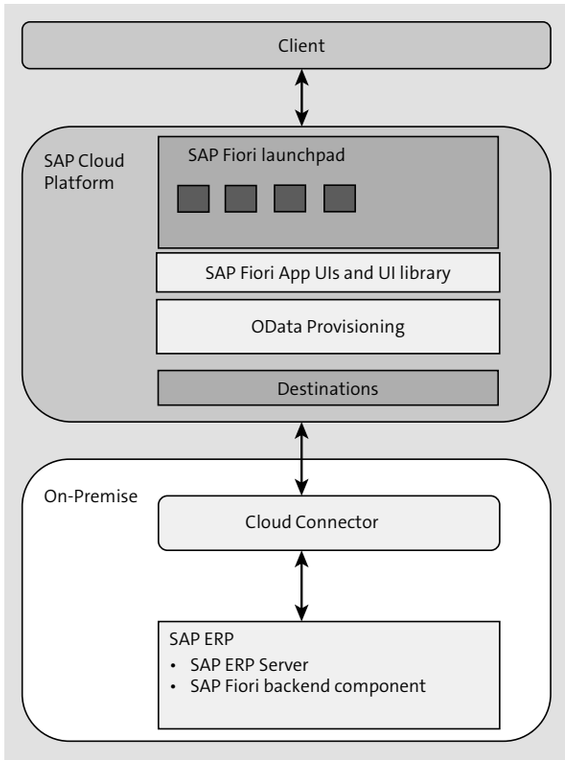


Figure 2.9 SAP Fiori Cloud Architecture: External Access Point with SAP Cloud Platform OData Provisioning Service

SAP Gateway Deployment Options

In this section, we'll consider the various deployment options of SAP Gateway and their advantages. The SAP Gateway product is made up of two functional components: a server component and a backend component. The server component is made up of the GW_CORE and IW_FND add-ons, while the backend component is made up of the IW_BEP software component. You register and expose OData services on the server with the IW_FND component. You implement (code) an OData service on the server with the IW_BEP component. The deployment options determine how these components are installed along with the SAP backend system.

However, starting from SAP NetWeaver version 7.40, all these components have been added to a new software component called SAP_GWFND. So, you no longer need to decide which software component is installed in which system; instead, you just need to choose which functionality is used from which system.

SAP Fiori supports two deployment options for on-premise installation: hub deployment and embedded deployment. Let's consider each of these options in detail.

Hub Deployment Option

In the hub deployment option, there is a dedicated SAP NetWeaver server available for SAP Gateway server functionalities. This server will be either an SAP NetWeaver 7.31 or a lower version with GW_CORE and IW_FND components or an SAP NetWeaver 7.40 or higher system leveraging component SAP_GWFND.

Figure 2.10 displays the hub architecture in which the service is registered on the SAP Gateway server (using Transaction /IWFND/MAINT_SERVICE).

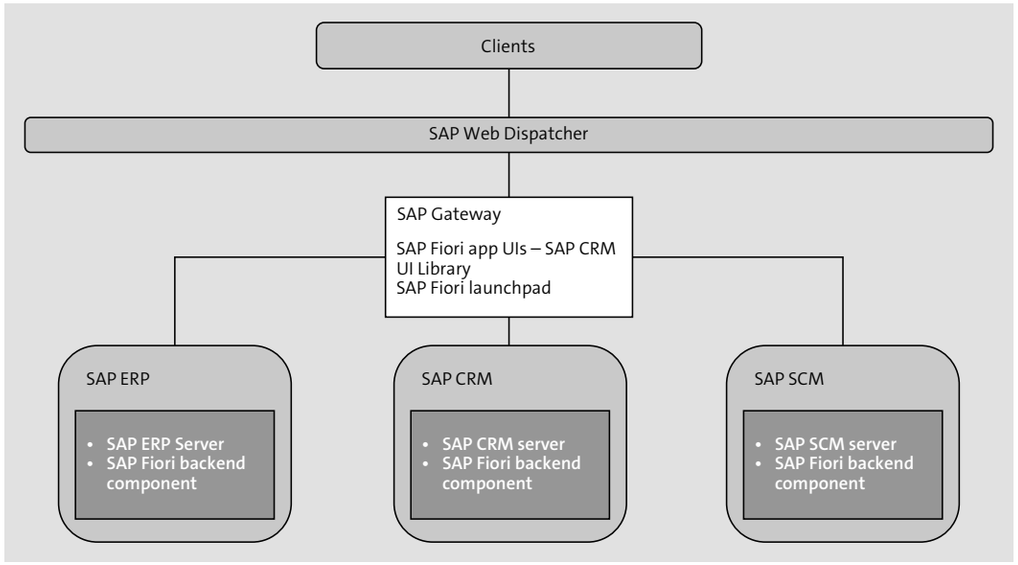


Figure 2.10 SAP Gateway Hub Deployment Architecture

Multiple Routings

In a hub architecture, as shown in Figure 2.10, a hub system (SAP Gateway) can be connected to multiple backend systems. These backend systems can be business systems such as SAP ERP, SAP CRM, or SAP SCM. Each backend system will have corresponding SAP Fiori app-specific backend software components installed. However, the frontend components of all these diverse SAP Fiori apps are installed on the hub system. SAP Fiori launchpad hosted on the hub system not only shows SAP Fiori apps based on SAP ERP but also other systems such as SAP CRM and SAP SRM. Thus, one of the main advantages of the hub architecture is the ability to provide a single starting point to diverse applications by providing a single integration point.

Tip

In an on-premise SAP Fiori architecture, the dedicated SAP Gateway system in the hub architecture is called the SAP Fiori frontend server. This is because this server also contains the central UI, that is, SAP Fiori launchpad, SAPUI5 library, and a repository for holding all SAP Fiori app-specific UI parts.

Other advantages of the hub deployment option are discussed in the following subsections.

Separating the User Interface Lifecycle

By having a separate server, the lifecycle of UI components, UI library, and SAP Fiori launchpad are separated from that of the backend business systems. Usually backend business systems are upgraded less frequently considering the huge regression testing requirements when any changes are made. However, UI components such as the SAPUI5 library get upgrades as often as every quarter. Having a separate SAP Gateway system allows customers to upgrade the SAP Gateway systems much faster than the business systems.

Better Security and Authentication

In a hub architecture, HTTP connections always end at the SAP Gateway system, protecting business systems from any direct attacks. In addition, because there is a separate system for SAP Gateway, this system can be deployed on the demilitarized zone (DMZ) for external Internet access.

Because the hub system can be based on a newer release, it can support a wide variety of authentication options such as Kerberos, SAML, or OAuth. However, the main disadvantage of the hub architecture is the need for maintenance of an additional server for SAP Gateway.

Embedded Deployment Option

In an embedded deployment option, both the server component (GW_CORE and IW_FND) and backend component (IW_BEP) are installed on the business systems for SAP NetWeaver 7.31 or earlier. On an SAP NetWeaver 7.40 business system, no additional add-on or server is required. Both service development and registration are done on the backend business system.

As you can see in the embedded system architecture shown in Figure 2.11, in an embedded deployment option, in addition to SAP Fiori app-specific backend components, frontend components are also installed on the backend business system. Unlike the hub architecture, you can't have a single SAP Fiori launchpad to represent all the backend business systems. Each backend system will have an SAP Fiori launchpad of its own.

Less Runtime Overhead

Because there is only one system to traverse, data latency is reduced. In addition, from SAP NetWeaver 7.50 SP4, there are enhancements specific to embedded deployment that take advantage of the co-deployment of SAP Gateway and business systems to further improve the performance.

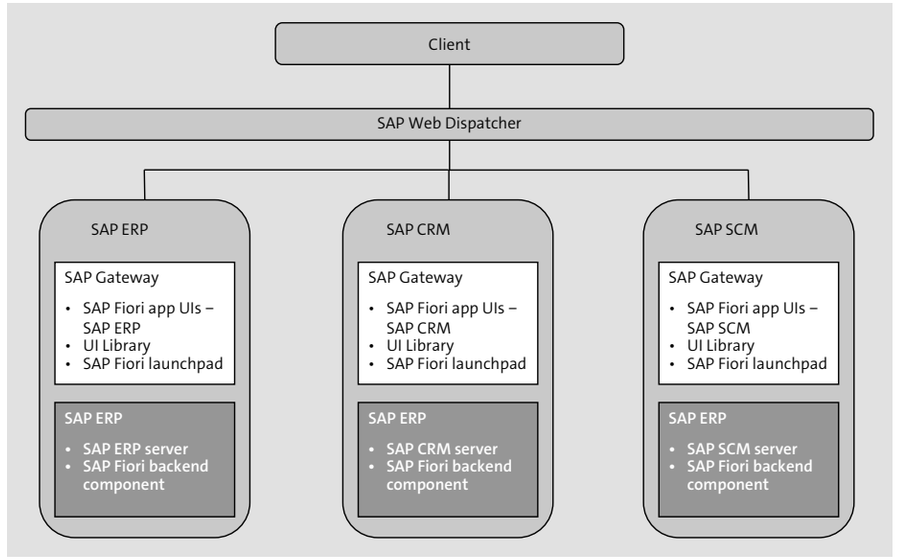


Figure 2.11 SAP Gateway: Embedded Deployment Option

No Extra System to Maintain

Because there is no additional SAP Gateway system involved, there is one less system to maintain, thus reducing costs.

Disadvantages

The embedded deployment option has the following disadvantages:

- Because the embedded system should not be used as a hub system for other business systems, you can no longer have a single integrated SAP Fiori launchpad for all your business systems.
- You can't get frequent innovations to the SAP Gateway system because SAP backend business systems can't be upgraded as frequently.
- You need to ensure additional security measures in the network because HTTP connections end at the business systems, thus exposing them to external Internet attacks.

SAP Cloud Platform OData Provisioning

In this deployment option, SAP Cloud Platform's OData provisioning service is used. SAP Cloud Platform's OData provisioning service replaces the SAP Gateway system in the hub deployment option. Thus, the advantages of this deployment option are like that of the hub deployment option, and this is one of the preferred deployment options by customers. Figure 2.12 illustrates the SAP Cloud Platform OData provisioning deployment option.

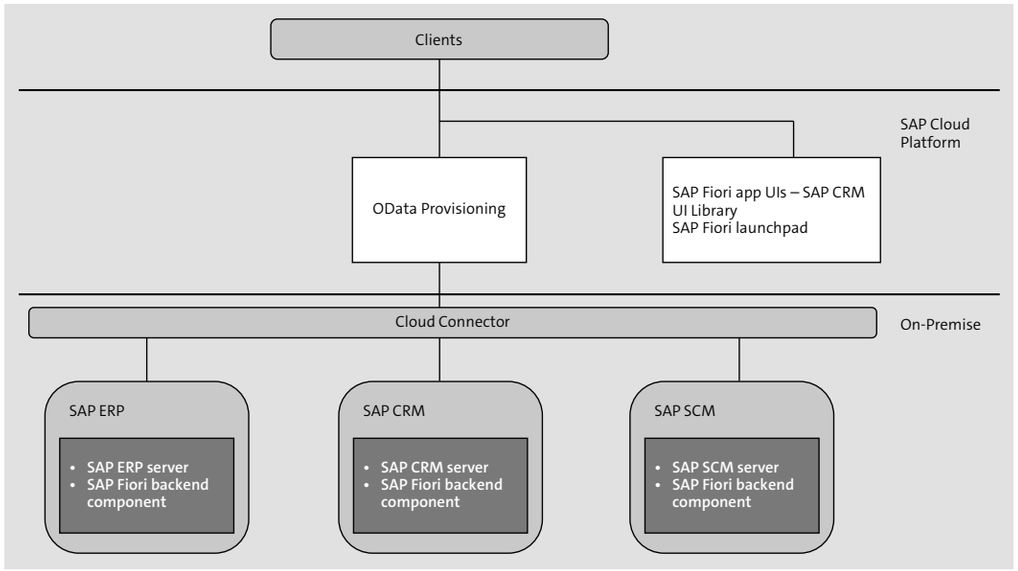


Figure 2.12 SAP Gateway Deployment: OData Provisioning

The advantages of this deployment are as follows:

- Lower total cost of ownership (TCO) compared with SAP Gateway
- Frequent software updates on the cloud
- Automated system monitoring and administration tasks
- Being part of the cloud brings in the quality of elasticity when required

When used in an SAP Fiori Cloud landscape, OData provisioning takes care of OData service registration and exposure capabilities of the SAP Gateway server. The remaining capabilities of acting as an SAP Fiori UI repository are handled by SAP Cloud Platform as shown in Figure 2.12.

Tip

The OData provisioning service is under active development, and you can expect this service to cover many more related use cases.

The OData provisioning service can be found and activated easily on SAP Cloud Platform, as shown in Figure 2.13.

To enable service registrations, each business system needs to be configured as a destination within the service configuration of the OData provisioning service. To go to destination maintenance, click on the service tile as shown in Figure 2.13, and then click on **Configure Service**. Click on **New Destination**, and maintain the details of a backend business system. Figure 2.14 shows the connection details for an SAP-provided ES4 demo system.

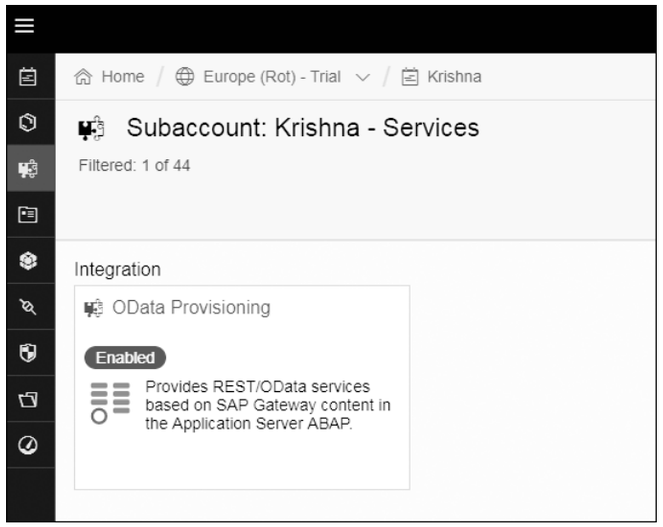


Figure 2.13 OData Provisioning Service

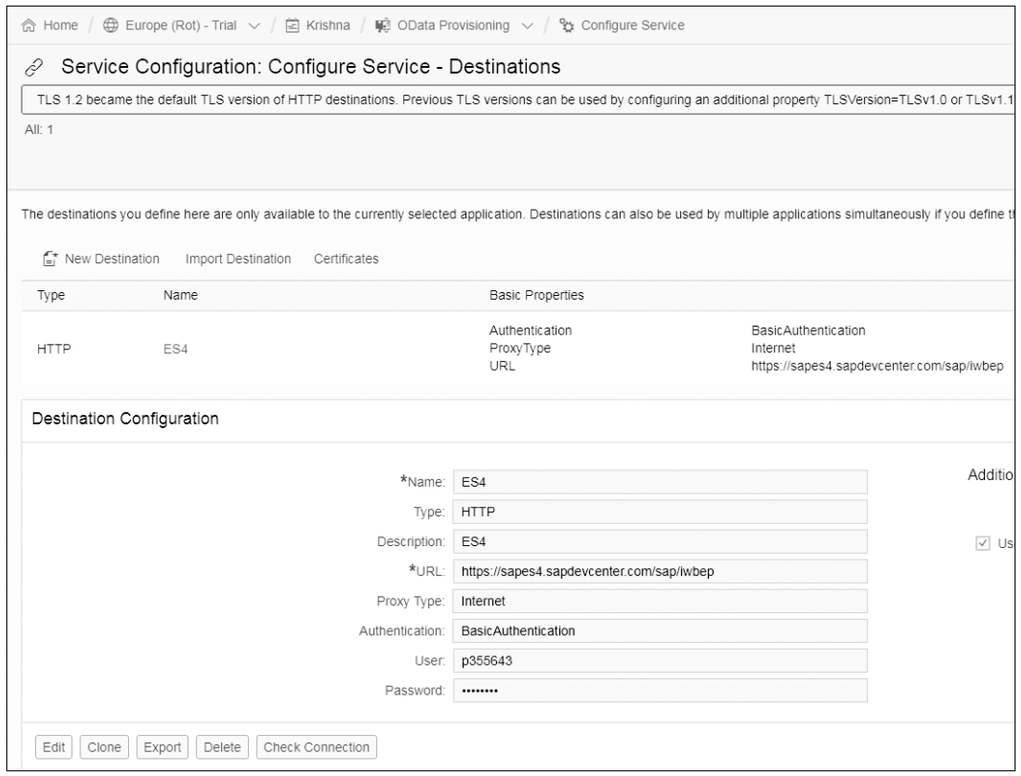


Figure 2.14 Destination for ES4 System for OData Provisioning

Now you can go to the OData provisioning service. Click on **Register**, and choose the destination **ES4**. Upon searching, all the services from the ES4 system will be shown. Select one of the services, and click the Register button shown in Figure 2.15 ❶. Upon successful registration, this service will be listed in the catalog of services, as shown in Figure 2.15 ❷.

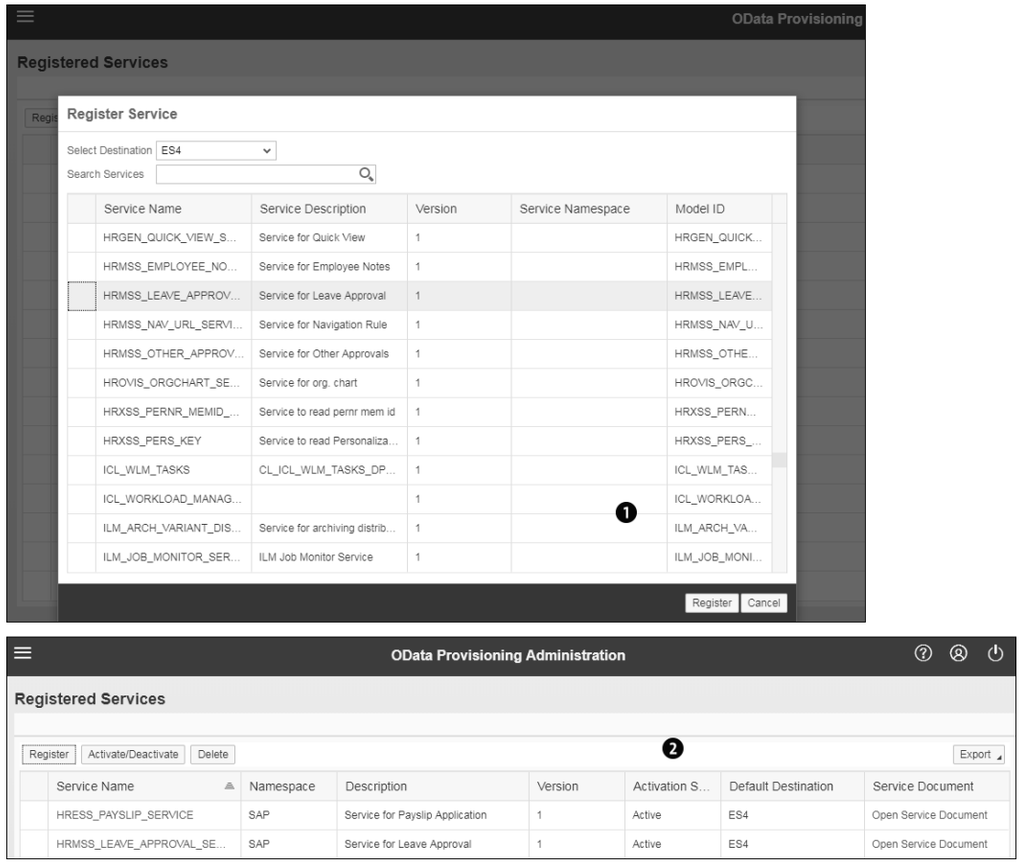


Figure 2.15 Service Registration in the OData Provisioning Service

SAP Fiori Launchpad Configuration

As you've seen so far, SAP Fiori launchpad is a central point to the SAP Fiori strategy, and it provides a single point of access to all the relevant SAP Fiori apps for a user. In this section, we'll discuss the techniques involved in assigning SAP Fiori apps to users by using the SAP Fiori catalog, SAP Fiori group, and Transaction PFCG roles.

SAP Fiori launchpad designer is a browser-based tool used to administer and configure SAP Fiori launchpad. It's recommended to use the SAP Fiori launchpad designer on a desktop screen.

SAP Fiori launchpad designer allows you to do the following:

- Configure static, dynamic, and news tiles
- Create catalogs and groups and then assign tiles to them
- Assign these artifacts to transport requests for transporting

Launching SAP Fiori Launchpad Designer

SAP Fiori launchpad designer can be launched with the following URL:

https://<server>:<port>/sap/bc/ui5_ui5/sap/arsrvc_upb_admn/main.html?sap-client=<client>&scope = <CONF/CUST>

<server>:<port> should point to the SAP Gateway server in a hub architecture system, whereas it will be pointing to the backend business system in an embedded system.

Scope for Content Adaption

As you can see, the SAP Fiori launchpad designer URL has two types of scopes, and a third scope is available as well:

- **CONF-configuration scope**
  Configuration scope refers to settings that are cross-client and thus system specific. All the standard content delivered by SAP is in this scope. You can use the URL parameter *scope=CONF* for opening the SAP Fiori launchpad designer in configuration mode.
- **CUST-customization scope**
  Customization scope is specific to a client. This is also the default scope for all SAP Fiori launchpad designer content if no URL parameter scope is specified. You can use the URL parameter *scope=CUST* to specify this scope, however. Changes made with the CUST scope take precedence when compared to those made with the CONF scope.
- **PERS-personalization scope**
  Although this scope isn't allowed from the SAP Fiori launchpad designer, whenever a user performs personalization on the SAP Fiori launchpad, the PERS scope is used. The personalization scope has the top precedence when compared to CONF and CUST scopes.

Figure 2.16 shows a screen of the SAP Fiori launchpad designer. On the master list on the left side panel, you can select either **Catalogs** (default) or **Groups**. The resulting screen shows details of either a catalog or a group.

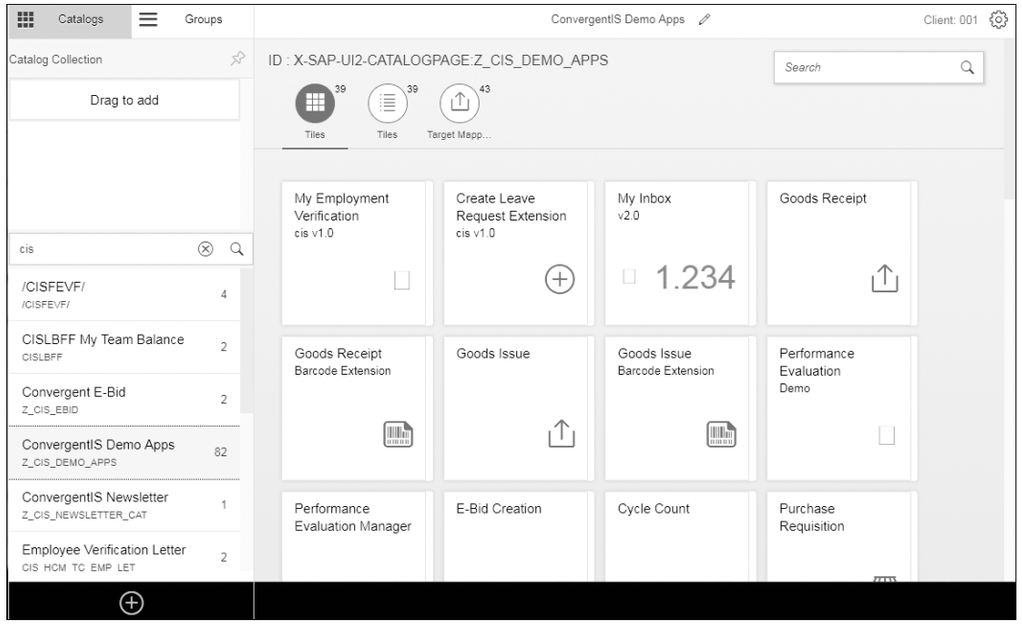


Figure 2.16 SAP Fiori Launchpad Admin Tool

SAP Fiori Catalogs

An SAP Fiori catalog is a set of applications that will be assigned to a role. Unless an application is part of a catalog and assigned to a user's role, the user won't be authorized to access the application.

However, all the applications assigned to a user using the catalog won't be available in the user's SAP Fiori launchpad entry page. The user needs to browse through his catalogs and choose applications to be available in the entry page.

SAP Fiori Groups

An SAP Fiori group, as the name suggests is a semantic group of all the SAP Fiori apps authorized to a user. When an SAP Fiori group is assigned to a user via a role, all these SAP Fiori apps will appear on the entry page of the user's SAP Fiori launchpad with the group title.

Figure 2.17 illustrates how the combination of catalogs/groups and roles affect the authorization and display of applications on the home screen.

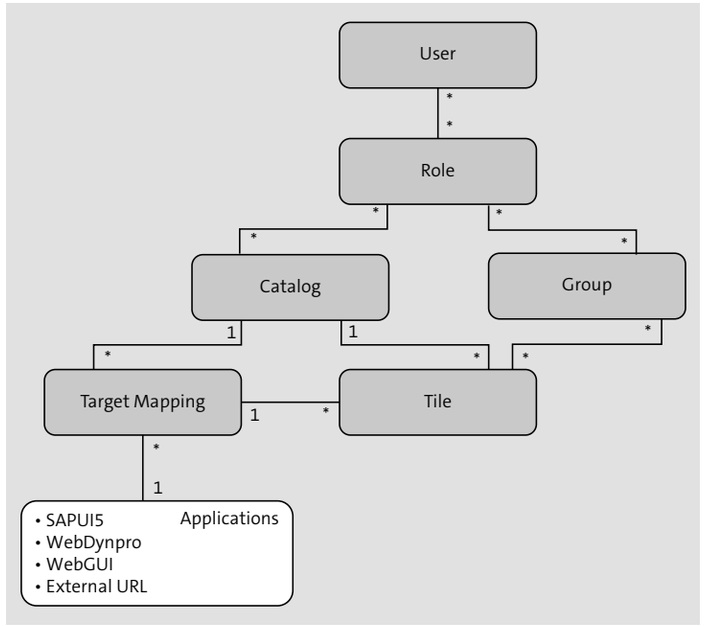
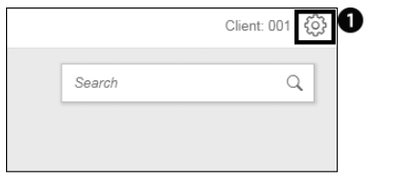


Figure 2.17 SAP Fiori Catalog, Group, and Role Relations

Let's do a small exercise of assigning an SAP Fiori app to a user using these concepts.

1. Navigate to the SAP Fiori launchpad designer.
2. Before making any changes, it's important to link a Customizing transport with the SAP Fiori launchpad designer, so that all further changes will get captured within that transport. To do so, click on the gear icon (Figure 2.18 ❶) on the top right of the SAP Fiori launchpad designer, and choose an available Customizing transport from the **Customizing Request** dropdown ❷, as shown in Figure 2.18.
3. Click **OK** ❸.



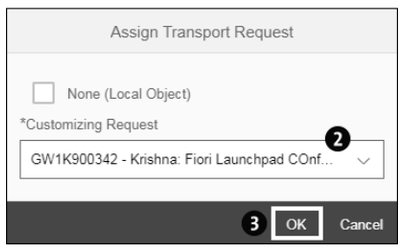
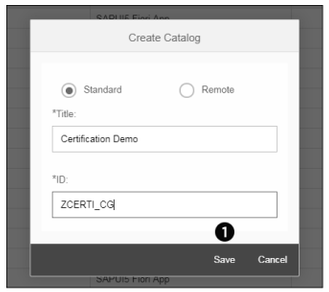


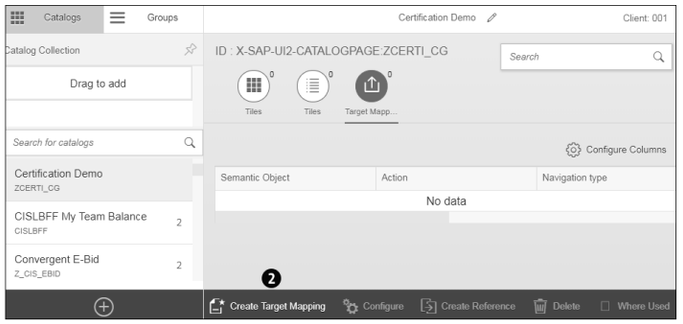
Figure 2.18 Assigning a Transport to SAP Fiori Launchpad Designer

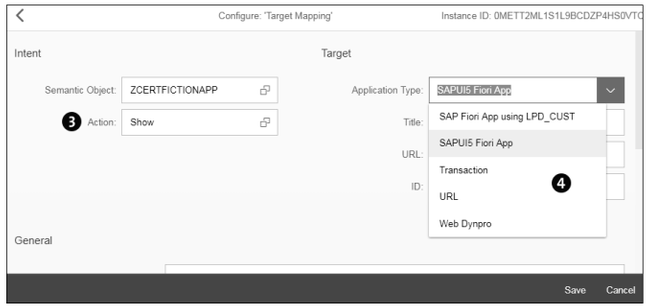
Note

If there are no transports available, create a new Customizing transport in Transaction SE09.

4. Next, create a new catalog by clicking on the **+** button at the bottom of the master list.
5. Provide a **Tile** and an **ID** in the customer's namespace, and click **Save**, as shown in Figure 2.19 ❶.







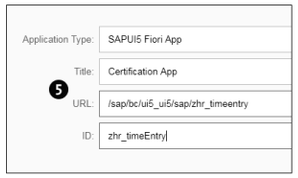


Figure 2.19 Creating a Catalog and a Target Mapping

6. Navigate to the **Target Mapping** tab, and click on **Create Target Mapping** ❷.
7. In the form that opens, in the **Intent** section, enter a **Semantic Object** and an **Action** ❸. A target mapping defines supported device types and application details for an intent. If the required semantic object isn't present, you can easily create one in Transaction /UI2/SEMOBJ.
8. Choose from several options in the **Application Type** list as shown in Figure 2.19 ❹. These are the various types of applications that can be opened from an SAP Fiori tile. You can create a target mapping for an SAP Fiori app, a transaction, an external URL, and a Web Dynpro application.
9. There are two options to create a target mapping for an SAP Fiori app. The first one, **SAP Fiori App using LPD_CUST**, is a legacy option that required the application details to be mentioned in Transaction LPD_CUST. For this exercise, choose the second option, **SAPUI5 Fiori App**, which will allow you to enter application

details (URL and component name) on the screen, as shown in Figure 2.19 ❺. Click on **Save** to save the target mapping.

Now you need to create a tile referring to the target mapping. Follow these steps:

1. Navigate to **Tiles**, and click on the **+** button (Figure 2.20 ❶) to create a new tile.
2. You have three options to choose from for tile templates:
   - **App Launcher – Dynamic** will have a tile with a number and some dynamic text on it. This can be useful if you want to give a numeric summary of some process on the tile itself.
   - **News Tile** will show a feed of news from a specific source.
   - **App Launcher – Static** will show a tile with only static content on it.
3. Choose **App Launcher – Static** for the tile, as shown in Figure 2.20 ❷.
4. In the form that opens, provide a **Title** and a **Subtitle** for the tile.
5. Under **Navigation**, choose the **Semantic Object** and **Action** ❸ you provided while creating the target mapping. This is how the tile is linked to the target mapping.
6. Click on **Save** to save the tile. You've now completed all the steps to create a catalog and add a tile to it.

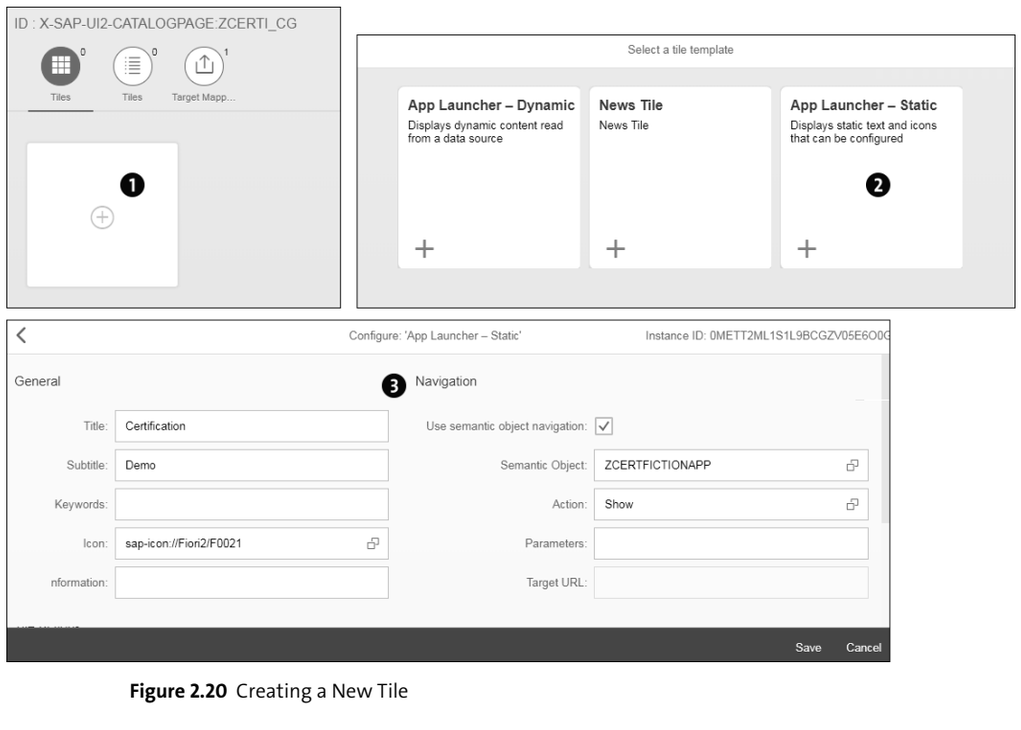


Figure 2.20 Creating a New Tile

Now let's create an SAP Fiori group for this tile by following these steps:

1. Click on **Groups** in the master list, and then click on the **+** icon on the master list to initiate creating a new group.

2. Provide a **Title** and an **ID** (Figure 2.21 ❶) just like you did when creating a catalog. The title mentioned here will be used as the group heading in the SAP Fiori launchpad.
3. To add a tile to the group, select the **Catalog** ❷ that contains the tile, and then click on the + button below the tile to add it to the group ❸.

Now that you've created a catalog and a group, both of these need to be assigned to a Transaction PFCG role ❹, so that all users having this role will see the new tile in their SAP Fiori launchpad shown in Figure 2.21, ❺.

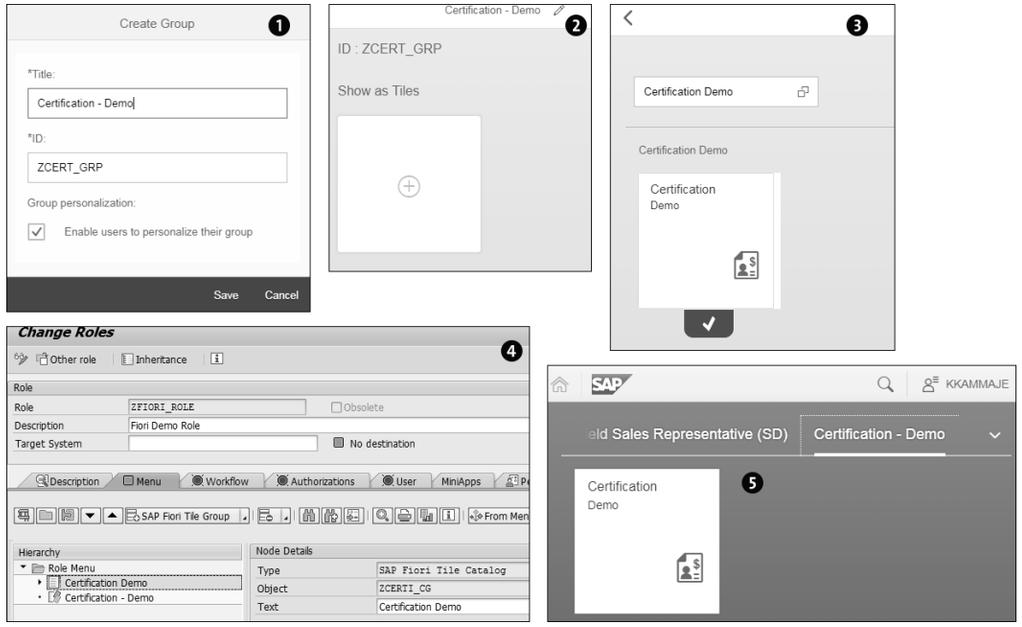


Figure 2.21 Creating a Group, Assigning the Catalog and Group to the Role, and the Result in SAP Fiori Launchpad

SAP Fiori Theming

A common requirement is to modify the SAP Fiori launchpad and the SAP Fiori apps with the corporate theme of the customers, for example:

- Replacing the SAP logo with the customer's logo
- Replacing the standard color combinations with the customer's corporate preferences
- Providing a corporate background image to the SAP Fiori launchpad

Although these requirements can be met by directly writing and including custom Cascading Style Sheets (CSS), it can be very tough to maintain these CSS files, thus increasing the TCO.

SAP provides the UI theme designer tool to make it easy to create and maintain custom themes. The UI theme designer is not only specific to SAPUI5 and SAP Fiori apps but also available to other SAP UI technologies such as Web Dynpro, BSP, HTMLB, SAP Enterprise Portal, SAP GUI for HTML, SAP NetWeaver Business Client (NWBC), and the WebClient UI Framework in SAP CRM.

Overview

UI theme designer is available as part of the central UI of the frontend server infrastructure. When working with SAP Fiori apps, you'll be using the UI theme designer hosted on the SAP Gateway server. However, the tool is also available in SAP Cloud Platform as a service and as part of SAP Enterprise Portal.

UI theme designer can be launched via Transaction /UI5/THEME_DESIGNER in your SAP Gateway/frontend system. You can also go to the following URL to open the tool:

https://<host:port of your Gateway System>/sap/bc/theming/theme-designer

The **UI Theme Designer** screen is shown in Figure 2.22. On the left, you can see a list of custom **Themes** created within this server so far. You can select any of these custom themes and perform various actions, including **Edit**, **Rebuild**, **Delete**, **Duplicate**, or **Rename**.

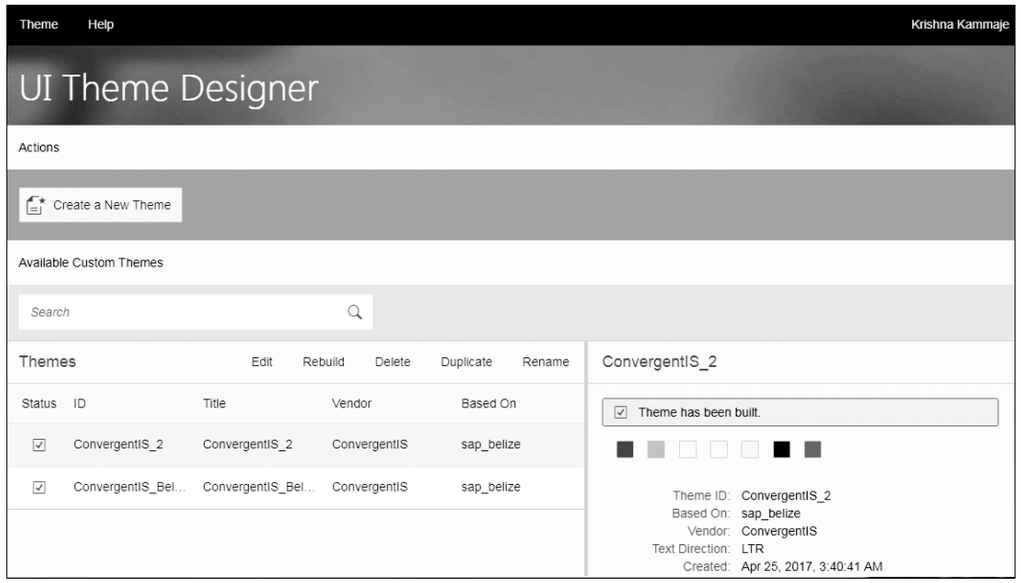


Figure 2.22 Home Screen of the UI Theme Designer

The **Create a New Theme** button is used for hcreating a brand-new theme based on one of the existing themes. The box on the right side of the screen displays the details about the selected theme.

Creating a Theme

Let's create a new custom theme based on the SAP-delivered Belize theme by following these steps:

1. Click on the **Create a New Theme** button. This opens a three-step wizard popup, as shown in Figure 2.23.
2. In the first step, you need to choose a **Base Theme** to start from because a theme contains numerous properties and attributes that are difficult for customers to create from scratch. By basing the theme on an existing theme instead, all those parameters will be copied, and customers can go about making incremental changes on the copied theme. For this example, choose **SAP Belize**.
3. Click on **Step 2**.

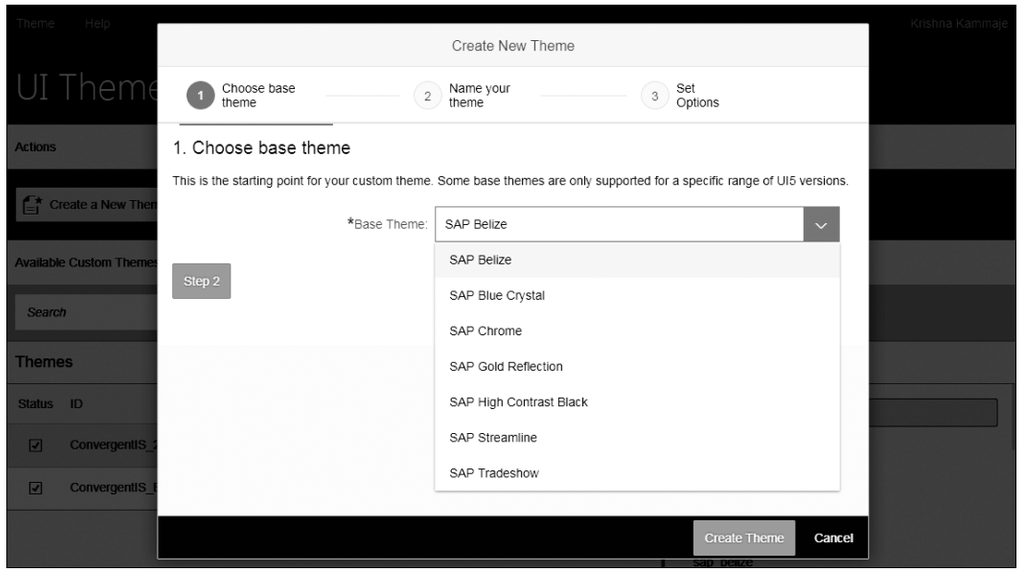


Figure 2.23 Step 1: Choosing a Base Theme

4. Provide a **Theme ID** and a descriptive **Title** for the new theme, as shown in Figure 2.24.
5. Click **Step 3**.
6. In the next screen, provide a **Vendor** name, and choose whether the new theme supports **RTL** (right-to-left scripts). Click on **Create Theme**.
7. In the new screen that appears, enter the target pages. Target pages are used to preview and test the changes you made to the theme for verifying the results. You can add multiple target pages, so that you can instantly see the effect of your theme on all those pages.

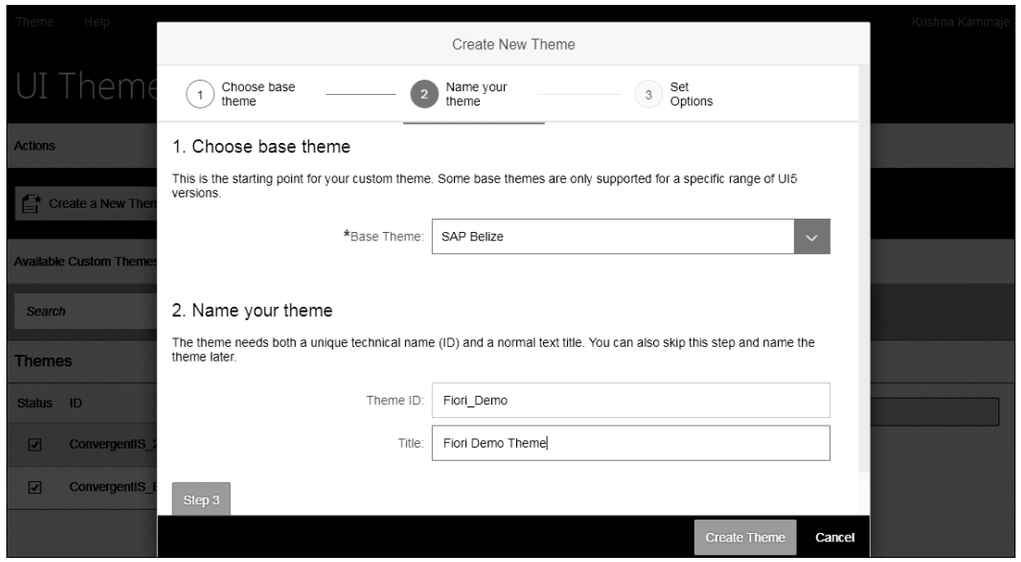


Figure 2.24 Step 2: Providing a Unique ID and a Name for the New Theme

Warning
Adding the target page doesn't assign the theme to the target page permanently.

In the **Link to Application** field, enter the SAP Fiori launchpad URL and a name identifying it in the **Name of Application** field, as shown in Figure 2.25.

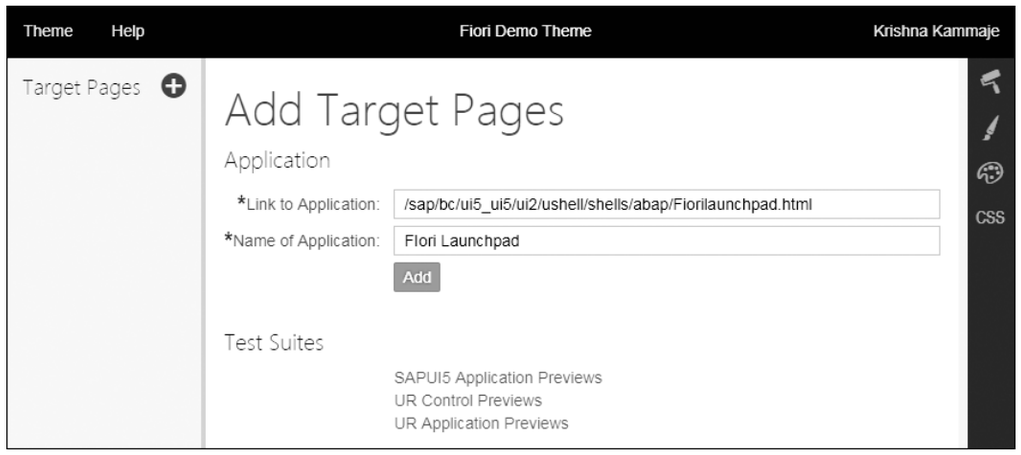


Figure 2.25 Adding Target Pages for the New Theme

8. Click on the **Add** button.

This will load the target page within the UI theme designer, as shown in Figure 2.26.

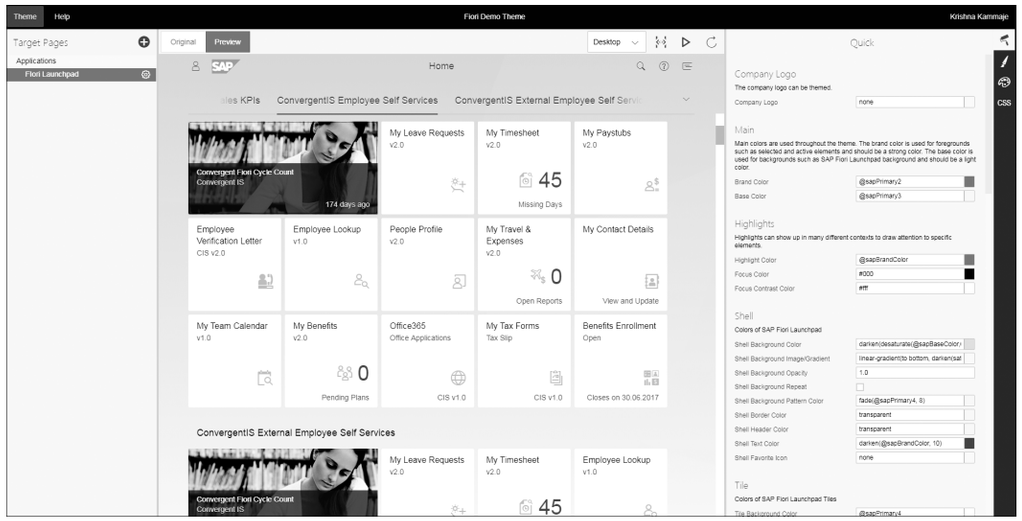


Figure 2.26 Target Page along with the Quick Toolbar

The central area where the target page is loaded is called a canvas, and it has a toolbar providing various options. Figure 2.27 describes the various available options on the canvas area. There are options to compare the current state with the original page ❶, view the page in different device types ❷, preview the page in full width ❸, run the preview in a new tab ❹, and refresh the preview ❺.

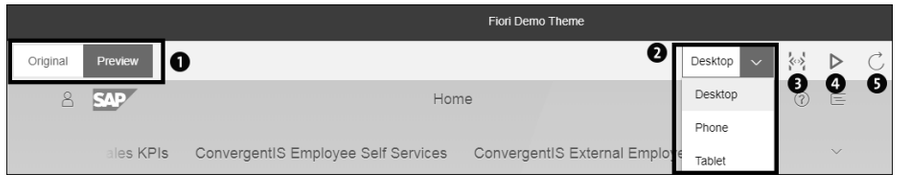


Figure 2.27 Options Available for Previewing

On the right, the **Quick** toolbar opens by default. This toolbar has enough options for beginners as well as for most frequent requirements. You can set the **Company Logo**, **Brand Color**, **Base Color**, and other properties for frequently required items such as **Shell**, **Application Backgrounds**, **Tile**, **Object Header**, and so on, as shown in Figure 2.28.

The second icon (paintbrush) on the right vertical toolbar opens an **Expert** window that provides advanced options based on theme parameters. These theme parameters are reused in all of SAP Fiori launchpad and its applications.

The third icon on the right vertical toolbar is **Palette**, which allows customers to predefine their corporate colors and reuse them in both the **Quick** and **Expert** windows. You can simply enter a parameter name in **New Parameter** and then select the color and its properties (e.g., hue). Click on the **+** button to add the parameter, as shown in Figure 2.29. There are two custom parameters now in total.

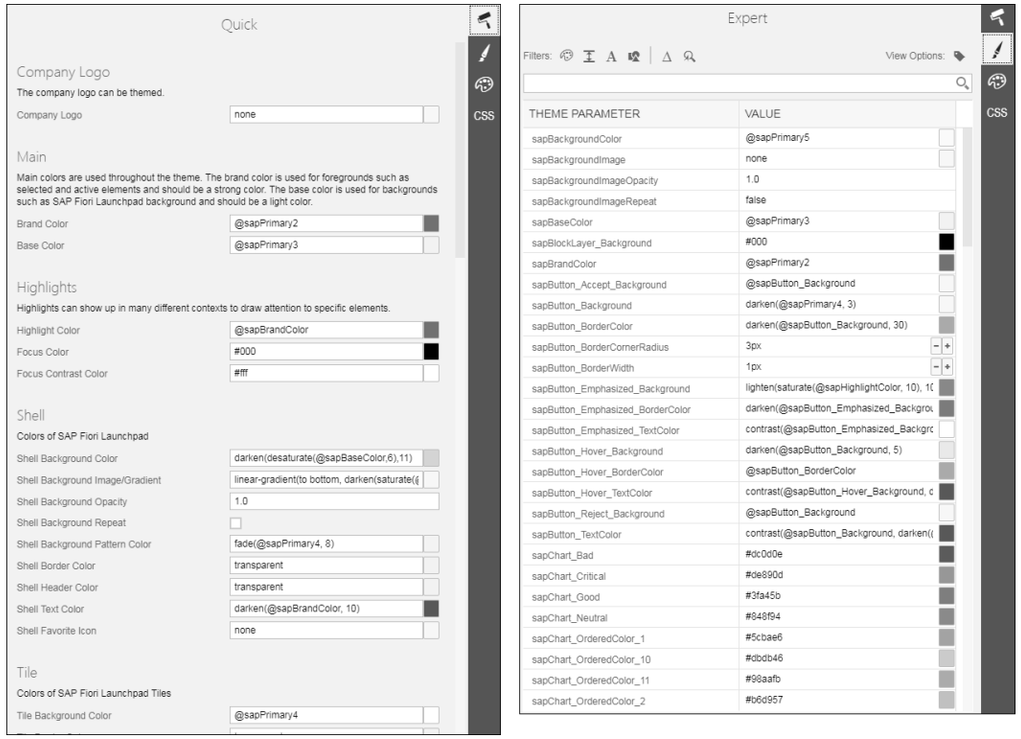


Figure 2.28 Quick and Expert Options

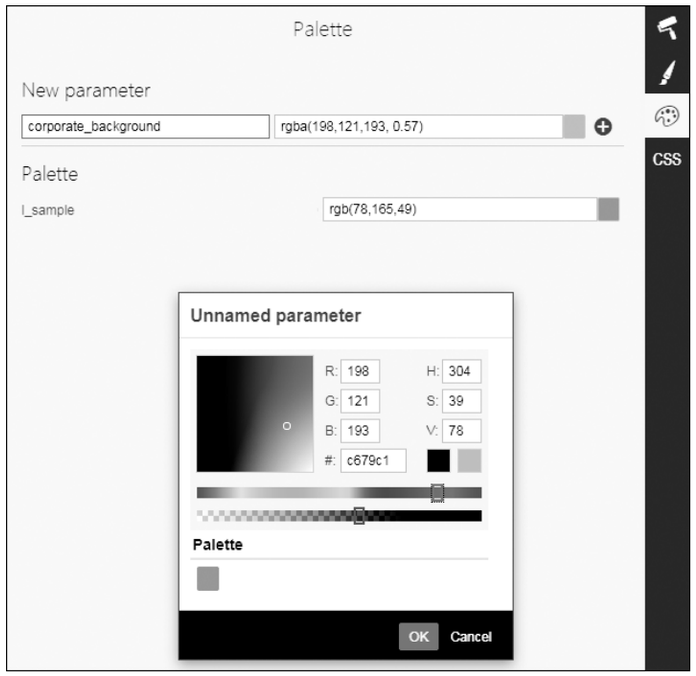


Figure 2.29 Using Palette to Create Custom Parameters with Color Values

Now, go back to either the **Quick** or **Expert** window, choose a parameter, and click on the color picker button. You'll see that new custom parameters are available (Figure 2.30).

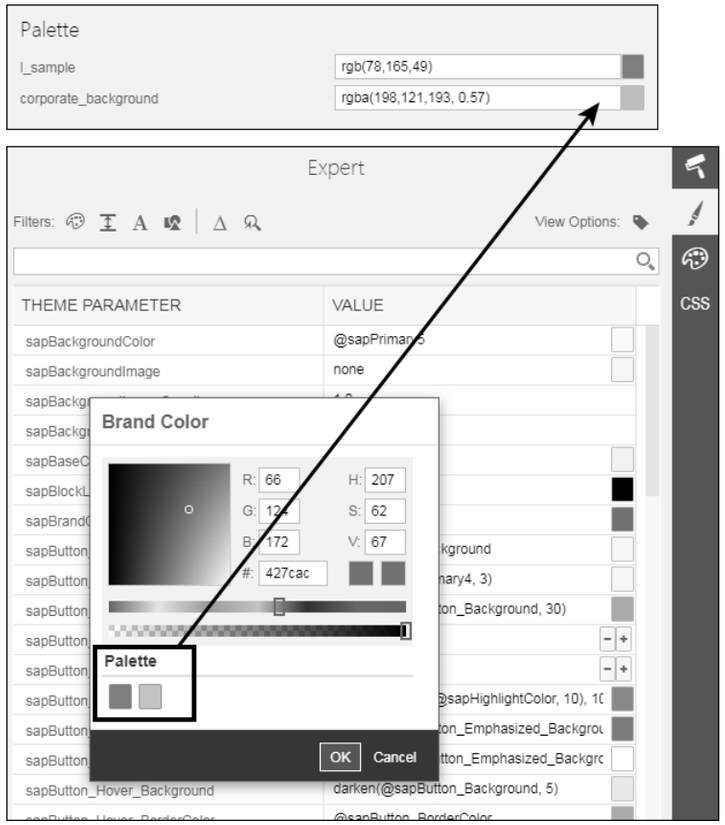


Figure 2.30 Option for Using Custom Palettes/Parameters

The last option on the right toolbar is **CSS**. As the name suggests, this enables customers to influence the appearance of apps by inserting their own CSS. However, changes achieved by adding custom CSS can break across upgrades, and SAP provides no guarantee of those changes working. In such cases, customers need to document the expected outcomes well and redo the changes after the upgrade.

Figure 2.31 shows where CSS class names for the status text **No Assignments** were copied and the color was changed. When you click **Apply**, the changes are reflected on the canvas.

After making changes, the theme can be saved by clicking on **Theme** on the menu bar and selecting **Save**. The theme needs to be built before it can be used. To build it, choose **Theme** • **Save & Build** from the menu bar. The new theme is now available in the home screen.

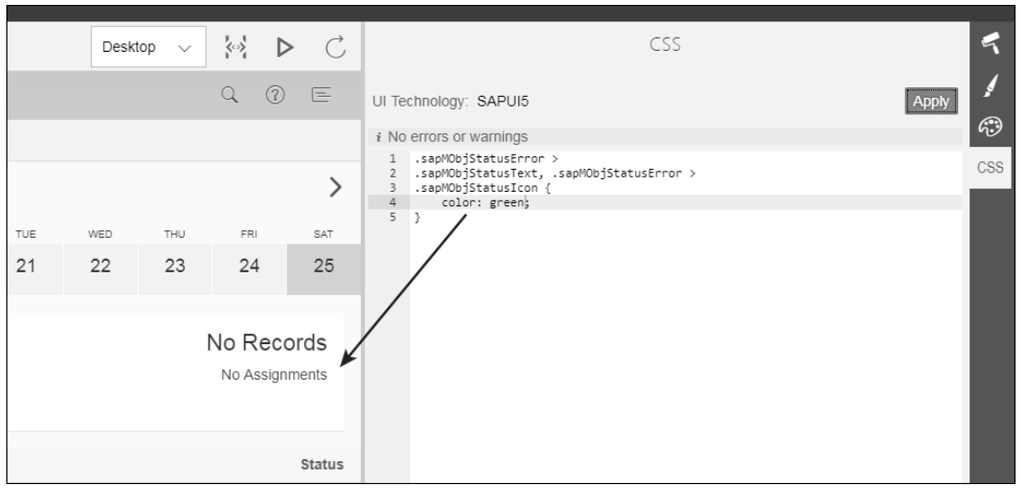


Figure 2.31 Using Custom CSS

Maintaining Themes

After a new theme is created in the development system, it needs to be transported to the testing and production systems so that real users can use it. You may also want to set it as a default theme for your users, so that users don't have to explicitly choose it. You can also export the theme from one system and import it into another. Let's explore all these possibilities in this section.

Setting the Default Theme

After a theme is created, it can be set as the default theme for all users so that they need not set the theme from the SAP Fiori menu. To set a default theme, you use Transaction /UI2/NWBC_CFG_CUST in the following steps:

1. Navigate to **Change Mode** ❶, and then click on **New Entries** ❷ in Transaction /UI2/NWBC_CFG_CUST, as shown in Figure 2.32.
2. In the **New Entries** screen, enter "SAP_FLP" in the **Filter** field, enter "Theme" in the **Parameter Name** field ❸, and press Enter.
3. This will open the **Value** field for input. Enter the technical name of the new theme that was created.
4. Click **Save** to save the entry. This setting will default the Fiori_Demo as the new theme for all the users. However, users can go to SAP Fiori menu options and choose a different theme to override this setting.

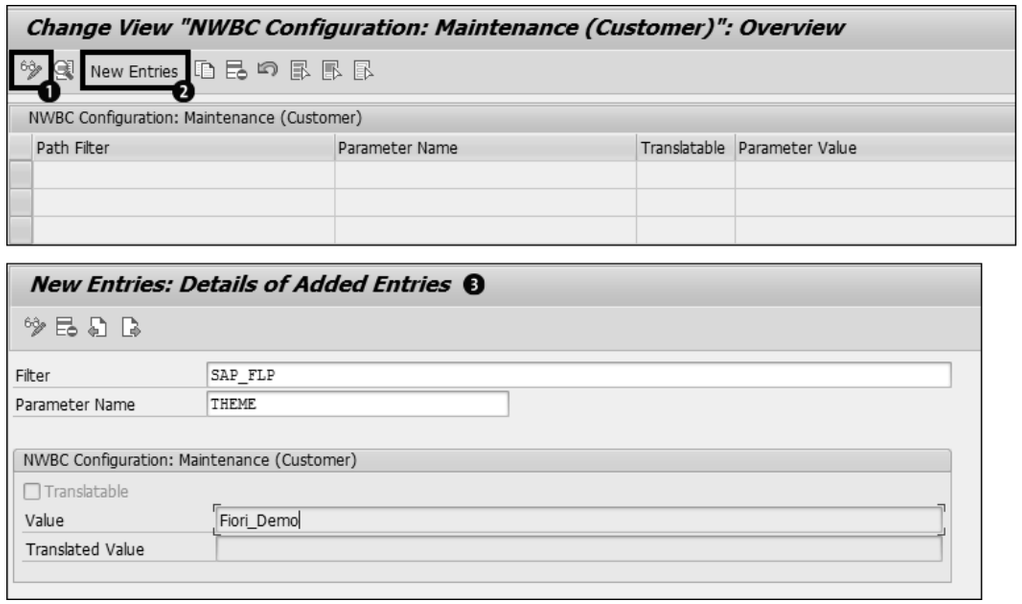


Figure 2.32 Setting the Default Theme

Tip

SAP sets a default theme initially using Transaction /UI2/NWBC_CFG_SAP. However, this should not be changed because further SAP upgrades will overwrite the changes.

Transporting Themes

After completing the development and testing of the theme in the development server, it's a normal requirement to transport the theme to other servers in the landscape. To transport the themes, you need to go to generic Transaction /UI5/THEME_TOOL. This tool will list each of the custom themes created in the server. Against each theme name, it provides options such as **Info**, **Transport**, **Download**, and **Delete,** as shown in Figure 2.33.

To transport a theme, click on the **Transport** button, and then click on the magnifying glass, as shown in Figure 2.33 (or just double-click on the **Transport** button ❶). A standard dialog opens allowing you to choose or create a transport to hold the changes in the theme ❷. After selecting a transport, click on the **OK** button to close the dialog.

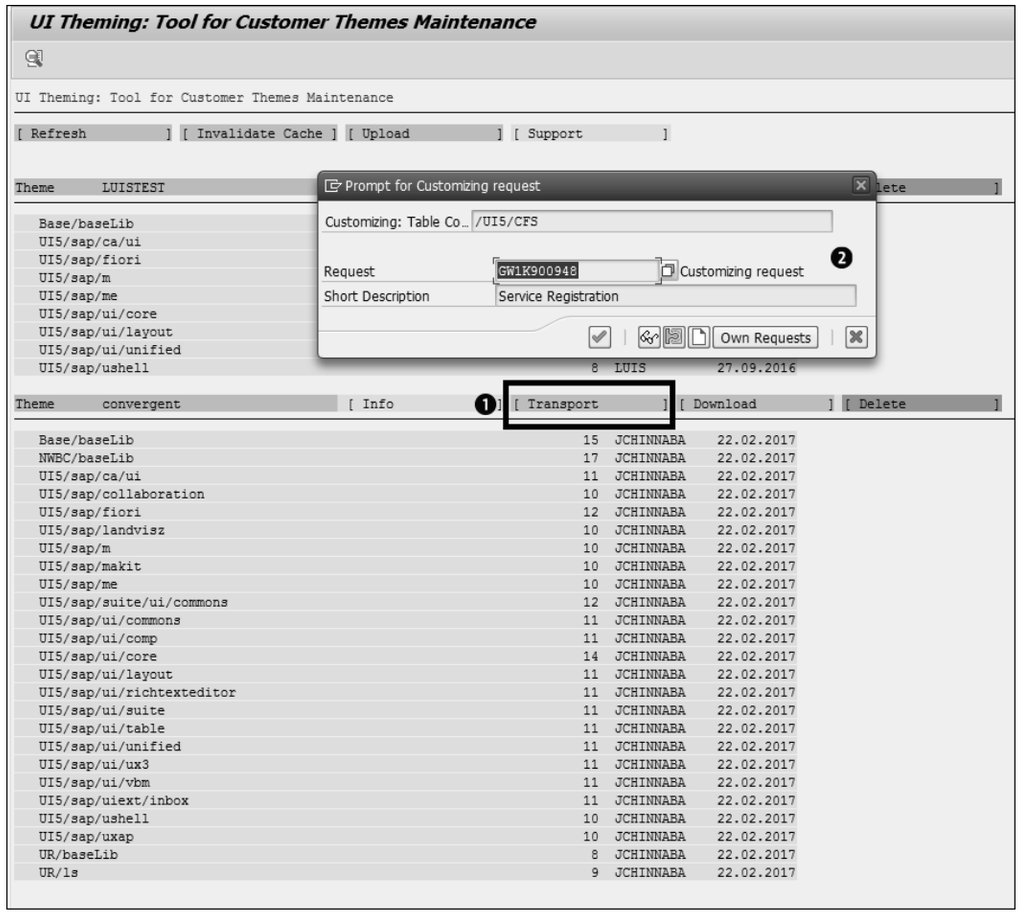


Figure 2.33 Transporting a Theme

Exporting/Importing a Theme

In some scenarios, there are no transport routes between two systems, and you need to manually repeat the theme changes. For such scenarios, SAP provides an export/import (**Download**/**Upload**) option (refer to Figure 2.33) through which you can achieve the desired changes.

To download a theme, double-click on the **Download** button. SAP will prompt you for a file name and location. Upon providing those details, the entire theme will be written to a file. Transport this file manually to a target system's file system. In the target system, click on the **Upload** button, and choose the previously downloaded file. This will create the new theme in the target system.

When you need details about the theme designer, such as its version and SAPUI5 version, you can use the **Support** button at the top of the screen. Double-click the button to see various related details, including the support component, as shown in Figure 2.34.

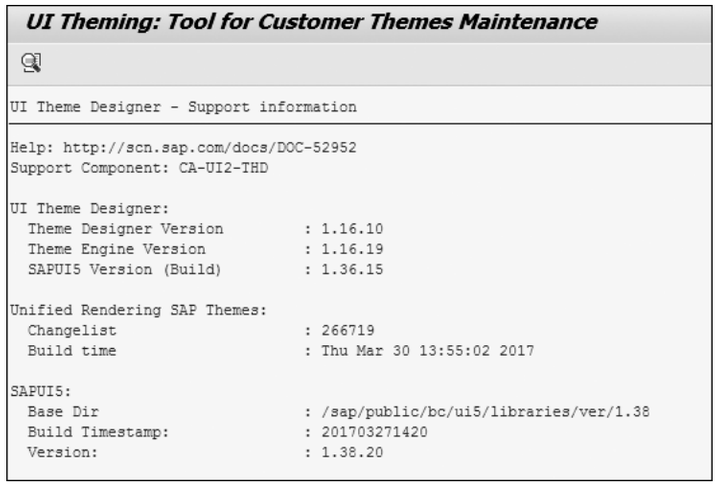


Figure 2.34 UI Theme Designer Support Information

Important Terminology

In this chapter, the following terminology was used:

- **Hub deployment**
  This is one of SAP Gateway's deployment options in which there is a separate system in the landscape for SAP Gateway that connects to the backend by Remote Function Call (RFC).
- **Embedded deployment**
  This is the other SAP Fiori-supported SAP Gateway deployment option in which SAP Gateway is installed as an add-on on the SAP ERP/backend system itself.
- **ABAP CDS views**
  ABAP CDS views are enhancements to the ABAP Dictionary concept, by which you can create semantically rich data models and use them directly from within ABAP programs.
- **Internal access point**
  This is one of the supported landscape options for SAP Fiori Cloud. In this option, business data always remains within the customer's network, and users will be able to access SAP Fiori only from within the customer's network.
- **External access point**
  This is another important landscape for SAP Fiori Cloud in which users can also access SAP Fiori from external places, that is, outside their corporate network.
- **UI theme designer**
  The UI theme designer is a tool provided by SAP, available on SAP Cloud Platform, on-premise, and SAP Enterprise Portal, which can be used to change the colors, logos, and backgrounds for various SAP UI technologies. It has four important features:

- Quick mode: In quick mode, you can edit most needed colors, backgrounds, and company logos. This is easy for everyone to use.
- Expert mode: This allows you to change the properties using semantic parameters. These semantic parameters will be reused throughout the UI; therefore, it's important to know and test the side effects.
- Palette: Palettes let you define your corporate colors and provide a name for reusing it, so that you don't have to define them every time you need them.
- CSS pane: This lets you override SAP theme colors by inserting custom CSS.

Practice Questions

These practice questions will help you evaluate your understanding of the topics covered in this chapter. The questions shown are similar in nature to those found on the certification examination. Although none of these questions will be found on the exam itself, they will allow you to review your knowledge of the subject. Select the correct answers, and then check the completeness of your answers in the "Practice Question Answers and Explanations" section. Remember, on the exam, you must select all correct answers and only correct answers to receive credit for the question.

1. Which of the following is *not* one of the advantages of SAP Web Dispatcher?

☐ A. Acts as a switch to allow or block certain HTTP requests
☐ B. Performs load balancing
☐ C. Replaces the SAP Gateway server
☐ D. Acts as a reverse proxy for SAP Fiori apps

2. What is true about an SAP Gateway server or frontend server in a hub architecture?

☐ A. It's the same as the SAP business system server.
☐ B. This is where OData implementations are coded.
☐ C. The SAP Fiori UI repository is located here.
☐ D. It can connect to only one SAP backend server.

3. In SAP Fiori architecture for SAP Business Suite systems, which type of application does *not* require SAP Web Dispatcher?

☐ A. Transactional apps
☐ B. KPI apps
☐ C. Fact sheet apps
☐ D. SAP Smart Business apps

4. Which of the following is *not* one of the advantages of the hub deployment architecture?

☐ A. Lower TCO
☐ B. Better security
☐ C. Routing to multiple backend business systems
☐ D. Separation of innovation lifecycles

5. Which of the following describe the OData provisioning service? (2 correct answers)

☐ A. Provides an SAP Fiori UI repository to store app-specific UIs
☐ B. Provides a way to register and expose OData services
☐ C. Lowers TCO of the SAP Gateway landscape
☐ D. Used in the internal access point scenario of SAP Fiori Cloud

6. UI theme designer is available in which platforms? (3 correct answers)

☐ A. SAP Cloud Platform
☐ B. SAP Mobile Platform
☐ C. SAP Enterprise Portal
☐ D. SAP ABAP Server (frontend server)

7. UI theme designer cannot be used to perform which of the following?

☐ A. Use a CSS editor to change CSS properties to affect an SAPUI5 application.
☐ B. Add background images to tiles using **Expert** mode.
☐ C. Add a custom logo to SAP Fiori launchpad.
☐ D. Change the background color of a button when it's hovered over.

8. Transaction /UI5/THEME_TOOL does *not* offer which of the following capabilities?

☐ A. Upload
☐ B. Download
☐ C. Transport
☐ D. Copy

Practice Answers and Explanations

1. Correct answer: **C**
   SAP Web Dispatcher can't replace SAP Gateway server because it has its own set of functionalities. Other answer options are the benefits of using SAP Web Dispatcher.
2. Correct answer: **C**
   In a hub architecture, SAP Fiori app-specific UIs are stored in the ABAP BSP repository.
3. Correct answer: **A**
   Transactional apps don't require SAP Web Dispatcher because all the HTTP connections required always end up at the SAP Gateway server. Other application types need to contact more than one server.
4. Correct answer: **A**
   Due to additional SAP Gateway systems, TCO is higher in the hub architecture. Thus, "Lower TCO" isn't one of the advantages. The rest of the options are advantages of the hub deployment option.
5. Correct answers: **B, C**
   The OData provisioning service provides a way to register and expose OData services. By moving this feature to SAP Fiori Cloud, it reduces the TCO of the SAP Gateway landscape. In the SAP Fiori Cloud landscape, the SAP Fiori UI repository on SAP Cloud Platform is where app-specific SAP Fiori UIs are stored. In the internal access point scenario, OData provisioning isn't used because one of the aims is to not let the business data pass through SAP Cloud Platform.
6. Correct answers: **A, C, D**
   UI theme designer is currently available only in three platforms. It's not available in SAP Mobile Platform.
7. Correct answer: **B**
   A background image can't be added to a tile using **Expert** mode. Instead, you need to use the CSS section of the UI theme designer, which isn't recommended due to the high TCO involved.
8. Correct answer: **D**
   Transaction /SAPUI5/THEME_TOOL doesn't offer copy functionality. This can be performed in the UI theme designer tool instead.

Takeaway

In this chapter, we explored the SAP Fiori architecture in SAP Business Suite systems. We then saw how the architecture evolved in the SAP S/4HANA landscape and discussed the one archetype concept.

Next, we saw the architecture in hybrid scenarios, including SAP Fiori Cloud. We saw a wide variety of options available such as internal and external access points.

We then discussed the deployment options within SAP Gateway and the advantages of each option. You learned how the OData provisioning service from SAP Cloud Platform can be used to lessen the TCO.

Next, we covered how SAP Fiori launchpad is configured using catalogs, groups, and roles to determine the assignment of tiles to the user and the entry page. Finally, we discussed how you can use the UI theme designer to customize SAP Fiori with corporate colors and logos.

Summary

SAP Fiori comes with various architectural options and landscapes, and understanding all the options is very important to zero in on the right approach for your scenario. In the next chapter, you'll learn about SAPUI5 foundations and how generic UIs are built for SAP Fiori apps.

Contents

Index

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Krishna Kishor Kammaje is currently a developer and architect working at ConvergentIS, SAP products and consulting provider. He was recognized as an SAP Community top contributor for SAP Fiori and SAP Gateway for several years and was named an SAP Mentor in March 2017. His latest interests are around machine learning and exploring ways to make SAP Fiori apps intelligent using those technologies.

Krishna Kishor Kammaje

SAP Fiori Certification Guide: Development Associate Exam

429 Pages, 2020, $79.95
ISBN 978-1-4932-2047-2

www.sap-press.com/5221