In this chapter excerpt, you’ll explore the complete API lifecycle. You’ll walk through the key SAP API Management components that support each phase, including the API designer, the API portal, the developer portal, and the API gateway.
Chapter 3
Managing the End-to-End Lifecycle of APIs

In this chapter, we look at the end-to-end lifecycle of application programming interfaces (APIs). After a short general discussion of API lifecycles, we’ll take a closer look at how SAP API Management supports that lifecycle. We’ll then run through a simple example that allows you to experience the different steps hands-on.

After reading the first chapters of this book, you should now have a good understanding of APIs as the building blocks of digital businesses and how they fit into the SAP strategy of an intelligent enterprise. In addition to this, you’ve read about how and why SAP API Management was implemented and how you can install and run it on your own instance of SAP Cloud Platform. In this chapter, we’ll take a closer look at the end-to-end lifecycle of APIs and at the tools SAP API Management offers to you in the lifecycle.

3.1 Overview of the API Lifecycle

Figure 3.1 shows the lifecycle of an API. As mentioned, APIs are used to create your digital business through apps and other channels.

Starting with channels in Figure 3.1, you see that several entities are mentioned, such as apps and business networks; however, this list isn’t complete as channels can also include social media and dedicated devices, among other things. In all cases, however, the APIs are the building blocks that allow you to address those channels.

APIs are well-defined interfaces that—in an ideal world—come with detailed documentation that allows every developer (right side of Figure 3.1) to build apps that support basically every channel. The developer doesn’t need to know anything about the backend or how the API is eventually implemented; with the information from the API, he or she can start developing right away.
And, as you can see from Figure 3.1, the data an API exposes can come from a lot of different API backends when you’re using SAP API Management—not only SAP backends but third-party backends too, as long as they support Representational State Transfer (REST)/OData or Simple Object Access Protocol (SOAP).

Figure 3.1 Lifecycle of an API

Now let’s take a closer look at the five phases of the lifecycle:

- **Design**
  In this phase, you design an API on the “coding” level.

- **Develop**
  In this phase, you implement your API using additional services and maybe even other APIs.

- **Manage**
  In this phase, you implement security and traffic management for your API. This is one of the most essential phases.

- **Meter and monetize**
  In this phase, you can meter your API and analyze its behavior, which will allow you to monetize it as well.

- **Engage**
  In this phase, you provide your API to the developers; the phase includes onboarding developers.

The aforementioned phases can be found in basically every API lifecycle. Sometimes, the phases may have different names, or there may be additional or extended steps, but, in general, the lifecycle will always look similar.

The following sections provide a closer look at what you can expect of the different phases in SAP API Management.

### 3.1.1 Design

The design phase actually may happen outside of SAP API Management. Given that you already have an existing API or suitable service somewhere in your backend, you can skip directly to the develop phase.

However, if you want to model your own API or maybe start with an existing API and make some changes to it on the specification level, this is the phase in which to do so.

SAP API Management provides the API designer tool supporting OpenAPI Specification (formerly known as Swagger) and RESTful API Modeling Language (RAML). In this phase, you can describe your own APIs, work on existing APIs, and eventually create your own API proxy in SAP API Management.

### 3.1.2 Develop

This phase is all about developing your own API. As with the next two phases, SAP API Management has you covered with the API portal. The API portal consists of various tools that you can use to build your own APIs.

In contrast to the design phase, you develop APIs on this level by working on existing APIs and/or services. You can easily build your own API proxies that rely on other APIs or on services. In the easiest implementation, your API will just reflect a service or API in the backend. In more complex scenarios, your API proxy can be based on several APIs and/or services as well as make calls out from within the policy editor, allowing you to include services in a complex configuration and implementation of your own API.
3.3 Manage
Managing an API is mainly about three things:
- Security
- Traffic management
- Mediation
Most likely, you’ll spend the bigger part of this phase in the policy editor, which allows you to apply policies to your APIs. With SAP API Management, you’ll get a set of predefined policies dealing with security, traffic management, and mediation. In addition, the editor will allow you to customize those policies, define your own policies, and even bundle them in templates.

3.4 Meter and Monetize
There are many reasons to check on your APIs and keep an eye on the traffic they generate and how they behave in general. Among the most prominent ones are these:
- Performance
- Security
- Monetization
If an API isn’t performing correctly, you may want to fix it, or if it isn’t really used, you probably want to remove it completely. In terms of security, you may want to check for any suspicious behavior in your system and analyze it. And, finally, you may want to know who is using your APIs and charge them for the usage.
All of these features are integrated into the API portal, where you can run analytics on your APIs and define rate plans to charge users of your APIs for the usage based on different criteria.

3.5 Engage
The last phase in the lifecycle should not be underestimated. Engaging with whomever is using your APIs is essential to the success of your APIs. This is not only true when your users are outside your company, but maybe even more so when they are inside your company.

3.2 Components of SAP API Management
By now, you should have a good understanding of what goes on in the different phases. In this section, we’ll look closer at the components mentioned earlier and what you can do with them.
Best practice is to get your SAP Cloud Platform trial account (see Chapter 2, Section 2.3.1) up and running and follow the explanations in your own instance. Please note that because we’re talking about a cloud implementation, you might find minor changes in the user interface (UI).

3.2.1 API Designer
The API designer is part of the SAP API Management service. After opening your SAP Cloud Platform cockpit, you can find the tool by clicking on Services and then Integration or by searching for it and clicking on API Management. After performing the steps in Chapter 2, your SAP API Management should already be enabled and look like the tile in Figure 3.2.

Figure 3.2 Enabled API Management Tile in the SAP Cloud Platform Cockpit
After clicking on **API Management**, your screen should now look like Figure 3.3.

![Figure 3.3 API Management with the Access API Designer Link](image)

On this overview page, you can find links to all the relevant tools, including the API designer. As you can see, the overview page also lets you know in which regions SAP API Management is available.

Let’s explore the API designer, beginning with the different options to access it and following with the key menus.

**Access**

After clicking on **Access API Designer**, the API designer will open as shown in Figure 3.4. At startup, the API designer opens with some skeleton coding that shows the basic entities to be expected in an OpenAPI description. With that said, don’t be concerned about the errors it shows on the right side, as the description is by no means complete. However, this is only one way to open the API designer; in your everyday work, you may prefer one of the others we’ll discuss next.

![Figure 3.4 API Designer at Startup](image)

Another even more convenient way to open the API designer in real life scenarios is by clicking on **Access API Portal** and then choosing **Develop** from the navigation pane in the API portal (see Figure 3.5).

![Figure 3.5 Accessing the API Designer from the API Portal](image)

A screen with a list of all existing APIs will open. From this screen, you can open the API designer by clicking on **Create in API Designer** (see Figure 3.6).
Managing the End-to-End Lifecycle of APIs

3.2 Components of SAP API Management

File

The File menu (see Figure 3.8) is probably the menu you’ll use the most. It allows you to generate new files (New) that will look exactly like the one you’ll see when you open an empty API designer.

Figure 3.8 The File Menu in the API Designer

The next three entries starting with Paste allow you to paste JavaScript Object Notation (JSON), RAML, and OData metadata into your file. By clicking on one of these entries, a small editor window will open. Figure 3.9 shows the editor for JSON. Note that the editor checks the coding for validity.

Figure 3.9 Pasting JSON Coding into Your File
In addition to pasting coding into your file, you can import complete files by clicking the **Import File** menu item. Accepted files include YAML and JSON formats. By clicking **Save**, you can create or update your API in SAP API Management. If you create an API from scratch, the API designer will ask you to provide a name. This new API will then show up in your list of APIs in the API portal. Finally, you export your APIs in YAML or JSON format by clicking on **Download YAML** and **Download JSON**, respectively.

### Preferences

The **Preferences** menu allows you to change the look and feel of the API designer according to your personal needs and preferences. In addition to setting the font size (**Font Size**) and going back to the default setting (**Reset Editor Settings**), you can make very specific changes through **Editor Settings** (see Figure 3.10). When clicking on **Editor Settings**, the possible settings (shown in Figure 3.11) will open on the right side of your API designer screen. As you can see, there are a lot of settings possible. It would go a bit too far to explain them all here, but most are self-explanatory. Feel free to play around with those settings until you find the user experience matching your needs.

The last menu item in **Preferences** allows you to modify the behavior of the API designer even further. As you can see from Figure 3.12, you can mainly change the “live” behavior of the editor here. This may be helpful if you encounter specific issues with live rendering or auto complete.

Usually, the preset settings will do just fine, but for more complex specifications, the features may be bothersome, and you probably want to turn **Live Rendering** and/or **Auto-complete** off. The first might make rendering the code slower, while the second might be tricky if you use coding with a lot of nonstandard expressions.

![Figure 3.10 Creating a Convenient Development Environment through Preferences](image)

![Figure 3.11 Editor Settings for the API Designer](image)

![Figure 3.12 Editor Preferences for the API Designer](image)

---

3.2 Components of SAP API Management
Generate Server

The Generate Server menu, shown in Figure 3.13, allows you to easily generate server stubs. You can choose between JAX-RS, Node.js, and Spring. With this feature, the API designer supports the complete lifecycle of an API.

![Generate Server menu](image)

**Figure 3.13** Generating Server Stubs in API Designer

Figure 3.14 shows an example dialog for a Node.js server stub generation. After changing the default entry of Artifact (recommended), you can now create your own server stub by clicking on Generate Project. A nodejs-server.zip will be created and downloaded or offered for download by your browser.

This file now represents your application that you can upload and use in other environments such as Cloud Foundry in SAP Cloud Platform.

![Generating a Node.js server Stub from the API Designer](image)

**Figure 3.14** Generating a Node.js server Stub from the API Designer

Help

The Help menu (see Figure 3.15) provides a link to the Open API Initiative (OAI) and allows you to go directly to the OpenAPI Specification GitHub by choosing Open API Specs.

![Help menu](image)

**Figure 3.15** Help in API Designer

The Show Introduction Help option provides you with an overlay that is pretty basic but is intended to help you get the lay of the land when it comes to getting started with the API designer (see Figure 3.16).

![Help with First Steps in API Designer](image)

**Figure 3.16** Help with First Steps in API Designer

3.2.2 API Portal

While the API designer is all about specifications and coding, the API portal takes a different approach and provides you with an interface that allows you to create, configure, and manage your APIs. Figure 3.17 shows the landing page of the API portal with an expanded “hamburger” menu. As you can see, the landing page provides you with an overview of what is happening with your APIs at a glance.
You can reach the most relevant actions directly from the landing page (Quick Actions) and see your API Traffic, how many errors happened (API Errors), how many of your APIs are already deployed (APIs), and much more, including the performance of the APIs, applications, products, and rate plans used for the monetization of your APIs.

We’ll go through the different tools in more detail in Section 3.3, so we’ll focus on the main menu options of the API portal here.

Discover
As the building blocks of your digital business, APIs can in many cases be taken quite literally. In other words, you build APIs in many scenarios that are based on already existing APIs and combine them in a way that works for you. Or you just consume an API, connect it to your backend, and then expose it through SAP API Management.

Unfortunately, in the real world, more often than not, you don’t know if a functionality or service is already implemented somewhere else or if there is an existing API that you could reuse.

SAP helps you with this problem by providing a central repository for SAP, partner services, and APIs called the SAP API Business Hub. You can reach this repository through https://api.sap.com.

However, just providing a URL isn’t a very integrated approach if you already offer an SAP API Management solution. This is where the Discover page comes in to provide a direct connection to the SAP API Business Hub repository from within the API portal (see Figure 3.18).

While the landing page provides you with the highlights of SAP API Business Hub, you can access all available resources by clicking on All. Figure 3.19 shows the list of available resources. As you can see, it can become confusing or overcrowded very quickly. Therefore, in addition to offering a search functionality, you can filter the entries by Lines Of Business (LoBs), Vendor, and so on.

After choosing a specific service, you’ll get detailed information for that service and can easily find out if it fits your needs. Figure 3.20 shows an example for SAP Leonardo Machine Learning Foundation. Note that you can go directly to the SAP API Business Hub from here by clicking on the View in API Business Hub link in the upper-right corner.
While additional information, such as documents with product information or links to further resources, might be found by clicking on Documents, you’ll likely investigate Artifacts most of the time. Through Artifacts, you can access the services that are offered as shown in Figure 3.21. As you can see, there is a button under Actions that you can click to open a context menu that allows you to copy the API (in this case, a REST API) directly into your SAP API Management instance.
After clicking on OK, your list of APIs will open and show you a new entry (see Figure 3.23). You’ve now copied the API into your SAP API Management instance and can use it as any other API in your list.

Through the Develop page, you can create your own APIs from scratch, import APIs from a file, or create them in the API designer. Then you can create products that usually are based on APIs. Applications are built in the developer portal, so you can’t create them here, but you can have a look at them, see who developed them, and get additional details on the app. Policy templates aren’t created here either; you can only import them from here. If you want to create a policy template, you have to do that in the policy editor.

Let’s take a closer look at the API itself. As you can see from Figure 3.25, the Overview page for an API gives you all the relevant information on an API. However, on this screen, you can also find some of the most relevant tools when dealing with your APIs.

While Proxy EndPoint and Target EndPoint give you more information on the API that you can use, the Resources tab gives an overview on the methods you can invoke on your APIs. As you can see from Figure 3.26, you not only get the method name but also what kind of method it is (Post), and you can try it out (choose Try out).
3 Managing the End-to-End Lifecycle of APIs

By clicking on the method, you can now expand it, get even more information on the method, and even set parameters before you try it out (see Figure 3.27).

You may have noticed that you can only try out your API here but can’t make any changes. To change your API, and especially the resources, you can open the sub-menu of Edit in the upper-right corner of your screen (see Figure 3.28).

The most important is the pencil icon, which will open an additional dialog when clicked. Figure 3.30 show the Edit Resources dialog that will open with an opened sub-menu. As you can see, you can change the name, the Path Prefix (basically the relative location of the resource in your SAP API Management instance), and even the Operations and the type (i.e., you can add and remove operations depending on your needs and what you want to allow a developer to do with your API).

In addition, you can add extensive documentation for your API and methods/operations here. As you can see, the dialog comes with an editor that allows you to create your own documentation, including different font styles, formatting, links, and so on.
While in Edit mode, the upper-right corner will offer different links/buttons, as shown in Figure 3.31. Here you can Save your changes or Cancel them, and all changes are void.

Warning

Even though you’ll get a warning, when trying to leave the page without saving, your changes will only be saved after clicking on Save. If you leave the page in an unexpected way (e.g., by closing the browser), your work will be lost.

After saving your changes, you can then return to the original links, including the aforementioned Policies, Copy, and Edit in the upper-right corner. While Copy simply allows you to copy your API (see Figure 3.32), which, in many cases, means just giving it a new name and then working on the copy, Policies is a much more interesting option as it opens the policy editor (see Figure 3.33).
The policy editor allows you to apply predefined policies on your APIs and even create your own policies to manage the behavior of your APIs. To use it, you first must click on Edit (see Figure 3.33).

As the policy editor will be explained in detail in Chapter 5, let’s just have a look at the UI and what it offers. Starting on the right, you see a set of policies you can use:

- **Security Policies**
- **Traffic Management Policies**
- **Mediation Policies**
- **Extension Policies**

The first three are predefined policies deal with a specific topic (e.g., Security), and you can configure them to your specific needs. The last one (Extension Policies) allows you to build your own policies.

If you already clicked on Edit, you’ll realize that the plus sign on the right side is still gray (i.e., inactive); this is because you haven’t yet decided to which flow you want to add the policy. The flow defines when a policy is executed. For example, click on *Pre-Flow* beneath ProxyEndpoint on the right (see Figure 3.34).

![Figure 3.34 Choosing a Flow](image)

The plus icon is now available, so you can add policies to the flow. Figure 3.35 shows an added policy (BasicAuthentication).

![Figure 3.35 Added Policy in the Policy Editor](image)

You may have noticed the three points in the Edit menu of your API that weren’t mentioned yet. Figure 3.36 shows them expanded. Under this menu entry, you can find the following:

- **Debug**
  - Allows you to access the debug functionality discussed under Test.
- **Deploy**
  - Deploys an API after it’s thoroughly tested.
- **Export**
  - Allows you to export your API to a file.
- **Delete**
  - Deletes your API.
Because dealing with APIs is so essential, we’ve taken the time to have a closer look. However, the functionality for products and applications is pretty simple and will be discussed in Section 3.3. Policy templates will be discussed in Chapter 5 when we take a deep dive into the policy editor. Now, let’s switch to the Configure page.

**Configure**

The Configure page (see Figure 3.37) allows you to deal with three entities:

- **API Providers**
  Allows you access to predefined API providers such as backend systems (e.g., an SAP Gateway system).

- **Certificates**
  Allows you to create/upload security certificates into SAP API Management.

- **Key Value Maps**
  Allows you to store (encrypted) key value maps.

**Figure 3.37 Configure Page in the API Portal**

API Providers (see Figure 3.37) are basically a representation of any kind of API provider, which means that you can create a representation here that is pointing to a backend system (e.g., an SAP Gateway system) or a specific service (OData/REST, SOAP).

Certificates also play an important role in securing your APIs. More specifically, they allow you to use the Secure Sockets Layer (SSL) to establish an encrypted connection between a web server and a web client. In other words, it allows you to use HTTPS instead of simple and unsecure HTTP.

As certificates and security in general are a rather complex topic, you can find the details on this topic in Chapter 8. Figure 3.38 shows the dialog you can use to create certificates (after clicking on Configure > Certificates > Create). As you can see, you can choose between Trust Store and Key Store certificates (for details, see Chapter 8).

**Figure 3.38 Creating Certificates in the API Portal**

For backend systems, it usually makes sense to create an API provider and later use this provider in the API creation; however, for services from a certain point of view, it only makes sense if you plan to use the service in more than one API as you can point to a specific service directly when creating an API.

**Note**

We recommend implementing specific rules for the API product manager and developer that ensure a certain consistency in your SAP API Management instance. One rule could be that you have to create an API provider for every service and backend that you’re using in an API, after checking that there is no duplicate in your list of API providers.
Key Value Maps allow you to create a set of key/value pairs, which come in handy when you need access to data during runtime and you don’t want this data to show up in your API proxy logic (i.e., any kind of coding).

One scenario is that you connect through SAP API Management to other systems that may not support the single sign-on (SSO) mechanisms of SAP API Management or that you want to connect to a backend system using a service user. In such a scenario, you can create a key value map that stores, for example, the user credentials for the service user. When configuring your API proxy logic, you can now access this information and use it as part of your coding. For example, you can allow or deny a user access to the backend system based on the information that user already provided to SAP API Management.

To access the dialog you can use to create key value maps (see Figure 3.39), choose Configure → Key Value Maps → Create (see Chapter 7, Section 7.5 for further details).

![Figure 3.39 Creating Key Value Maps in the API Portal](image)

Monetize

The Monetize page (see Figure 3.40) gives you access to the following:

- **Rate Plans**
  A rate plan allows you to charge for the usage of your APIs by attaching a rate plan to a product.

- **Bills**
  The bills and billing details for specific users (usually developers) in a specific month are provided.

![Figure 3.40 API Portal Access to the Relevant Monetization Tools](image)

The Rate Plans tab allows you to charge for the usage of your APIs. As you can see in Figure 3.41, the rate plan has an associated product; that is, a rate plan is always attached to a product.

![Figure 3.41 Creating a Rate Plan in the API Portal](image)
You can add a basic charge for the usage and/or charge by API call. In addition, you can define different costs depending on the overall usage (e.g., give a discount if a certain number is reached). For more details, see Chapter 9.

Bills (see Figure 3.42) are generated automatically per user (i.e., developer) using your APIs. Access to your APIs is always established through the products you created, and you can define your own rate plan for every product. Based on those rate plans, SAP API Management automatically calculates the costs for specific developers (see Figure 3.42) and generates a bill. You can access those bills for any given month. For more information on monetization and bills, see Chapter 9.

![Figure 3.42 Example for a Bill in the API Portal](image)

Analyze
The Analyze page (see Figure 3.43) allows you to run analytics on your APIs. By default, you'll already see the most relevant information such as API Response Time, Total API Calls, or Total API Errors. While some of the information is available as simple numbers, other information is displayed as a graphic.

![Figure 3.43 Analyze: Analytics for Your APIs in the API Portal](image)

To make things easier, you can access this information for predefined time periods: Last 6 Months, Last 30 Days, Week, Day, and Hour. However, if you want to look into a different time period, you can define it through the Custom option. For detailed information on how to use analytics on your APIs, see Chapter 11.

Test
The Test page (see Figure 3.44) provides a test console for your APIs. As you can see, you can choose your API on the left and then test it. Through the test console, you can
not only call the API but also define the method type to be invoked and provide headers and URL parameters to be used when calling your API. For example, if an API key is required in the header information, you can provide it here. In addition, you can provide authentication information during the call using the Authentication: None link.

Figure 3.44  Testing Your APIs in the SAP API Portal

**Note**

Although most developers prefer their own tools to test their APIs, using the API URL in other tools, such as Postman, is supported and is an alternative or addition to the test console in the API portal.

Through the Test page, you can also start the debugger for your APIs (Debug button). Figure 3.45 shows a running—yet empty—debugger. In a live system with relevant usage, you would directly see any traffic on the API and the debug information. If you use a test system, use another tab or any app to call your API to see what is happening in the debugger. Because debugging is expensive in terms of resources, the debugging will automatically stop after 10 minutes.

Figure 3.45  A Started—Yet Empty—Debugger in the API Portal

### 3.2.3 Developer Portal

The developer portal (see Figure 3.46) is specifically designed for the onboarding of your developers, that is, the people you expect to use your APIs. While the developer portal can be used internally, it’s also designed to be used externally as an entry point for external developers and partners. So, if you want to expose your APIs to the outside world, for example, through api.yourcompanyname.com, this web address might point to the developer portal.

**Note**

We show the developer portal in the predefined SAP design. Customizing the developer portal to fit your needs and corporate identity is possible though, as mentioned in Chapter 2, Section 2.1.
Let’s look closer at Figure 3.46. Basically, there are four main links, which we’ll explore further in the following sections:

- **Home**
  This link takes you back to the landing page.

- **Manage**
  Depending on your role, you may or may not see this link as it allows you to manage your developer portal.

- **My Workspace**
  This is where the developer finds an overview of all his applications and—if applicable—any involved costs.

- **Test Console**
  This link allows you to access the SAP API Management test console in a similar fashion as in the API portal.

The Home page (refer to Figure 3.46) is the landing page for your developer portal and displays, by default, the APIs (i.e., products in SAP API Management terminology) available to a developer. As you can see on the upper right, the developer has to be logged in, hence the greeting and the Logout option. The API developer/manager/administrator can decide which APIs a developer will see based on the user and the assigned roles.

### Note

Assuming that our readers have a keen eye, we don’t want to hide the Classic Design link on the screenshot. At the time of writing (fall 2019), the link would have taken you to the classic design of the developer portal. However, by the time you read this book, the link should be gone.

The Home page not only allows a developer to display existing APIs/products but also allows a developer to view the details directly and, more importantly, subscribe to a product (see Figure 3.47).
Clicking on **Subscribe** in Figure 3.47 will open a popup that allows you to choose between **Create New Application** and **Add to Existing Application**. Note that this means that your applications can be based on more than one product.

**Manage**

The **Manage** page (see Figure 3.48) helps you manage your users, that is, developers. You can use the following capabilities:

- **E-mail Configuration**
  Enter an email address that will get information on all incoming and pending user requests.

- **Pending Requests**
  View all pending user requests, including some additional information and relevant actions for the administrator.

- **Registered Users**
  View all already registered users, including some additional information and all relevant actions for administrators to deal with users.

All links will take you to a central page on which you can directly deal with the configuration and the pending and registered users.

**Note**

Probably the most common way to set up your developer portal is with a self-registration option. This means when a user not yet known to the system navigates to your developer portal, he will get a link to get registered. This will generate a request and—if configured correctly—your administrator will get an email that there is a pending request to be dealt with. In addition, note that we recommend using a generic email handle for the administrator email instead of a specific person (e.g., administrator@yourapidevportal.com).

**My Workspace**

The **My Workspace** page (see Figure 3.49) gives you an overview of all existing applications and allows you to deal with applications in general. The most common action is probably available through the plus icon, which allows you to create a new application based on existing products. If you don’t need an application anymore, you can use the little trash can icon to delete the application.
Creating new applications or adding a product to an application can be done directly on the landing page too. By clicking on any product on the landing page, you can add that product to an existing application or create a new application based on that product.

Through My Workspace, you can also access information on how your applications are performing (Performance Analytics) and determine whether there are any errors (Error Analytics). Assuming that everything is running smoothly, the most interesting option for most developers will be the costs (Cost). Figure 3.50 shows an example of what the cost report could look like.

If you think that this can’t be all, you’re right. In addition to the development itself, which isn’t considered in this chapter (see Chapter 10 for details on that), developers will spend some time in the application screen itself. By simply clicking on an application, you can access the application itself.

Figure 3.51 shows the application screen for the Procurement Application. As you can see, you get an overview of the general Application Info, any Products the application is based on, and application-specific Analytics. One of the most relevant pieces of information, the Application Key, is stored in the Application Info. This key, sometimes referred to as the API key, is a necessary prerequisite to access most APIs/products.
API portal; however, it basically offers the same options to test your APIs minus the debugging capability.

![Test Console Page in the Developer Portal](image)

**Figure 3.52 Test Console Page in the Developer Portal**

### 3.2.4 API Gateway

The API gateway underlies all lifecycle steps from *manage* to *engage*. In short, the API gateway (yes, the name might be a bit misleading), is the runtime component for SAP API Management. Then again, the component supports the implementation of all the features of SAP API Management from security through traffic management to monetization. So, in the end, it kind of is the API gateway.

We’ll take a closer look at what you can do with it in the next section.

### 3.3 Lifecycle of APIs

Let’s now close the circle by diving into the lifecycle of an API by running through the complete process in the system using some example data.

We’ll walk through a simple API creation starting with an SAP Gateway system as an API provider.

#### 3.3.1 Create an API Provider

As mentioned before, it isn’t necessary to create an API provider in every instance. We do, however, recommend it in many cases because it’s cleaner and helps other users understand existing APIs easier. This is especially true if there is the need or possibility to reuse the API provider.

For our example, we’ll use the ES5 demo system that is available to everyone and gives you access to a SAP Gateway system for testing and developing (which you’ll remember from Chapter 2, Section 2.3).

**Note**

In Chapter 4, Section 4.3, we’ll discuss API providers in more detail. We’ll walk through an on-premise backend example and take a look at further options.

Given that you have a user in the ES5 system, you can now focus on creating an API provider. To create an API provider, you can either follow the path **Configure • API Provider • Create** or simply click on **API Provider** on the home screen of your API portal (in **Quick Actions**).

Either way, the resulting screen will be as shown in Figure 3.53. On the **Overview** tab of the **Add API Provider** screen, you can provide a name for your API provider and a description. In this case, use "ES5" as the **Name** and enter whatever you feel is appropriate for the **Description**.
Contents

Foreword ........................................................................................................................................ 15
Preface ........................................................................................................................................ 17
Acknowledgments ....................................................................................................................... 19

PART I Getting Started

1 Introduction ................................................................................................................................ 23

1.1 Trends in Digital Business ...................................................................................................... 24
  1.1.1 Emergence of the Intelligent Enterprise ................................................................. 25
  1.1.2 The Experience Economy ......................................................................................... 25
  1.1.3 Business Networks, Digital Ecosystems, and Collaboration .................................. 26
  1.1.4 Outcome-Based Business Models ............................................................................ 27

1.2 APIs as Digital Building Blocks ......................................................................................... 28
  1.2.1 Accelerate Innovative Digital Apps ........................................................................ 29
  1.2.2 Real-Time Interactions with Business Networks .................................................. 30
  1.2.3 Omni-Channel Access and Single Source of Truth .............................................. 31
  1.2.4 Business Model Innovations and Monetization ..................................................... 31

1.3 Why Manage APIs? .............................................................................................................. 33
  1.3.1 Harmonized, Secure, and Simplified Access .......................................................... 33
  1.3.2 Traffic Management ................................................................................................. 35
  1.3.3 Governance and Reuse .......................................................................................... 36
  1.3.4 Usage Analytics and Monitoring ............................................................................ 38
  1.3.5 Metering and Monetization ................................................................................... 39

1.4 SAP API Management: A Full-Lifecycle API Management Platform ......................... 39
  1.4.1 End-to-End API Lifecycle ...................................................................................... 40
  1.4.2 API Lifecycle Personas ......................................................................................... 41

1.5 Related and Complementary Products .............................................................................. 42

1.6 Summary ................................................................................................................................. 43
2 SAP API Management at a Glance 45

2.1 Architecture ............................................................................................................. 45

2.2 Introduction to the Real-World Scenario ................................................................. 48
  2.2.1 Retail ....................................................................................................................... 48
  2.2.2 Food Services ......................................................................................................... 49
  2.2.3 Health Care ............................................................................................................ 50
  2.2.4 Utilities .................................................................................................................. 50
  2.2.5 High-Tech ............................................................................................................. 51
  2.2.6 Pharmaceuticals .................................................................................................... 52

2.3 Deploying and Running Your First API Proxy ......................................................... 52
  2.3.1 Prerequisites ......................................................................................................... 53
  2.3.2 Set Up the Landscape and the Technical Conditions ........................................... 53
  2.3.3 Create the API Proxy ........................................................................................... 54
  2.3.4 Manage the API Proxy with Policies .................................................................. 58
  2.3.5 Deploy .................................................................................................................. 61

2.4 Summary .................................................................................................................. 62

3 Managing the End-to-End Lifecycle of APIs 63

3.1 Overview of the API Lifecycle .................................................................................. 63
  3.1.1 Design .................................................................................................................... 65
  3.1.2 Develop ............................................................................................................... 65
  3.1.3 Manage ................................................................................................................. 66
  3.1.4 Meter and Monetize ............................................................................................ 66
  3.1.5 Engage ................................................................................................................... 66

3.2 Components of SAP API Management .................................................................... 67
  3.2.1 API Designer ....................................................................................................... 67
  3.2.2 API Portal .......................................................................................................... 75
  3.2.3 Developer Portal ................................................................................................. 95
  3.2.4 API Gateway ....................................................................................................... 102

3.3 Lifecycle of APIs ...................................................................................................... 103
  3.3.1 Create an API Provider ....................................................................................... 103
  3.3.2 Create an API Proxy .......................................................................................... 106

4 Developing APIs 123

4.1 API Development Methodology .............................................................................. 123

4.2 Access the API Designer .......................................................................................... 125

4.3 API Providers ........................................................................................................... 126
  4.3.1 Create .................................................................................................................... 127
  4.3.2 Import ................................................................................................................... 134

4.4 Develop New APIs .................................................................................................. 135
  4.4.1 Microservices ..................................................................................................... 136
  4.4.2 Software Development Kit ................................................................................ 148
  4.4.3 SAP Cloud Application Programming Model ................................................... 150

4.5 Add Resilience to Your APIs .................................................................................. 150

4.6 Summary .................................................................................................................. 152

5 Managing Enterprise APIs 153

5.1 Policy Editor Basics .................................................................................................. 153

5.2 Using Policies ........................................................................................................... 156

5.3 Traffic Management ................................................................................................. 158

5.4 Mediation .................................................................................................................. 160

5.5 Security ...................................................................................................................... 162
# Contents

6 Traffic Management  
6.1 Access Control .......................................................... 168  
6.2 Concurrent Rate Limit .................................................. 170  
6.3 Quotas ...................................................................... 177  
6.4 Response Cache .......................................................... 181  
6.5 Spike Arrest ................................................................ 184  
6.6 Summary ................................................................... 186  

7 Mediation ...................................................................... 187  
7.1 Assign Message .......................................................... 188  
7.2 Extract Variables .......................................................... 195  
7.3 JSON to XML ............................................................... 200  
7.4 XML to JSON ............................................................... 205  
7.5 Key Value Map Operations .......................................... 208  
7.6 Raise Fault .................................................................. 212  
7.7 Service Callout ............................................................. 214  
7.8 Summary .................................................................... 217  

8 Security ......................................................................... 219  
8.1 Authentication ............................................................ 220  
8.1.1 Verify API Key ........................................................ 220  
8.1.2 OAuth .................................................................... 222  

9 Monetizing APIs with API Products .................................... 251  
9.1 Create API Products and Rate Plans ......................... 252  
9.1.1 API Products .......................................................... 252  
9.1.2 Create a Rate Plan ................................................... 257  
9.1.3 Subscribe to API Products ....................................... 261  
9.2 View Bills .................................................................... 263  
9.3 Summary .................................................................... 265  

PART III Consuming APIs and API Analytics

10 Consuming APIs ............................................................ 269  
10.1 Building SAP Fiori Applications with SAP Web IDE .......... 270  
10.1.1 Prerequisites .......................................................... 270  
10.1.2 Develop Your API ............................................... 271  
10.1.3 Build Your Application .......................................... 277  
10.2 Building Native Applications with SAP Cloud Platform Mobile Services ............................................. 290  
10.2.1 Prerequisites .......................................................... 290  
10.2.2 Develop Your API ................................................... 291  
10.2.3 Build Your Application .......................................... 295  
10.3 Generating Integration Flows from APIs in SAP Cloud Platform Integration ................................................. 305  
10.4 Summary .................................................................... 309  

11 Contents

5.6 Extension .................................................................... 164  
5.7 Summary .................................................................... 166  

8.1.3 Basic Authentication .................................................. 229  
8.1.4 SAML ...................................................................... 232  
8.2 API Security Threats .................................................... 241  
8.2.1 XML Threat Protection ............................................. 241  
8.2.2 JSON Threat Protection ............................................ 246  
8.3 Summary .................................................................... 248  

9 Monetizing APIs with API Products .................................... 251  
9.1 Create API Products and Rate Plans ......................... 252  
9.1.1 API Products .......................................................... 252  
9.1.2 Create a Rate Plan ................................................... 257  
9.1.3 Subscribe to API Products ....................................... 261  
9.2 View Bills .................................................................... 263  
9.3 Summary .................................................................... 265  

PART III Consuming APIs and API Analytics

10 Consuming APIs ............................................................ 269  
10.1 Building SAP Fiori Applications with SAP Web IDE .......... 270  
10.1.1 Prerequisites .......................................................... 270  
10.1.2 Develop Your API ............................................... 271  
10.1.3 Build Your Application .......................................... 277  
10.2 Building Native Applications with SAP Cloud Platform Mobile Services ............................................. 290  
10.2.1 Prerequisites .......................................................... 290  
10.2.2 Develop Your API ................................................... 291  
10.2.3 Build Your Application .......................................... 295  
10.3 Generating Integration Flows from APIs in SAP Cloud Platform Integration ................................................. 305  
10.4 Summary .................................................................... 309
11 API Analytics

11.1 Analyzing API Consumption ......................................................... 312
  11.1.1 Performance Analysis ...................................................... 312
  11.1.2 Error Analysis .............................................................. 313
11.2 Developing Custom Charts ....................................................... 315
11.3 Statistics Collector Policy .......................................................... 317
11.4 Analytics on Cloud Foundry ....................................................... 320
11.5 Summary ................................................................................ 323

PART IV Next Steps

12 API Program

12.1 Why an API Program? ............................................................... 327
  12.1.1 Govern the API Sprawl ................................................... 328
  12.1.2 Build the Right API through Business Alignment ............... 329
  12.1.3 Developer Ecosystem Engagement .................................... 329
12.2 Best Practices for Setting Up an API Program ......................... 330
  12.2.1 Develop an API Program ................................................ 330
  12.2.2 Use an API Platform ...................................................... 332
12.3 API Products and the API Product Manager ......................... 333
  12.3.1 API Products ................................................................. 334
  12.3.2 Role of the API Product Manager ..................................... 335
12.4 SAP’s API Program: A Case Study ........................................ 336
  12.4.1 API Portfolio Management ............................................. 337
  12.4.2 The API Makers Community ......................................... 337
  12.4.3 SAP API Business Hub .................................................. 338
  12.4.4 Best Practices and Accelerators ..................................... 338
12.5 SAP API Business Hub ............................................................. 339
  12.5.1 API Catalog ................................................................. 340
  12.5.2 API Sandbox ............................................................... 341

13 Recent and Future Developments ............................................. 345

13.1 Recent Innovations ................................................................. 345
  13.1.1 Customize the Developer Portal for Developer Engagement ... 346
  13.1.2 API Health and Advanced Analytics ................................. 347
  13.1.3 Manage Third-Party APIs with Open Connectors .............. 348
  13.1.4 Custom Attributes for Dynamic Policy Enforcement ........ 349
  13.1.5 Native Support for JSON Web Tokens .............................. 350
13.2 Future Roadmap ...................................................................... 351
  13.2.1 Multicloud API Management ........................................... 351
  13.2.2 Support for Hybrid Deployment ...................................... 352
  13.2.3 Continuous Improvement in API Lifecycle Management ...... 352
13.3 Summary ............................................................................... 354

The Authors ................................................................................. 355
Index .......................................................................................... 357
Index

Developer portal (Cont.)
destination ......................................................... 282
home page ........................................ 96
manage page ........................................ 98
my workspace page ........................................ 99
published products ........................................ 277
self-registration ........................................ 99
subscribe to a product ..................................... 97, 261
test console page ........................................ 101
view bills ................................................ 263
Developers ................................................ 41, 95
ecosystem engagement ..................................... 329
types ....................................................... 338
Development ............................................. 123, 135
methodology ............................................. 123
Digital apps .............................................. 29
Digital businesses ....................................... 24, 27, 33
Digital enterprise ....................................... 23
Discovery ................................................ 37, 56, 76, 355
business partner resource ................................ 275
copy ....................................................... 273
HERe Maps .................................................. 271
sales order ................................................ 292
Documentation ......................................... 58, 83, 334
Dynamic policy enforcement ......................... 349

E
eCommerce ................................................ 32
Edge caching ............................................. 36
Electronic data interchange (EDI) .................... 30
Email configuration ..................................... 98
Engage phase ............................................ 41, 65, 66
Enterprise API layer ...................................... 33
Environments ............................................. 342
Error analysis ........................................... 100, 263, 313
information ............................................... 314
Errors ...................................................... 116
Experience data ......................................... 26
Experience economy .................................... 25
Exporting .................................................. 27
Extensible Stylesheet Language Transformations (XSLT) ........................................ 162
Extension ................................................ 164
Extract variables ........................................ 162, 195
configure ............................................... 196
headers ................................................... 196
JSON payload .......................................... 198
query parameters ....................................... 197
URI .......................................................... 197
XML payload ........................................... 199

F
FaultRule flow ............................................. 172
Flow variables .......................................... 194
Flows ...................................................... 86, 110, 153, 154
Food services ........................................... 49

G
Governance ............................................... 36
API programs ........................................... 328

H
Health ...................................................... 38, 321
Health care ............................................... 50
HERe Maps .................................................. 271
Heterogeneous IT landscapes ......................... 28
High-tech ................................................ 51
Hybrid deployment ..................................... 352
Hybrid IT landscapes ................................... 28

I
Implementation .......................................... 124
index html file ......................................... 283
index js file ............................................ 143
Innovations .............................................. 345
Integration accelerators ................................. 339, 342
Integration flows ....................................... 305
details .................................................... 307
generate .................................................. 305, 306
view ....................................................... 309
Intelligent enterprise ................................... 25
Internal application developers ..................... 338
Invalid data cache ...................................... 160

J
JAR files ................................................... 236
create ..................................................... 237
Java ....................................................... 148
JavaScript ................................................. 165
JavaScript Object Notation (JSON) array ........... 204
convert to XML ......................................... 200
files ....................................................... 72
read ....................................................... 198
threat protection ......................................... 163, 246, 247
vulnerabilities ........................................... 246
JSON to XML ........................................... 162, 200, 201, 203, 204
options ................................................... 202
JSON Web Tokens (JWTs) ............................ 350

K
Key store .................................................. 239
Key value map operations .............................. 208, 209, 213
update and delete ....................................... 211
Key value maps ......................................... 88, 90, 162, 208
backend credentials .................................... 210
configure ............................................... 208
create ..................................................... 208
Key verification ......................................... 109, 110

L
Lifecycle management .................................... 39, 63, 103
improvements ......................................... 352
persons ................................................... 41
phases .................................................... 40, 64, 123
Live rendering .......................................... 72
Lookup cache .......................................... 160

M
Machine learning ........................................ 353
Manage phase .......................................... 40, 64, 66
Managed file transfer (MFT) ......................... 30
manifest.yaml file ...................................... 144
Match rules ............................................. 169

N
Namespace handling .................................... 203
Native applications .................................... 290
build ...................................................... 295
Neo ......................................................... 52, 144
connect ................................................... 147

O
OAuth ..................................................... 163, 222
attributes ............................................... 225
call token ............................................... 214
client ID .................................................. 227
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Cloud Platform Mobile Services</td>
<td>46, 290, 296, 298, 340</td>
</tr>
<tr>
<td>enable</td>
<td>290</td>
</tr>
<tr>
<td>SDK</td>
<td>343</td>
</tr>
<tr>
<td>SAP Cloud Platform Open Connectors</td>
<td>43</td>
</tr>
<tr>
<td>SAP Enterprise Architecture Designer</td>
<td>40</td>
</tr>
<tr>
<td>SAP Fiori apps</td>
<td>270</td>
</tr>
<tr>
<td>build</td>
<td>270, 277</td>
</tr>
<tr>
<td>design</td>
<td>283</td>
</tr>
<tr>
<td>run</td>
<td>288</td>
</tr>
<tr>
<td>SAP Gateway</td>
<td>43, 53, 58, 105, 130, 275</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>38, 43</td>
</tr>
<tr>
<td>SAP Leonardo Machine Learning</td>
<td>77</td>
</tr>
<tr>
<td>Foundation</td>
<td>77</td>
</tr>
<tr>
<td>SAP Mobile Cards</td>
<td>296, 297, 298</td>
</tr>
<tr>
<td>SAP S/4HANA</td>
<td>127, 150</td>
</tr>
<tr>
<td>SAP S/4HANA Cloud</td>
<td>291</td>
</tr>
<tr>
<td>package</td>
<td>292</td>
</tr>
<tr>
<td>SAP Subscription Billing</td>
<td>39</td>
</tr>
<tr>
<td>SAP Web IDE</td>
<td>46, 270, 277, 340, 343</td>
</tr>
<tr>
<td>enable</td>
<td>270</td>
</tr>
<tr>
<td>open</td>
<td>280</td>
</tr>
<tr>
<td>project</td>
<td>281</td>
</tr>
<tr>
<td>Scenario-driven API design methodology</td>
<td>329, 335</td>
</tr>
<tr>
<td>Schema repository</td>
<td>36</td>
</tr>
<tr>
<td>Scripts</td>
<td>154, 155</td>
</tr>
<tr>
<td>Secure Sockets Layer (SSL)</td>
<td>89</td>
</tr>
<tr>
<td>Security</td>
<td>34, 109, 219</td>
</tr>
<tr>
<td>overview</td>
<td>162</td>
</tr>
<tr>
<td>threats</td>
<td>241</td>
</tr>
<tr>
<td>Security Assertion Markup Language</td>
<td>164, 232</td>
</tr>
<tr>
<td>(SAML)</td>
<td>241</td>
</tr>
<tr>
<td>audience</td>
<td>241</td>
</tr>
<tr>
<td>generate assertion</td>
<td>236, 238</td>
</tr>
<tr>
<td>validate assertion</td>
<td>232, 234</td>
</tr>
<tr>
<td>Server stubs</td>
<td>74</td>
</tr>
<tr>
<td>Node.js</td>
<td>74</td>
</tr>
<tr>
<td>Service callout</td>
<td>166, 195, 214, 215</td>
</tr>
<tr>
<td>API local target connection</td>
<td>216</td>
</tr>
<tr>
<td>API providers</td>
<td>216</td>
</tr>
<tr>
<td>Service-level agreements (SLAs)</td>
<td>321</td>
</tr>
<tr>
<td>Services economy</td>
<td>26</td>
</tr>
<tr>
<td>Shed load</td>
<td>351</td>
</tr>
<tr>
<td>Simple Object Access Protocol (SOAP)</td>
<td>125</td>
</tr>
<tr>
<td>create request</td>
<td>197</td>
</tr>
<tr>
<td>request message</td>
<td>193</td>
</tr>
<tr>
<td>Single source of truth</td>
<td>31</td>
</tr>
<tr>
<td>Software development kits (SDKs)</td>
<td>148, 334, 353</td>
</tr>
<tr>
<td>download</td>
<td>149</td>
</tr>
<tr>
<td>generate</td>
<td>148</td>
</tr>
<tr>
<td>Source Address Mask</td>
<td>159</td>
</tr>
<tr>
<td>Spike arrest</td>
<td>35, 160, 184</td>
</tr>
<tr>
<td>configure</td>
<td>185</td>
</tr>
<tr>
<td>per minute</td>
<td>185</td>
</tr>
<tr>
<td>per second</td>
<td>186</td>
</tr>
<tr>
<td>Statistics collector</td>
<td>166, 317</td>
</tr>
<tr>
<td>code editor</td>
<td>319</td>
</tr>
<tr>
<td>create</td>
<td>318</td>
</tr>
<tr>
<td>Style guidelines</td>
<td>36</td>
</tr>
<tr>
<td>Subscription</td>
<td>261, 334</td>
</tr>
<tr>
<td>Subscription-based model</td>
<td>31</td>
</tr>
<tr>
<td>Swagger</td>
<td>65, 124</td>
</tr>
<tr>
<td>swagger yaml file</td>
<td>143</td>
</tr>
<tr>
<td>Target endpoint</td>
<td>46, 81, 171</td>
</tr>
<tr>
<td>response</td>
<td>171</td>
</tr>
<tr>
<td>segment request</td>
<td>58</td>
</tr>
<tr>
<td>segment response</td>
<td>59</td>
</tr>
<tr>
<td>spike arrest</td>
<td>160</td>
</tr>
<tr>
<td>Template customization</td>
<td>282</td>
</tr>
<tr>
<td>Test</td>
<td>93</td>
</tr>
<tr>
<td>Test console</td>
<td>93, 96, 101, 125, 264</td>
</tr>
<tr>
<td>Testing</td>
<td>124, 274</td>
</tr>
<tr>
<td>Third-party APIs</td>
<td>348</td>
</tr>
<tr>
<td>Throttling</td>
<td>35</td>
</tr>
<tr>
<td>Tiered rate plans</td>
<td>257, 259</td>
</tr>
<tr>
<td>Tools integration</td>
<td>343</td>
</tr>
<tr>
<td>Traffic management</td>
<td>35, 167</td>
</tr>
<tr>
<td>overview</td>
<td>158</td>
</tr>
<tr>
<td>Trust store</td>
<td>132, 234, 235</td>
</tr>
<tr>
<td>Twilio</td>
<td>32</td>
</tr>
<tr>
<td>Unit isolation</td>
<td>151</td>
</tr>
<tr>
<td>Usage</td>
<td>322</td>
</tr>
<tr>
<td>Utilities</td>
<td>50</td>
</tr>
<tr>
<td>XML convert to JSON</td>
<td>205</td>
</tr>
<tr>
<td>namespaces</td>
<td>202</td>
</tr>
<tr>
<td>read</td>
<td>199</td>
</tr>
<tr>
<td>threat protection</td>
<td>241, 244, 245</td>
</tr>
<tr>
<td>transform</td>
<td>162</td>
</tr>
<tr>
<td>vulnerabilities</td>
<td>241</td>
</tr>
<tr>
<td>XML schema definition (XSD)</td>
<td>241</td>
</tr>
<tr>
<td>configure</td>
<td>242</td>
</tr>
<tr>
<td>import</td>
<td>242</td>
</tr>
<tr>
<td>XML threat protection</td>
<td>164</td>
</tr>
<tr>
<td>XML to JSON</td>
<td>162, 205, 206</td>
</tr>
<tr>
<td>options</td>
<td>207</td>
</tr>
<tr>
<td>YAML files</td>
<td>72, 137</td>
</tr>
<tr>
<td>X509 certificates</td>
<td>232, 236</td>
</tr>
<tr>
<td>private keys</td>
<td>236</td>
</tr>
<tr>
<td>upload</td>
<td>233</td>
</tr>
</tbody>
</table>
Carsten Bönnen works for SAP SE within the strategic product management for SAP Cloud Platform. He received his MA in computer linguistics and artificial intelligence in Germany in 2001 and started working at SAP that same year. Initially a Java developer and trainer, he soon became a consultant and led strategic projects in the then-new field of enterprise portals. Since 2002, he has worked as a product manager for SAP NetWeaver Portal, SAP NetWeaver Visual Composer, SAP Gateway, and SAP API Management. For another four years, he has worked as director for technology strategy for the Strategic Alliance Management at Microsoft.

Harsh Jegadeesan is the vice president and head of product management for SAP’s integration and API platform. Harsh helps enterprises craft and execute their digital strategies by accelerating integration and opening up to real-time digital interactions with APIs. Harsh was instrumental in establishing SAP’s own global API program and the SAP API Business Hub—a marketplace for enterprise APIs.

Divya Mary works for SAP SE within the product management team for SAP Cloud Platform Integration. She has worked at SAP for more than 14 years. Before starting at SAP, Divya received her bachelor’s in technology in electronics and telecommunications in India.

Shilpa Vij is a product manager for SAP’s integration platform. She started her professional journey at Tata Consultancy Services Pvt. Ltd. as a C++ developer. She received her bachelor’s degree in electronics and communication engineering in 2007, and then worked towards becoming a business analyst, pursuing her passion for client interfacing and consultancy roles. Shilpa has spent almost a decade at SAP Labs India, where she has been a seasoned cloud engineering expert, program lead, product expert, and now works as a product manager.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.