

Exercise 1.1 - Manage your microservice built using SAP Cloud Application Programming

Overview

BestRun AG plans to build modern applications like chatbots, mobile applications and web applications for their employees, business users and plans to use APIs to collaborate and co-innovate with their partners. They are worried that as the usage of their chatbots and mobile applications would grow, the load on their business applications would grow. Scaling these business applications to meet the high-volume load from the chatty applications would be costly for them in the long run and, they fear that it would increase the API response time. BestRun AG plans to replicate some of their data from business applications to SAP HANA services on cloud and then expose these data as microservices running on cloud

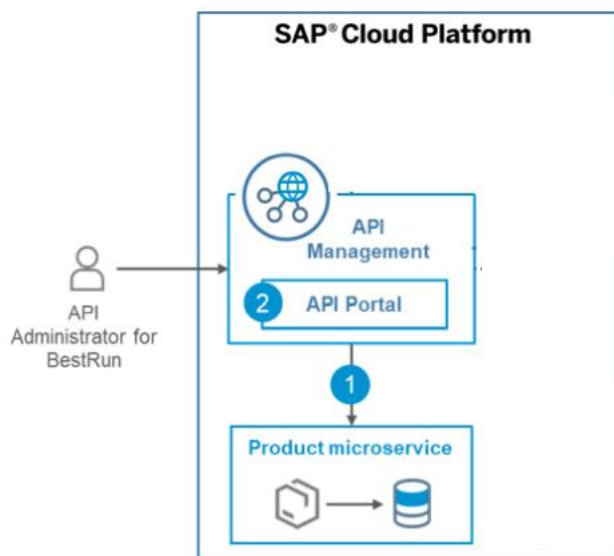
IT team of BestRun AG uses SAP Cloud Application Programming Model to quickly create business applications

<https://dmzca0qiv9fn5be4-productsrvsrv.cfapps.eu10.hana.ondemand.com/catalog>

based on their product domain model. They also plan to use SAP Cloud Platform API Management to manage this microservice and apply the BestRun IT API governance policy.

In this exercise you will configure the following for BestRun AG :-

1. Connect to non-SAPCRM system of BestRunAG using **SAP Cloud Platform Open Connectors**.
2. Create an API Proxy to microservice <https://dmzca0qiv9fn5be4-productsrvsrv.cfapps.eu10.hana.ondemand.com/catalog> built by the BestRun IT team
3. Apply the BestRun IT governance policy of Quota, Spike Arrest and Verify API Key to the product service microservice.



Step	Explanation	Screenshot
1	<p>Create API Proxy for your product Microservices</p> <p>To create APIProxy, launch APIPortal and Click on Create</p>	<p>The screenshot shows the SAP Integration Suite API Portal interface. The top navigation bar includes 'SAP Integration Suite' and 'API Portal'. The main content area is titled 'Develop' and contains the instruction 'Create APIs, Products, import Policy Templates and view Applications here.' Below this, there are tabs for 'APIs (2)', 'Products (0)', 'Applications (0)', and 'Policy Templates (2)'. The 'APIs (2)' tab is selected. In the top right corner of the main content area, there is a blue 'Create' button, which is highlighted with a red rectangle. To the right of the 'Create' button is an 'Import AP' button. Below these buttons is a table with columns 'Name', 'Title', and 'Status'.</p>

2

You can create an API Proxy by directly providing your API URL.

Click ☐ URL to provide your microservice URL details.

Create API

Select: ☒ API Provider ☐ API Proxy ☐ URL

API Provider: * ☒ Link API Provider

URL *

API Details

Name: *

Title: *

API State: *

Host Alias: *

3

In this exercise, you would be using the microservice already built by the BestRun IT team using SAP Cloud Application Programming model.

Enter following values in fields & click on **Create**

URL : <https://dmzca0qiv9fn5be4-productsrv-srv.cfapps.eu10.hana.ondemand.com/catalog>

Name : **BestRunProductsAPIs**

Title : **APIs to get products and product availability**

API Base Path : **bestrun/productservice**

Create API

Select: ☐ API Provider ☐ API Proxy ☒ URL

URL *

API Details

Name: *

Title: *

API State: *

Host Alias: *

API Base Path: *

Version:

Service Type:

Create

4

In the **Overview** tab enter " Get information about products, price and product availability " as your API Proxy Description

SAP Integration Suite API Portal

Create API

BestRunProductsAPIs

Overview Proxy EndPoint Target EndPoint Resources

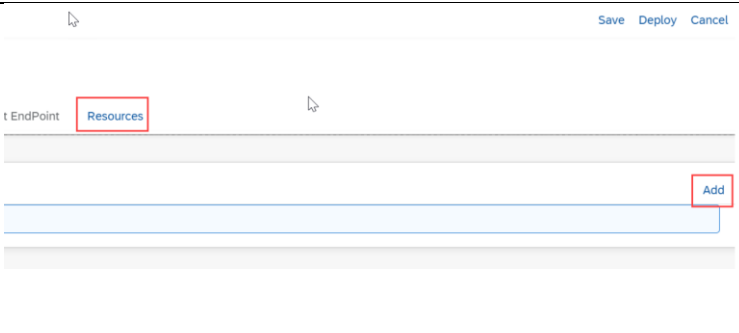


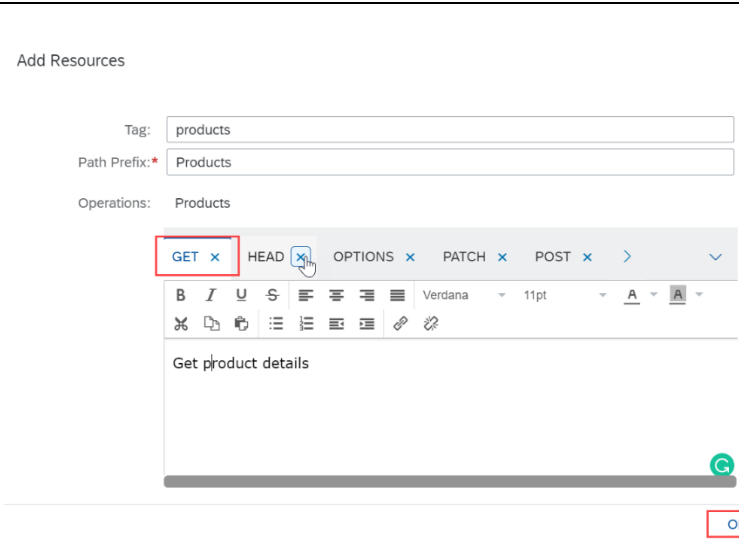

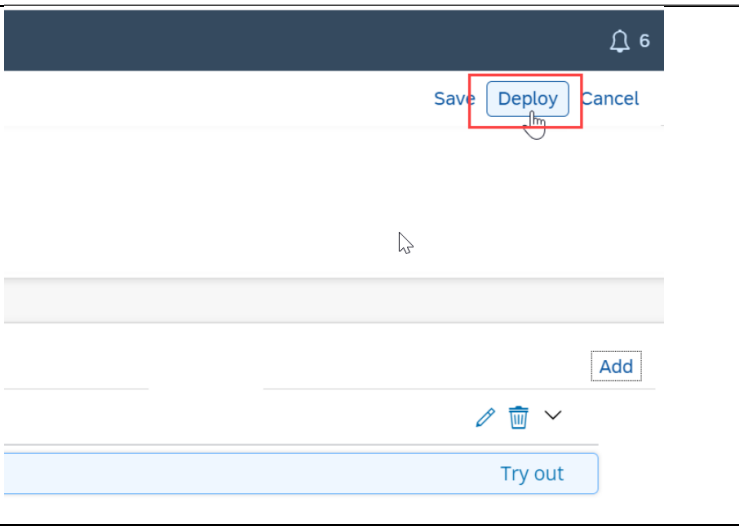
Active


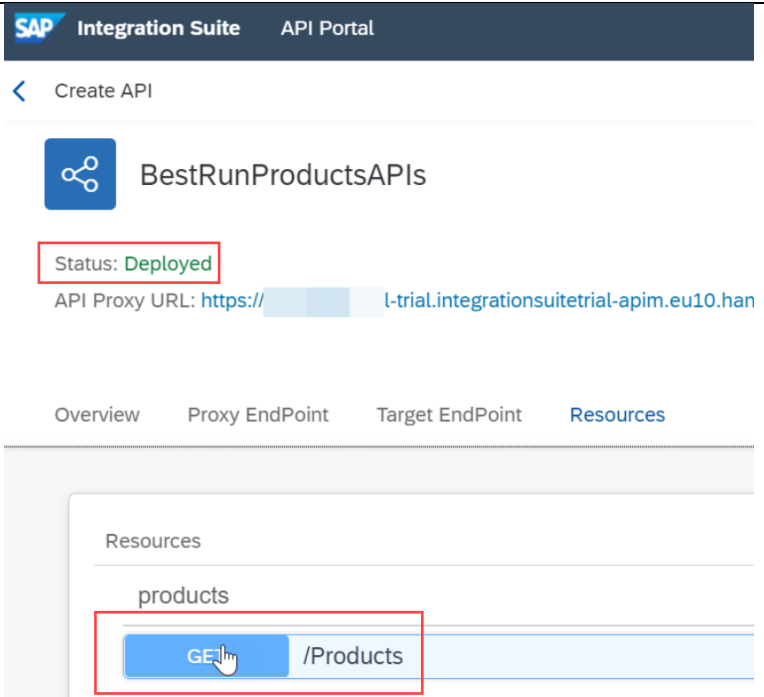

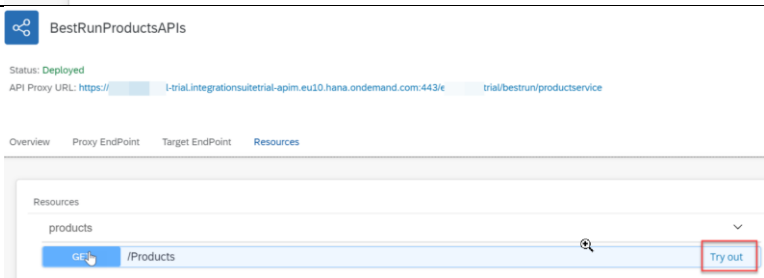

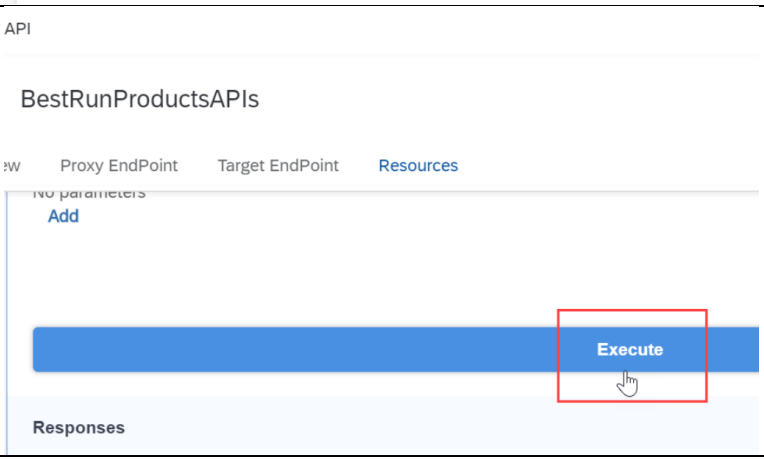
Description:

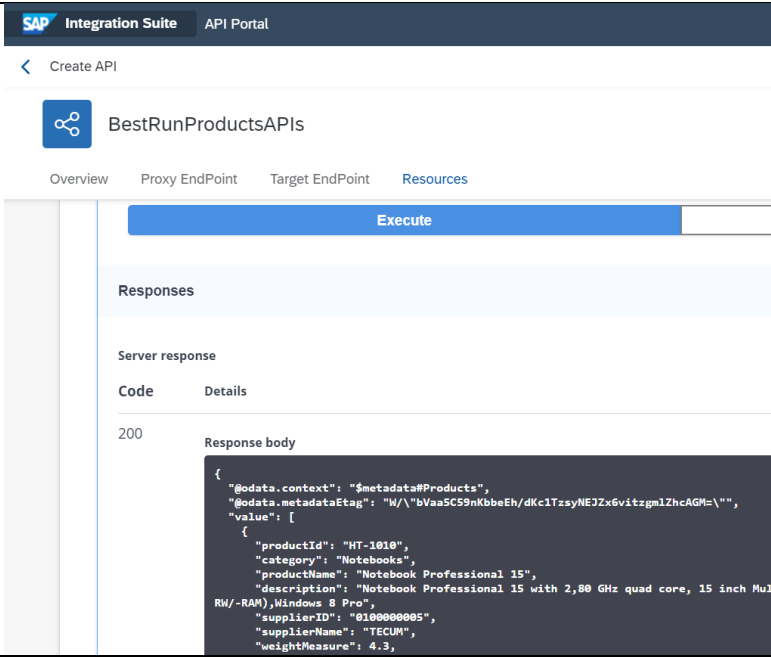
Get information about products, price and product availability

5

Click on **Resources** to navigate to Resources tab and click on **Add** to manually add resource

	<p>Based on your added resources, the path, operations field of the API Proxy definition in OpenAPI specification format will be generated. Furthermore, the corresponding conditional flows based on your added resources would be added to the the polices flow. You would be able to view the added conditional flows from Policy Designers.</p>	
6	<p>Enter Products as the Path Prefix, you can optionally set the tags to products To enable API access to only GET call, delete all operations other than GET by clicking on  next to Operation name</p> <p>Now, click on the  to add resources to API Proxy</p>	
7	<p>Click on  to activate the API Proxy to your microservice.</p>	

8	<p>APIProxy is deployed, check Status: Deployed</p> <p>Click  /Products to view the API documentation and test the select resources.</p>	 <p>The screenshot shows the SAP Integration Suite API Portal. At the top, it says 'SAP Integration Suite API Portal'. Below that is a 'Create API' button. The main heading is 'BestRunProductsAPIs'. Below this, the status is 'Status: Deployed' (highlighted with a red box). The API Proxy URL is 'https://l-trial.integrationsuitetrial-apim.eu10.han'. There are four tabs: 'Overview', 'Proxy EndPoint', 'Target EndPoint', and 'Resources'. The 'Resources' tab is selected, showing a list of resources. The 'products' resource is selected, and the 'GET /Products' button is highlighted with a red box.</p>
9	<p>Click  to test the selected resource</p>	 <p>The screenshot shows the same API Portal as before, but now the 'Try out' button is highlighted with a red box. The 'products' resource is still selected, and the 'GET /Products' button is still highlighted.</p>
10	<p>Click  to test the API Proxy</p>	 <p>The screenshot shows the same API Portal as before, but now the 'Execute' button is highlighted with a red box. The 'products' resource is still selected, and the 'GET /Products' button is still highlighted.</p>

11	The response from your microservice can viewed inline in the Server response section	 <p>The screenshot shows the SAP Integration Suite API Portal interface. At the top, there's a header with 'SAP Integration Suite' and 'API Portal'. Below that, a navigation bar includes 'Create API', 'Overview', 'Proxy EndPoint', 'Target EndPoint', and 'Resources'. The main section is titled 'BestRunProductsAPIs' and features a prominent blue 'Execute' button. Underneath, the 'Responses' section is visible, showing a 'Server response' with a status code of '200'. The 'Response body' is displayed as a JSON object containing metadata and product details.</p> <pre>{ "@odata.context": "\$metadata#Products", "@odata.metadataEtag": "W/\"bVaa5C59nKbbeEh/dKc1TzsyNE3Zx6vitzgm1ZhCAGM=\"", "value": [{ "productId": "HT-1010", "category": "Notebooks", "productName": "Notebook Professional 15", "description": "Notebook Professional 15 with 2,80 GHz quad core, 15 inch Mu RM/-RAM),Windows 8 Pro", "supplierID": "0100000005", "supplierName": "TECUM", "weightMeasure": 4.3, }] }</pre>
	Congratulations, you have successfully created API Proxy to manage your microservice. In the next exercise you will be adding the BestRun IT governance policies to your microservice	

Continue to [Exercise 1.2 - Add Corporate Governance Policy](#)