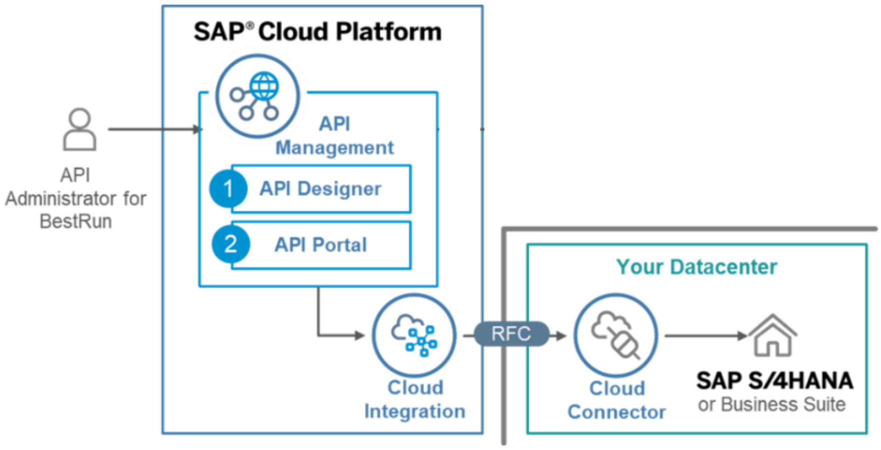
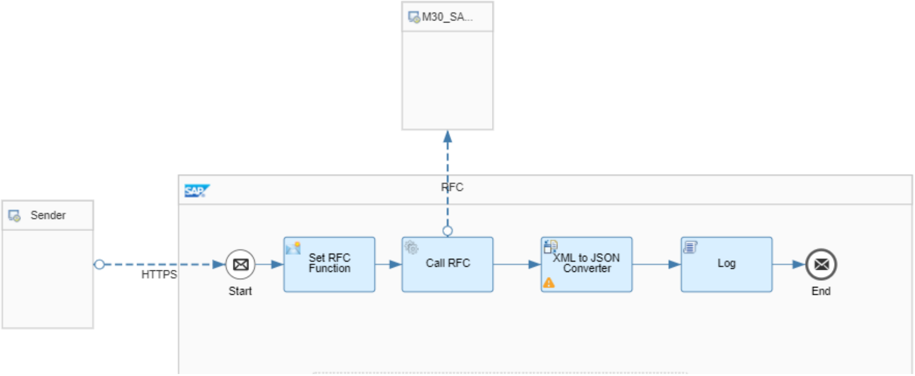
2.1 - Manage REST APIs from SAP Cloud Platform Integration

Overview

BestRun AG would like to offer to their corporate employees a mobile and chat bot application to check for flight seat availability and book their corporate trip request. BestRun AG in the long-term plan to migrate to SAP Concur to automate their employee trip requests and currently they are still using a custom built legacy on-premise travel addon which exposes RFC for integration. For building the mobile and chatbot application, they require RESTful APIs with interactive API documentation so that their application developers can build this mobile and chatbot application.

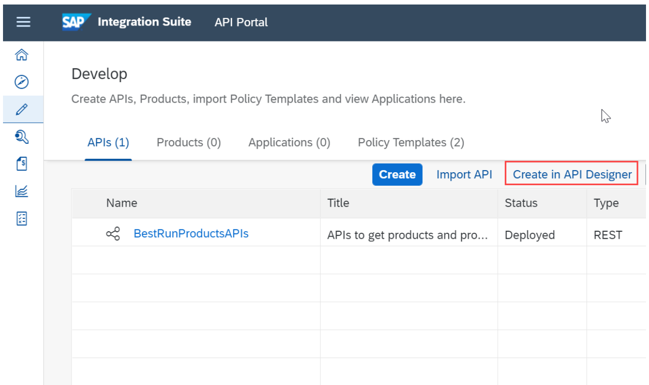
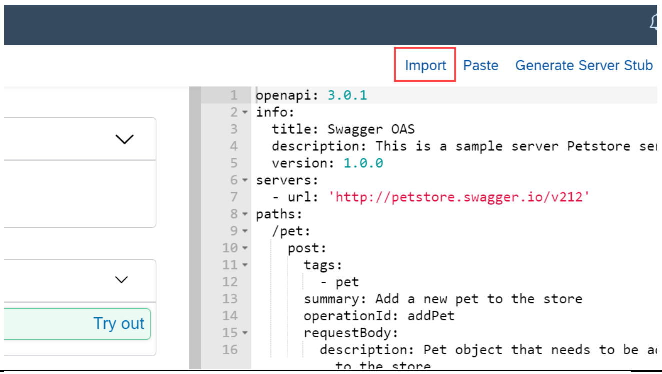
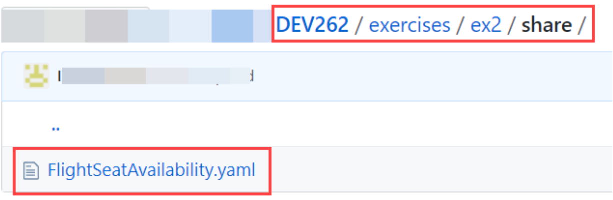


IT team of BestRun AG has used **SAP Cloud Platform Integration** to build a REST APIs that connects to their *on-premise RFC interface using RFC adapter* in SAP Cloud Platform Integration connecting securely via SAP Cloud Connectors, which is then converted to JSON response format



IT team of BestRun AG has also described this in-house built custom REST APIs using OpenAPI specification 2.0, so that they can share the API documentation with their mobile application development team for quick prototyping and faster application development

Exercise Steps:

Step	Explanation	Screenshot
1	To create an API Proxy from OpenAPI specification file. Launch APIPortal and click on Create in API Designer to create an API Proxy via API Designer.	
2	API Designer enables you describe your REST API in OpenAPI specification format. You can use the API Designer to document and manage the REST APIs created using the HTTP sender adapter of SAP Cloud Platform Integration Click to Import in the OpenAPI	
	IT team of BestRun AG has already created an OpenAPI Specification file describing the REST APIs built using SAP Cloud Platform Integration. Download this OpenAPI specification FlightSeatAvailability.yaml from **share** folder	

Step 1 screenshot:

Develop

Create APIs, Products, import Policy Templates and view Applications here.

Name	Title	Status	Type
BestRunProductsAPIs	APIs to get products and pro...	Deployed	REST

Step 2 screenshot:

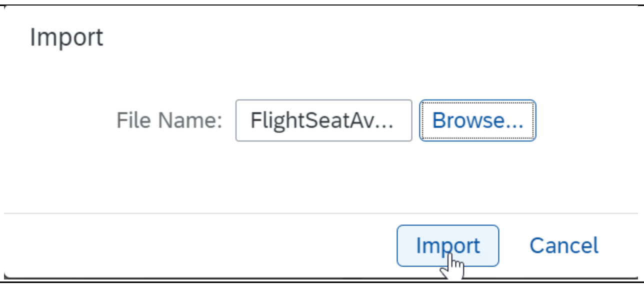
```
openapi: 3.0.1
info:
  title: Swagger OAS
  description: This is a sample server Petstore se
  version: 1.0.0
servers:
  - url: 'http://petstore.swagger.io/v212'
paths:
  /pet:
    post:
      tags:
        - pet
      summary: Add a new pet to the store
      operationId: addPet
      requestBody:
        description: Pet object that needs to be a
          to the store
```

Step 3 screenshot:

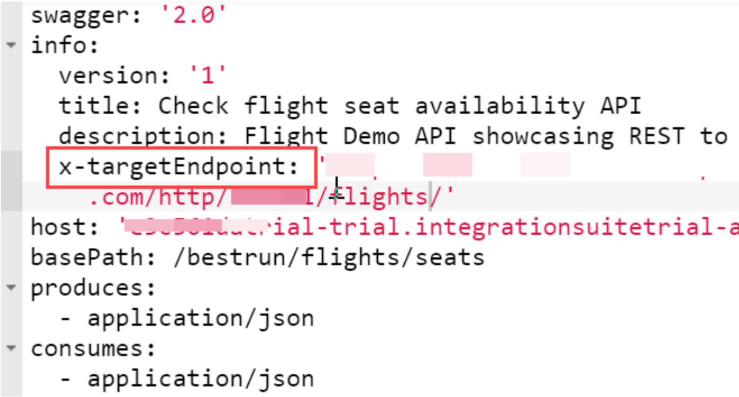
DEV262 / exercises / ex2 / share /

..

FlightSeatAvailability.yaml

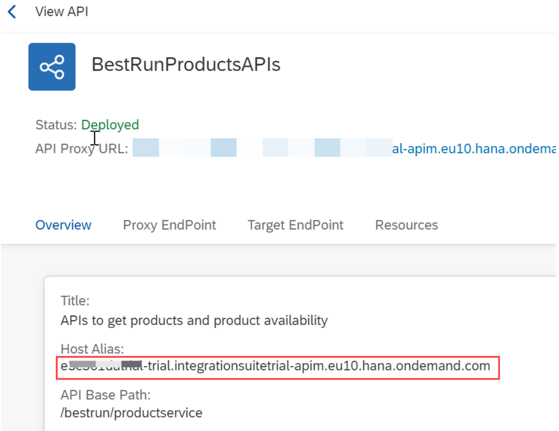
<table>
<tr>
<td>3</td>
<td>Click Browse... and select the downloaded FlightSeatAvailability.yaml file and click on
<b>Import</b></td>
<td>
Import
File Name: FlightSeatAv... Browse...
Import Cancel</td>
</tr>
<tr>
<td>4</td>
<td>OpenAPI 2.0 specification enables you to provide information about your REST APIs like title, description and version.

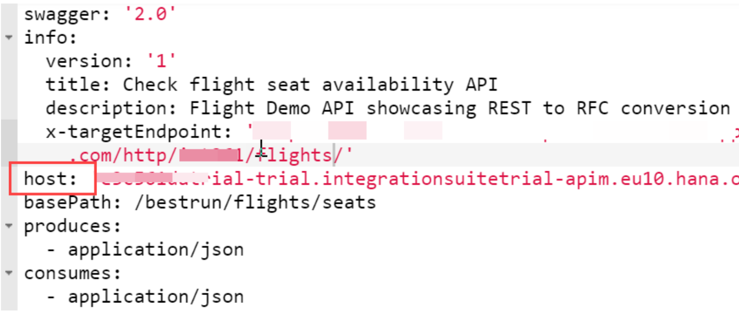
You can use the <b>x-targetEndpoint</b> custom property under info to specify your target API URL i.e., in this case the target API URL of the integration flow is <b>https://d700066-iflmap.cfapps.us10.hana.ondemand.com/http/flights</b>

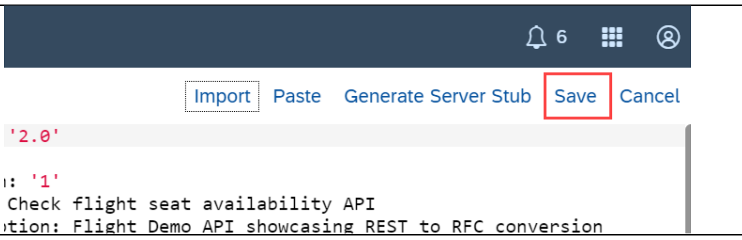
<b>basePath</b> field of your OpenAPI file would be used as the API Proxy base path i.e., in this case the basePath of your API Proxy is <b>/bestrun/flights/seats</b></td>
<td><pre>
swagger: '2.0'
info:
  version: '1'
  title: Check flight seat availability API
  description: Flight Demo API showcasing REST to
  x-targetEndpoint: '[redacted]
    .com/http/[redacted]/flights/'
host: '[redacted]ial-trial.integrationsuitetrial-a
basePath: /bestrun/flights/seats
produces:
  - application/json
consumes:
  - application/json
</pre></td>
</tr>
<tr>
<td>5</td>
<td>Now to get the value for the <b>host</b>, launch APIPortal in another browser tab and open <b>BestRunProductsAPIs</b> API Proxy(created in the previous exercise)

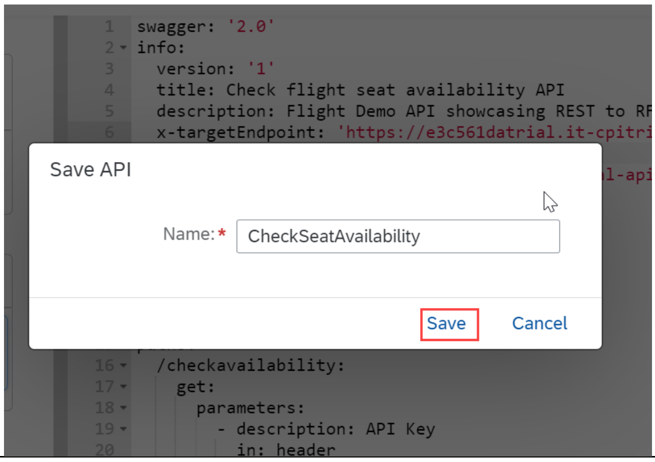
Copy <b>Host Alias</b> value of API Proxy <b>BestRunProductsAPIs</b>

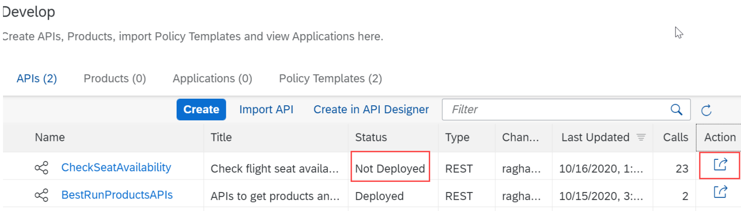
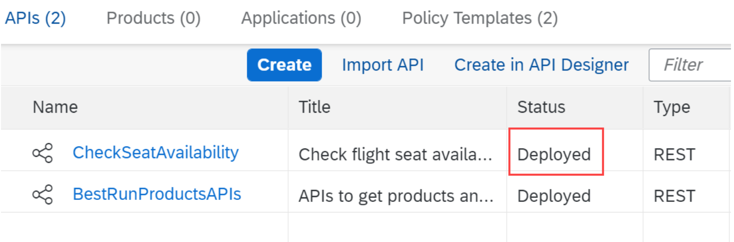
Now again switch back to the <b>API Designer</b> and provide value copied for the <b>host</b>

This host alias provides you with proxy URL protecting your target endpoint.</td>
<td>
View API
BestRunProductsAPIs
Status: Deployed
API Proxy URL: [redacted]al-apim.eu10.hana.ondema
Overview Proxy EndPoint Target EndPoint Resources
Title:
APIs to get products and product availability
Host Alias:
[redacted]-trial.integrationsuitetrial-apim.eu10.hana.ondemand.com
API Base Path:
/bestrun/productservice

<pre>
swagger: '2.0'
info:
  version: '1'
  title: Check flight seat availability API
  description: Flight Demo API showcasing REST to RFC conversion
  x-targetEndpoint: '[redacted]
    .com/http/[redacted]/flights/'
host: '[redacted]ial-trial.integrationsuitetrial-apim.eu10.hana.c
basePath: /bestrun/flights/seats
produces:
  - application/json
consumes:
  - application/json
</pre></td>
</tr>
</table>

6	Click Save to save OpenAPI file as an API Proxy.	
		6
		Import Paste Generate Server Stub Save Cancel
		`'2.0'`
		`ı: '1'`
		`Check flight seat availability API`
		`ɪtion: Flight Demo API showcasing REST to RFC conversion`

7	Enter CheckSeatAvailability in the API Name text field and click Save	
		`swagger: '2.0'`
		`info:`
		`  version: '1'`
		`  title: Check flight seat availability API`
		`  description: Flight Demo API showcasing REST to RF`
		`  x-targetEndpoint: 'https://e3c561datrial.it-cpitri`
		Save API
		Name:* CheckSeatAvailability
		Save Cancel
		`  /checkavailability:`
		`    get:`
		`      parameters:`
		`        - description: API Key`
		`          in: header`

8	API Proxy from the API Designer would be in **Not Deployed** state, i.e., the API Proxy endpoint is not active. To Deploy the API Proxy, Click ↗ and select **Deploy**	 Develop Create APIs, Products, import Policy Templates and view Applications here. APIs (2) Products (0) Applications (0) Policy Templates (2) Create Import API Create in API Designer Filter Name | Title | Status | Type | Chan... | Last Updated | Calls | Action CheckSeatAvailability | Check flight seat availa... | Not Deployed | REST | ragha... | 10/16/2020, 1:... | 23 BestRunProductsAPIs | APIs to get products an... | Deployed | REST | ragha... | 10/15/2020, 3:... | 2
9	The status of the API Proxy would change to Deployed.	 APIs (2) Products (0) Applications (0) Policy Templates (2) Create Import API Create in API Designer Filter Name | Title | Status | Type CheckSeatAvailability | Check flight seat availa... | Deployed | REST BestRunProductsAPIs | APIs to get products an... | Deployed | REST
10	Congratulations, you have successfully created an API Proxy from the OpenAPI specification file. The OpenAPI Specification file describes the RFC based interface from SAP S/4HANA which is converted to RESTful API using SAP Cloud Platform Integration	

Continue to **Exercise 2.2 - Add Policies to Securely Connect to Cloud Platform Integration API**