Hands-On Lab: Creating SAGTours REST API from Scratch

Introducing SAGTours

SAGTours is a leading provider of yachting holidays offering worldwide yachting trips. It offers pre-arranged tours that can be booked by individuals (Sailing Cruises) as well as yacht chartering (Charter Cruises). SAGTours primarily operates through its retail agency business.

SAGTours Business Goals

SAGTours has recognized the opportunity that APIs allow to generate additional revenue.

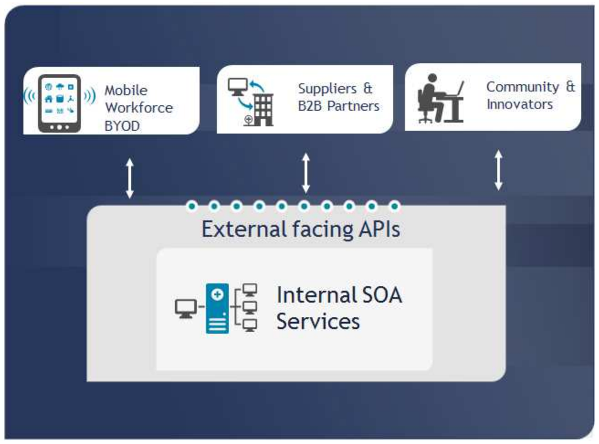
- Support Mobile initiatives and use APIs to serve data to mobile devices
- Leverage APIs for supply-chain and e-commerce initiatives to simplify B2B integration and expand channel outreach
- Encourage external innovation through the community and create incremental revenue

Therefore, SAGTours has decided to start an open API initiative along with its traditional sales channels. SAGTours expects that these parties will create innovative new travel applications and mobile applications with the content coming from SAGTours provided through its APIs. Through APIs they want to allow third parties to access booking capabilities that SAGTours uses in its current channels.

To support their API initiative, SAGTours plan to use Software AG's **webMethods API Management** suite providing their APIs to consumers.

For runtime governance and policy enforcement, SAGTours will use **webMethods API Gateway** for aspects like logging, security, transformation, consumer-based monitoring, etc.

For API developers, SAGTours will use **webMethods Developer Portal** exposing their APIs to API developers. All APIs will be published to the Developer Portal, so that internal and external consumers can search for APIs using the Developer Portal.



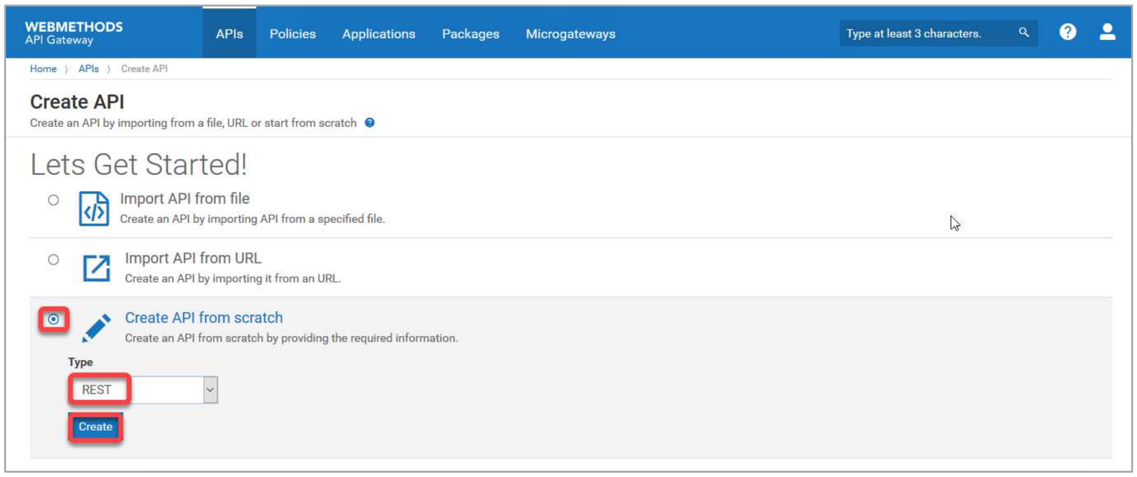
Lab Objectives

In this hands-on lab you will create the SAGTours SearchCruise REST API from scratch.

Steps

1. Open the **Windows Services** panel and double-check that the following service, needed for API Gateway, is up and running. If a service is not running, start the service.
    - **Software AG Integration Server 10.11 (default)**
2. Open a browser tab and login to **API Gateway** as user **Sumala | manage**.
3. Create an API of type REST from scratch. The service enables users to search for tours offered by **SAGTours**. The search service provides the following search options:
    - Show all available cruises
    - Show cruises based on filter criteria
    - Show a specific cruise based on its cruise ID.

    If an error occurs, the API returns an HTTP status code and, when appropriate, a plain text message in the message body providing the reason for the error.
    To create the API, navigate to **Manage APIs,** click **+ Create API.** Select **Create API from scratch**, choose type **REST** and click **Create.**



Create API
Create an API by importing from a file, URL or start from scratch

Lets Get Started!

Import API from file
Create an API by importing API from a specified file.

Import API from URL
Create an API by importing it from an URL.

Create API from scratch
Create an API from scratch by providing the required information.

Type: REST

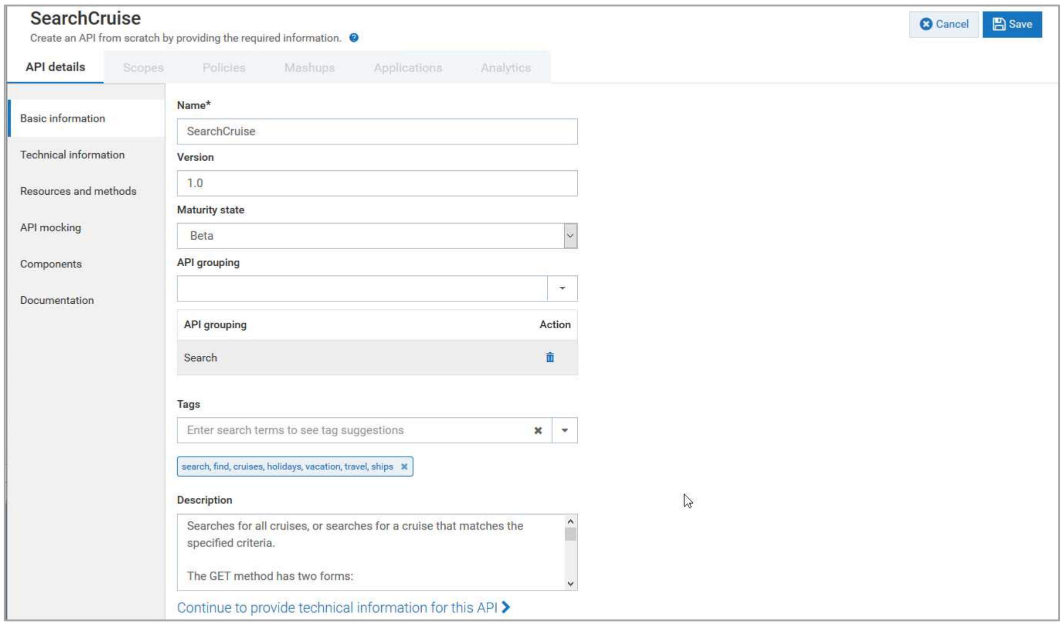
4. Add properties of the new REST service.

Note: You can copy and paste content from file **CruisesAPIResourceCruises.txt** in folder:

C:\Training\E456B-7BE\Lab2

a) Basic Information:

i.	Name:	**SearchCruise**
ii.	Version:	**1.0**
iii.	Maturity state:	**Beta** (select from drop-down)
iv.	API grouping:	**Search** (select from drop-down)
v.	Tags:	**search, find, cruises, holidays, vacation, travel, ships** (*type them in one by one, for each press the + icon or hit <enter>*)
vi.	Description**:**	**Searches for all cruises, or searches for a cruise that matches the specified criteria. The GET method…** (*copy entire description from .txt file*)


SearchCruise
Create an API from scratch by providing the required information.

Name*: SearchCruise
Version: 1.0
Maturity state: Beta
API grouping

API grouping	Action
Search	

Tags: Enter search terms to see tag suggestions
search, find, cruises, holidays, vacation, travel, ships

Description:
Searches for all cruises, or searches for a cruise that matches the specified criteria.

The GET method has two forms:

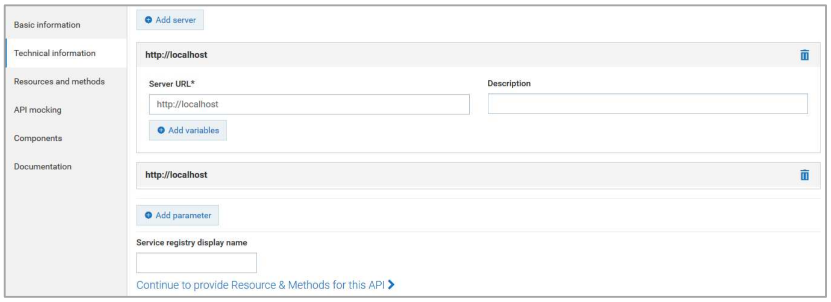
Continue to provide technical information for this API >

Click **Continue to provide technical information for this API >**.

b) Technical information:

i. Server URL: **http://localhost**

Click **Add**.



Server URL*: http://localhost
Description:
Service registry display name:

Click **Continue to provide Resource & Methods for this API >**.

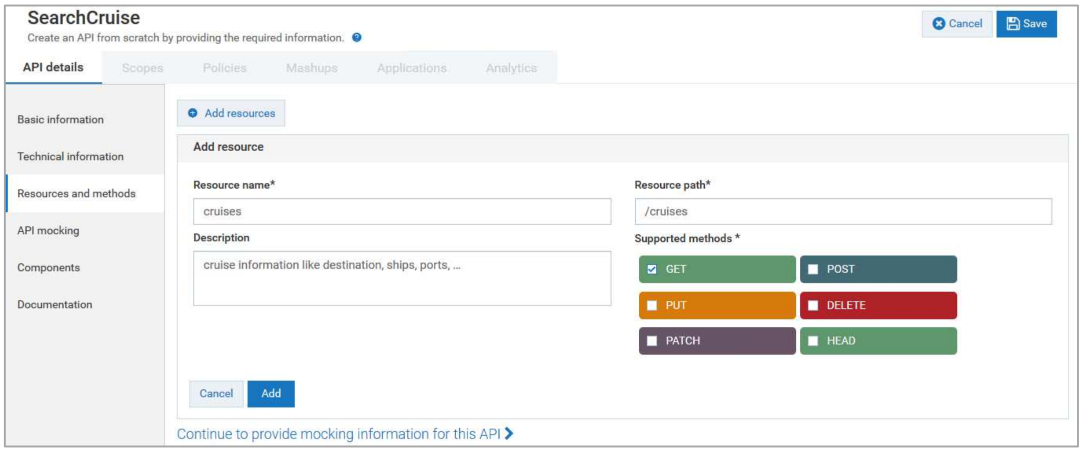
c) Resources and methods:
Click **Add resources**.

i. Resource name: **cruises**

ii. Resource path: **/cruises**

iii. Description: **Cruise information like destination, ships, ports, …**

iv. Supported methods: **GET**



SearchCruise
Create an API from scratch by providing the required information.

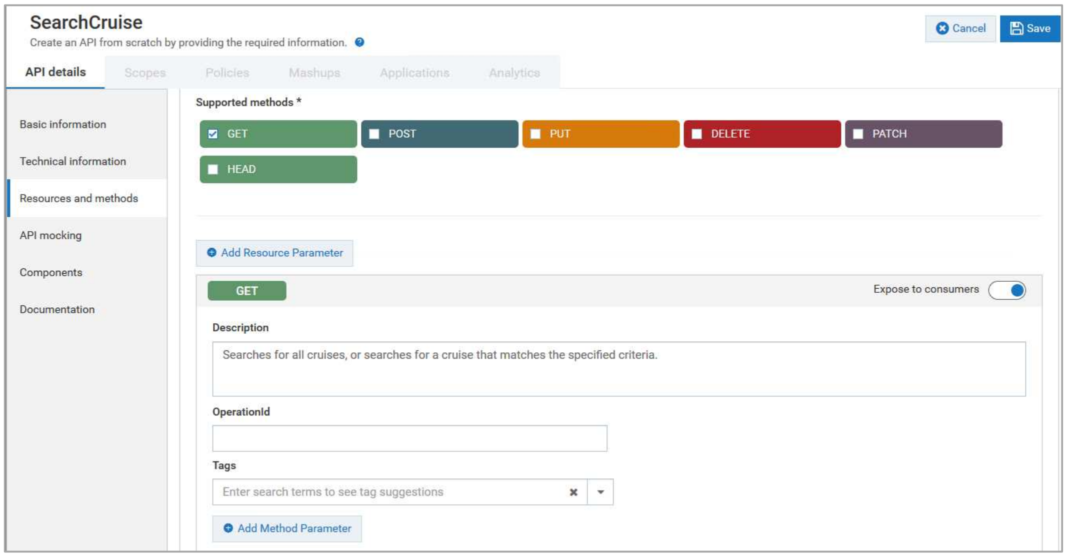
Add resource

Resource name*: cruises
Resource path*: /cruises
Description: cruise information like destination, ships, ports, …
Supported methods *: [✓ GET] POST PUT DELETE PATCH HEAD

Click **Add**.

d) Scroll down to the GET method and provide the following properties:

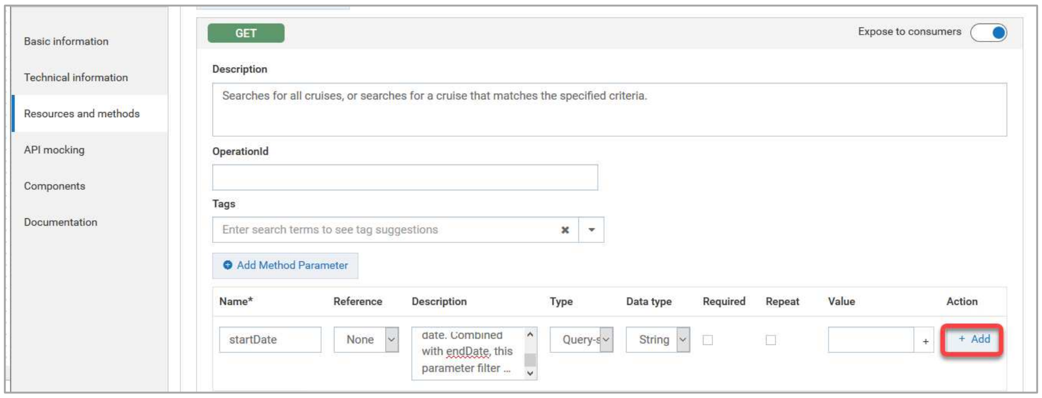
i.	Description:	**Search for all cruises, or searches for a cruise that matches the specified criteria.**



Description
Searches for all cruises, or searches for a cruise that matches the specified criteria.

e) **Add** a **Method Parameter** to the GET method. Specify the following properties:

i.	Name:	**startDate**
ii.	Reference:	**None**
iii.	Description:	**Cruise start date, in the format YYYYMMDD. Selects cruises that start on this date. Combined with endDate, this ...** (*copy entire description text from .txt file*)
iv.	Type:	**Query-string**
v.	Data Type:	**String**
vi.	Required:	<***uncheck***>
vii.	Repeat:	<***leave unchecked***>
viii.	Value:	<***leave empty***>



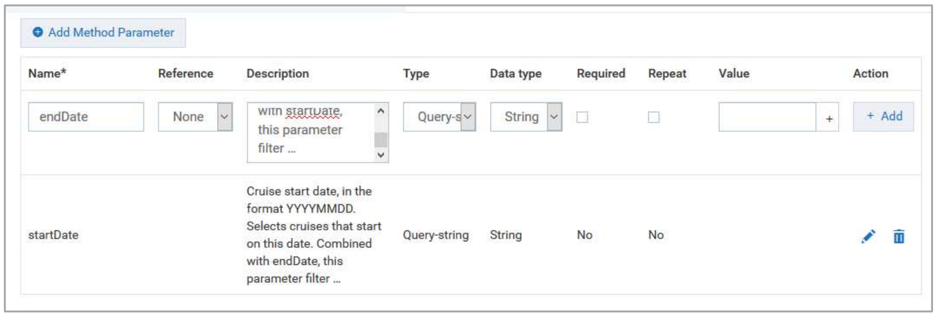
Description
Searches for all cruises, or searches for a cruise that matches the specified criteria.

Name*	Reference	Description	Type	Data type	Required	Repeat	Value	Action
startDate	None	date. Combined with endDate, this parameter filter ...	Query-s	String	☐	☐		+ Add

Click **+ Add**.

f) Click **Add Method Parameter** again and provide the following properties:

i.	Name:	**endDate**
ii.	Reference:	**None**
iii.	Description:	**Cruise end date, in the format YYYYMMDD. Selects cruises that end on this date. Combined with startDate, this ....** (*copy entire description text from .txt file*)
iv.	Type:	**Query-String**
v.	Data Type:	**String**
vi.	Required:	<***leave unchecked***>
vii.	Repeat:	<***leave unchecked***>
viii.	Values:	<***leave empty***>

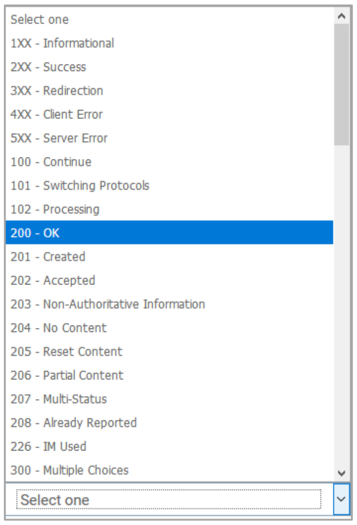


Name*	Reference	Description	Type	Data type	Required	Repeat	Value	Action
endDate	None	with startDate, this parameter filter ...	Query-s	String	☐	☐		+ Add
startDate		Cruise start date, in the format YYYYMMDD. Selects cruises that start on this date. Combined with endDate, this parameter filter ...	Query-string	String	No	No		

Click **+ Add**.

g) Click **Add Response** to add an HTTP Status code.

i. Status code: **200 – OK**



ii. Description: **Search (with filter) was successful. GET method returned a list of all cruises that match the specified criteria, in JSON format.**

Click **Add**.

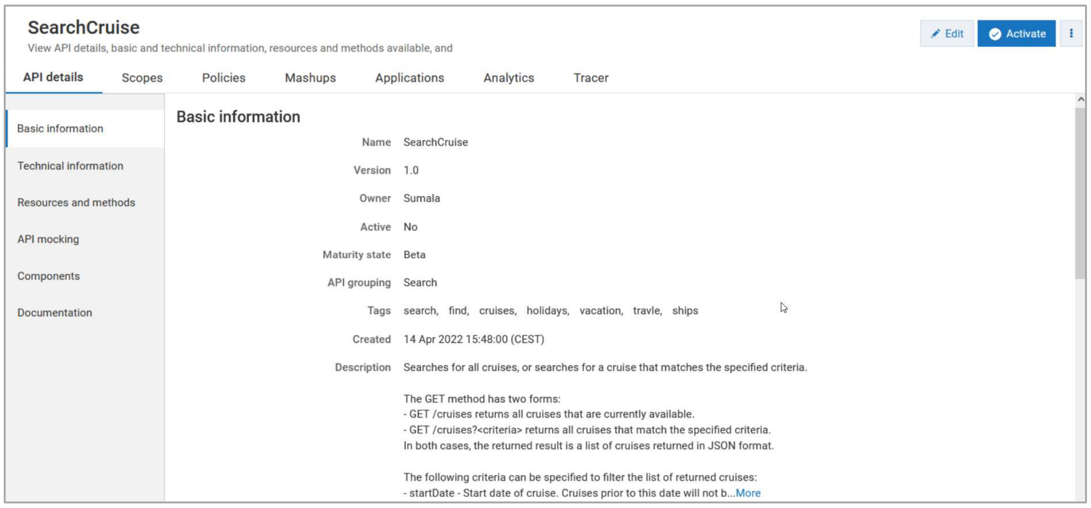
h) Add another Response Code:

   i. Status Code: **400 - Bad Request**

   ii. Description: **Query failed because the filter parameters are not valid. Check the resulting message for detail.**

   Click **Add.**

i) Click **Save.**



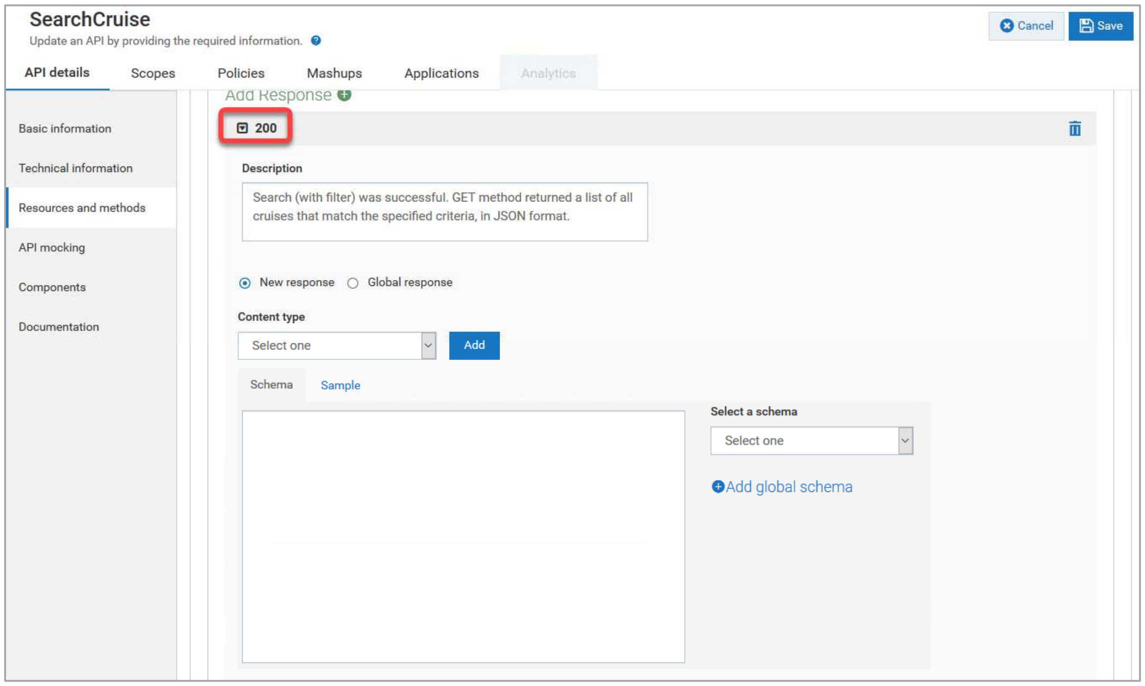
Basic information

Name	SearchCruise
Version	1.0
Owner	Sumala
Active	No
Maturity state	Beta
API grouping	Search
Tags	search, find, cruises, holidays, vacation, travle, ships
Created	14 Apr 2022 15:48:00 (CEST)
Description	Searches for all cruises, or searches for a cruise that matches the specified criteria.

The GET method has two forms:
- GET /cruises returns all cruises that are currently available.
- GET /cruises?<criteria> returns all cruises that match the specified criteria.
In both cases, the returned result is a list of cruises returned in JSON format.

The following criteria can be specified to filter the list of returned cruises:
- startDate - Start date of cruise. Cruises prior to this date will not b...More

5. Switch over to **Edit** mode and navigate to **Resources and methods**. Expand **/cruises** and select the response code **200.**



Description

Search (with filter) was successful. GET method returned a list of all cruises that match the specified criteria, in JSON format.

[✓ New response] Global response

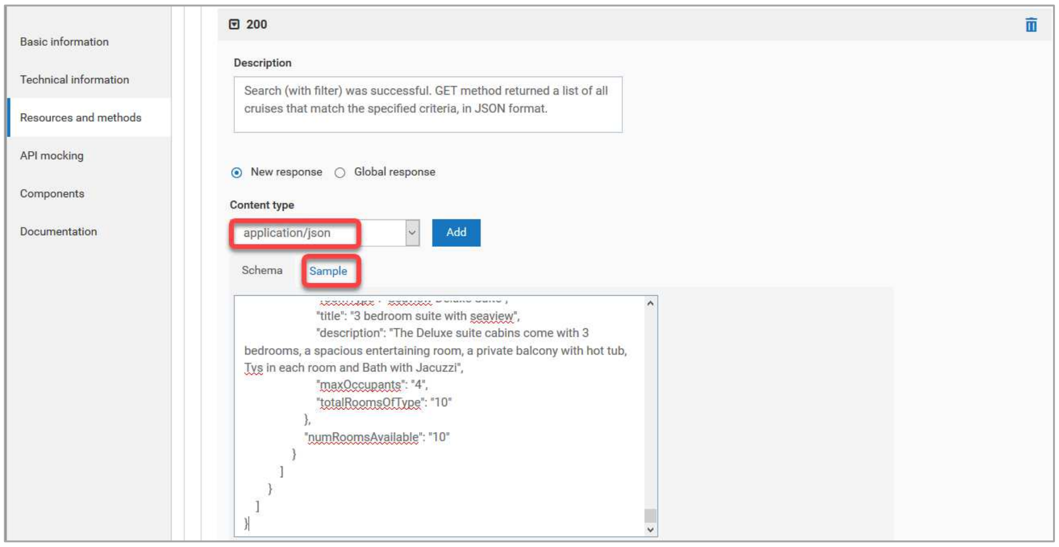
Content type

Select one

a) Switch to tab **Sample** to configure a sample response document. Provide the following data:

i. Content Type: **application/json**

ii. Sample: ***<copy and paste response 200 sample code from .txt file mentioned above>***



Search (with filter) was successful. GET method returned a list of all cruises that match the specified criteria, in JSON format.

Content type: application/json

```
                    "title": "3 bedroom suite with seaview",
                    "description": "The Deluxe suite cabins come with 3 bedrooms, a spacious entertaining room, a private balcony with hot tub, Tvs in each room and Bath with Jacuzzi",
                    "maxOccupants": "4",
                    "totalRoomsOfType": "10"
                },
                "numRoomsAvailable": "10"
            }
        ]
    }
  ]
}
```

Click **Add**.

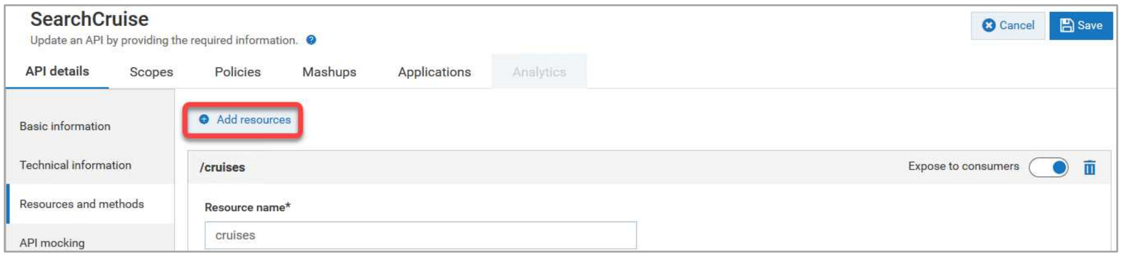
b) Click **Save**.

6. Add another resource to API SearchCruises using the cruiseID as Path parameter:

Note: You can copy and paste content out of file **CruisesAPIResourceCruiseWithID.txt** in folder: **C:\Training\E456B-7BE\Lab2**

a) Click **Edit** to change to edit mode.

b) Select **Resources and methods** from the API details menu on the left-hand side. Click **+ Add resources.**



Resource name*: cruises

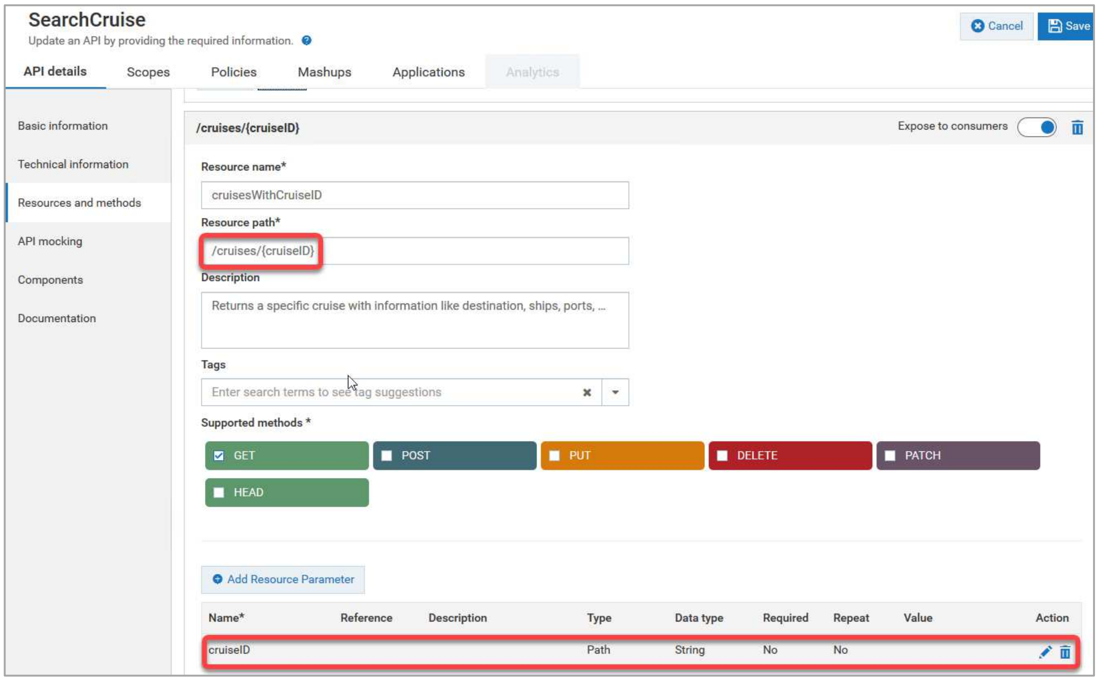
c) Provide the following details under Add resource:

i.	Resource name:	**cruisesWithCruiseID**
ii.	Resource path:	**/cruises/{cruiseID}**
iii.	Description:	**Returns a specific cruise with information like destination, ships, ports, ...**
iv.	Supported Methods:	**GET**

Click **Add**.

7. Enhance the newly created resource **cruisesWithCruiseID**:

a) Ensure the Resource path parameter has been created.


SearchCruise
Update an API by providing the required information.

API details | Scopes | Policies | Mashups | Applications | Analytics

/cruises/{cruiseID} — Expose to consumers

Resource name*: cruisesWithCruiseID
Resource path*: /cruises/{cruiseID}
Description: Returns a specific cruise with information like destination, ships, ports, ...
Tags: Enter search terms to see tag suggestions
Supported methods *: [✓ GET] POST PUT DELETE PATCH HEAD

Name*	Reference	Description	Type	Data type	Required	Repeat	Value	Action
cruiseID			Path	String	No	No		

b) Scroll down to the **GET** method and provide the following property:

i. Description: **Returns a specific cruise based on the specified cruise ID.**

c) Click **Add Response** to add an HTTP Status Code:

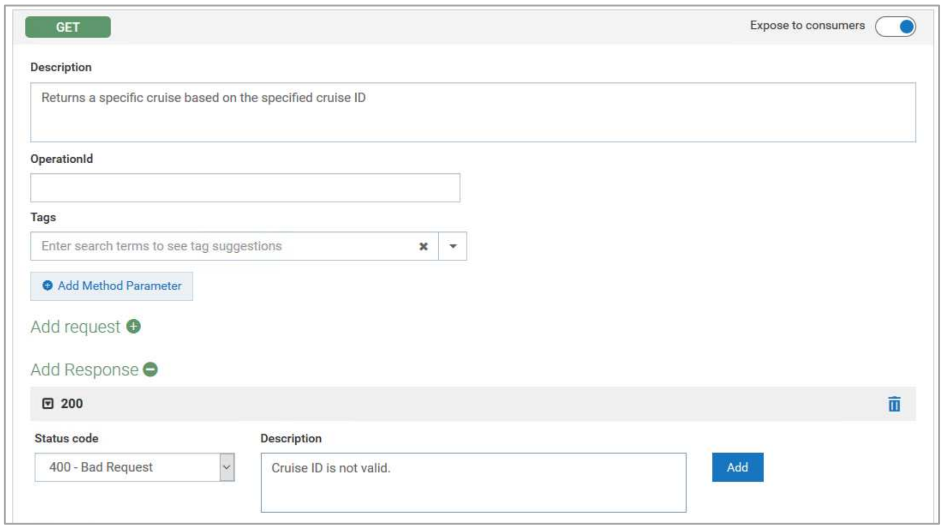
i.	Status Code:	**200 - OK**
ii.	Description:	**Search with cruiseID was successful. GET methods returns a specific cruise.**

Click **Add.**

d) Add another Response Code:

i. Status Code: **400 - Bad Request**

ii. Description: **Cruise ID is not valid.**



GET

Expose to consumers

Description
Returns a specific cruise based on the specified cruise ID

OperationId

Tags
Enter search terms to see tag suggestions

Add Method Parameter

Add request

Add Response

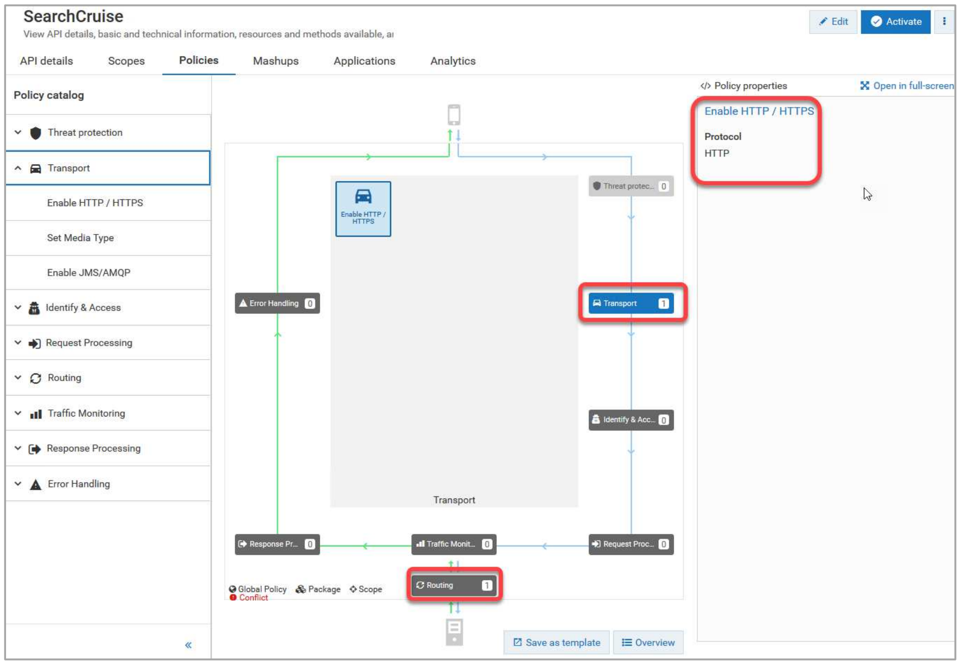
200

Status code: 400 - Bad Request
Description: Cruise ID is not valid.

Add

Click **Add.** Click **Save.**

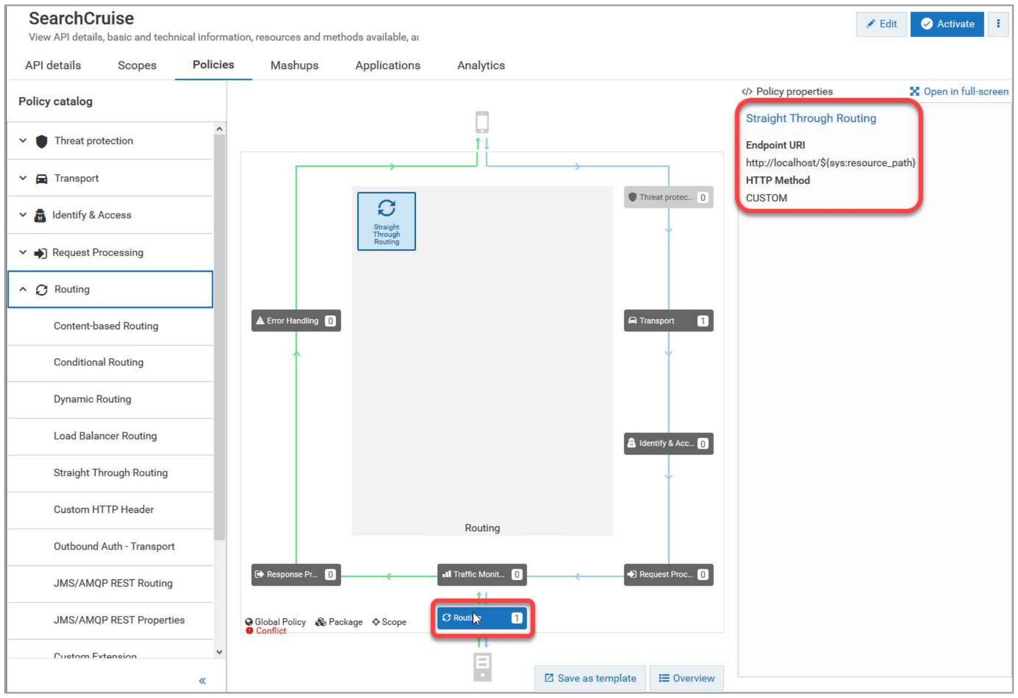
8. To review the default Policy definitions, select the **Policies** tab for API **SearchCruise**. The only policies in place are inside the sections **Transport** and **Routing**.
Section Transport is already highlighted. Policy **Enable HTTP/HTTPS** action is selected by default and configured, so that Protocol is set to **HTTP**.



SearchCruise
View API details, basic and technical information, resources and methods available, a

Enable HTTP / HTTPS
Protocol
HTTP

9. Review the Routing policy by clicking the **Routing** box in the diagram. The **Straight Through Routing** policy action is selected by default and pre-configured.

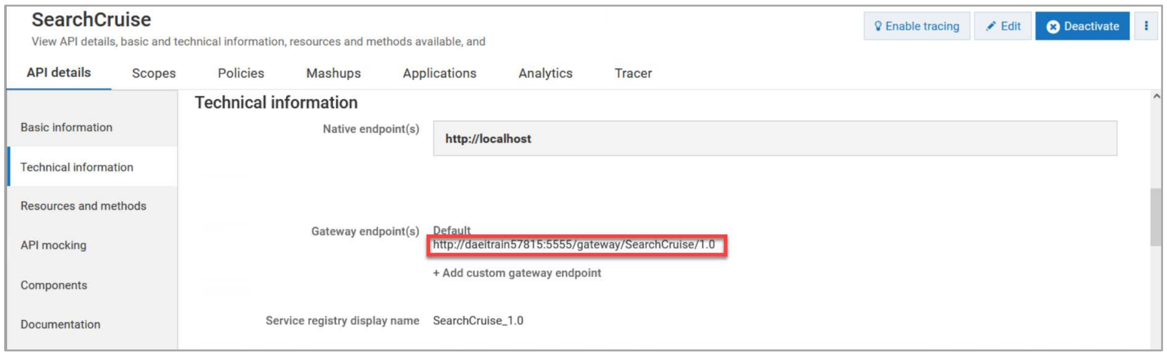


Straight Through Routing
Endpoint URI
http://localhost/${sys:resource_path}
HTTP Method
CUSTOM

10. Click **Activate** to active the SearchCruise API. Confirm with **Yes**.

Note: The service will now be virtualized and created in API Gateway.

11. Switch to tab **API details** and navigate to section **Technical information**. You will find a reference to the native REST endpoint and the URL to access the Gateway endpoint(s). Copy the **Gateway endpoint** access URL into your clipboard.



Technical information
Native endpoint(s) http://localhost
Gateway endpoint(s) Default
http://daeitrain57815:5555/gateway/SearchCruise/1.0
Service registry display name SearchCruise_1.0

Note: Of course, the server name in the URL depends on your environment.

12. Open pre-installed **Postman** as a REST client. Configure a GET test request in a collection that invokes our newly created API in API Gateway.

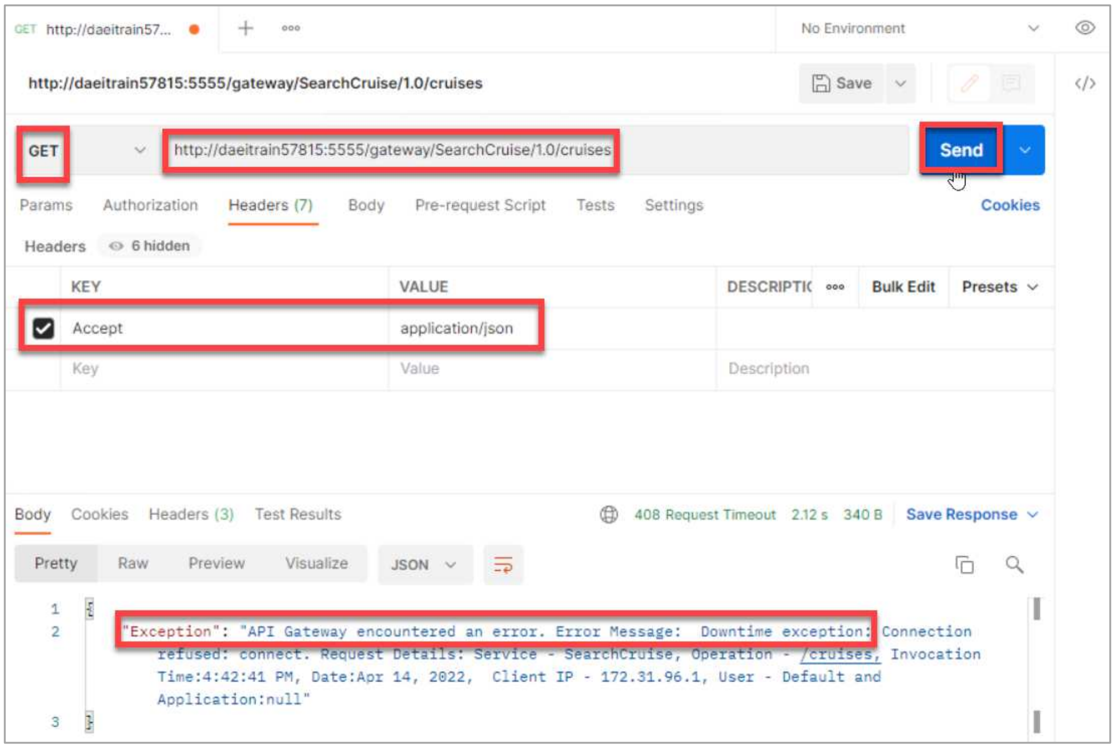
    a) Method: **GET**

    b) URL: <Paste the access URL from the clipboard and append the **/cruises** resource from the Resources and methods profile>

    c) Headers:

       i. Accept: **application/json**

    Run the request by pressing the **Send** button.



```
{
    "Exception": "API Gateway encountered an error. Error Message:  Downtime exception: Connection
        refused: connect. Request Details: Service - SearchCruise, Operation - /cruises, Invocation
        Time:4:42:41 PM, Date:Apr 14, 2022,  Client IP - 172.31.96.1, User - Default and
        Application:null"
}
```

Note: The error response for the request shows a downtime exception because we haven't provided a real endpoint for the native service. As a bypass we will use the Mocking feature of API Gateway in the next hands-on lab.

13. **Log out** from API Gateway.