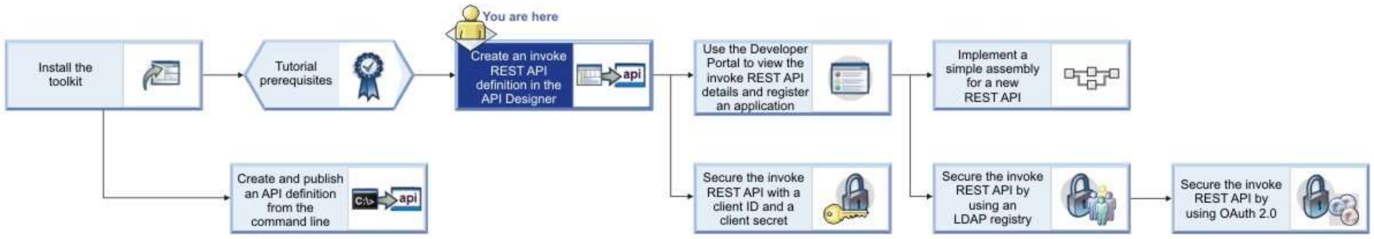
Table of contents

utorial: Creating an invoke REST API definition

is tutorial shows you how to define and implement a REST API definition that proxies an existing service.

Before you begin

The following diagram shows the sequential flow through the IBM® API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

In this tutorial you will complete the following lessons:

- Creating a REST API definition
- Testing the REST API
- Creating a Product and a Plan for the REST API
- Publishing your Product

Creating a REST API definition

Add and define a REST API to return the branch details of an example BankA.

To add and define a REST API, complete the following steps:

1. Create a folder to hold your API and Product definitions, and change to that folder in a command window.
2. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   Copy to clipboard

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   Copy to clipboard

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note

The login page prompts you to **Sign in with Bluemix**. Enter your Bluemix credentials, which authenticates you on Bluemix and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.

Table of contents

Note

If you need to run the editor on a different port, use the following command:

Linux Mac OS X

```
PORT=port_number apic edit
```

Copy to clipboard

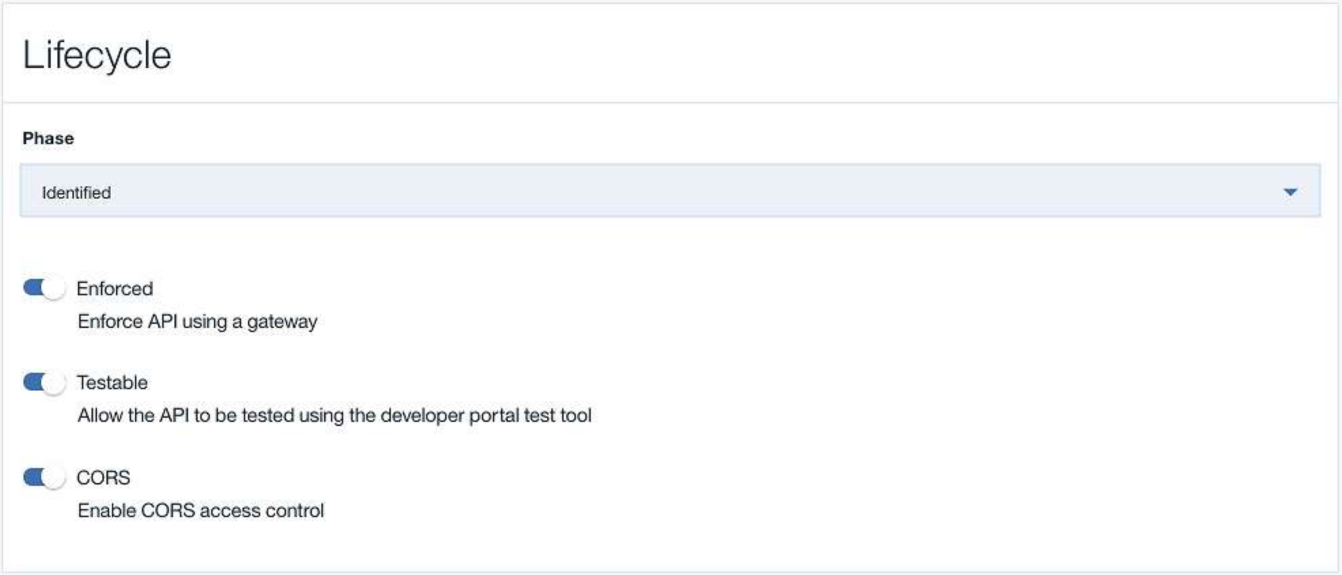
Windows

```
set PORT=port_number && apic edit
```

Copy to clipboard

where *port_number* is the port number to use.

3. Log in to APIConnect Designer with the appropriate method.
4. Click **Add > New API**.
5. Enter the appropriate information to create a REST API definition.
   a. In the **Title** field, enter `Branches`.
   b. The **Name** and **Base Path** fields autopopulate with the terms `branches` and `/branches` respectively.
   c. Leave the **Version** field at `1.0.0`.
   d. Leave the default **Additional properties** as they are.
6. You do not add a product at this time, click **Create API**.
7. If the API Editor help screen appears, click the sentence **Learn more about composing APIs**, or click **Got it!** to access the main screen immediately.
8. In the side bar, click **Lifecycle** to display the **Lifecycle** panel. Ensure that the **Enforced**, **Testable**, and **CORS** toggles are set to the **On** position as shown in the following screen capture:



Lifecycle

Phase

Identified

Enforced
Enforce API using a gateway

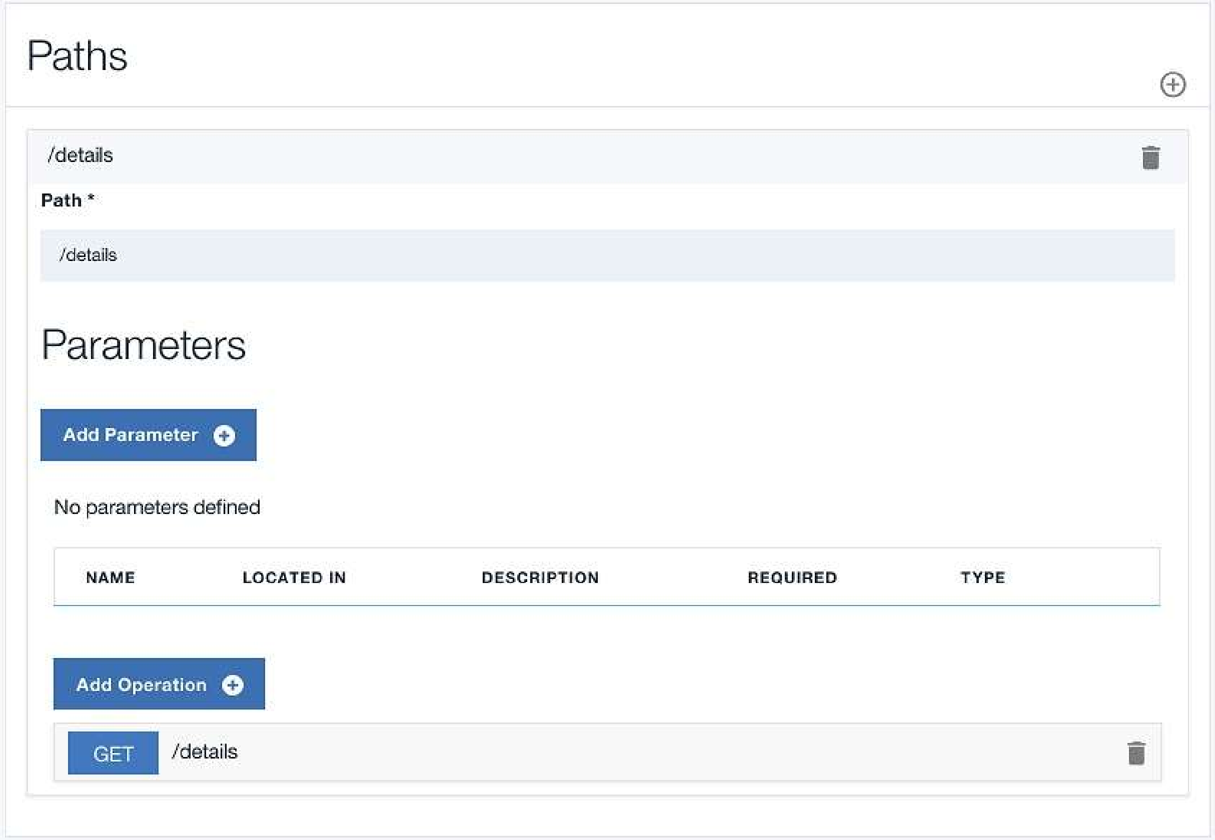
Testable
Allow the API to be tested using the developer portal test tool

CORS
Enable CORS access control

9. In the side bar, click **Security Definitions** to display the **Security Definitions** panel. Notice that `clientIdHeader` security definition already exists, and in the **Security** section you see that `Option 1` is active with `clientIdHeader (API Key)`.
10. In the side bar, select **Paths** to display the **Paths** panel. Create a new path by clicking the **Add Path** icon ⊕.

Table of contents

11. In the **Path** field, replace the default path segment with `/details`. When an operation is called, this path segment is appended to the URL of your API.



Paths

/details

Path *

/details

Parameters

No parameters defined

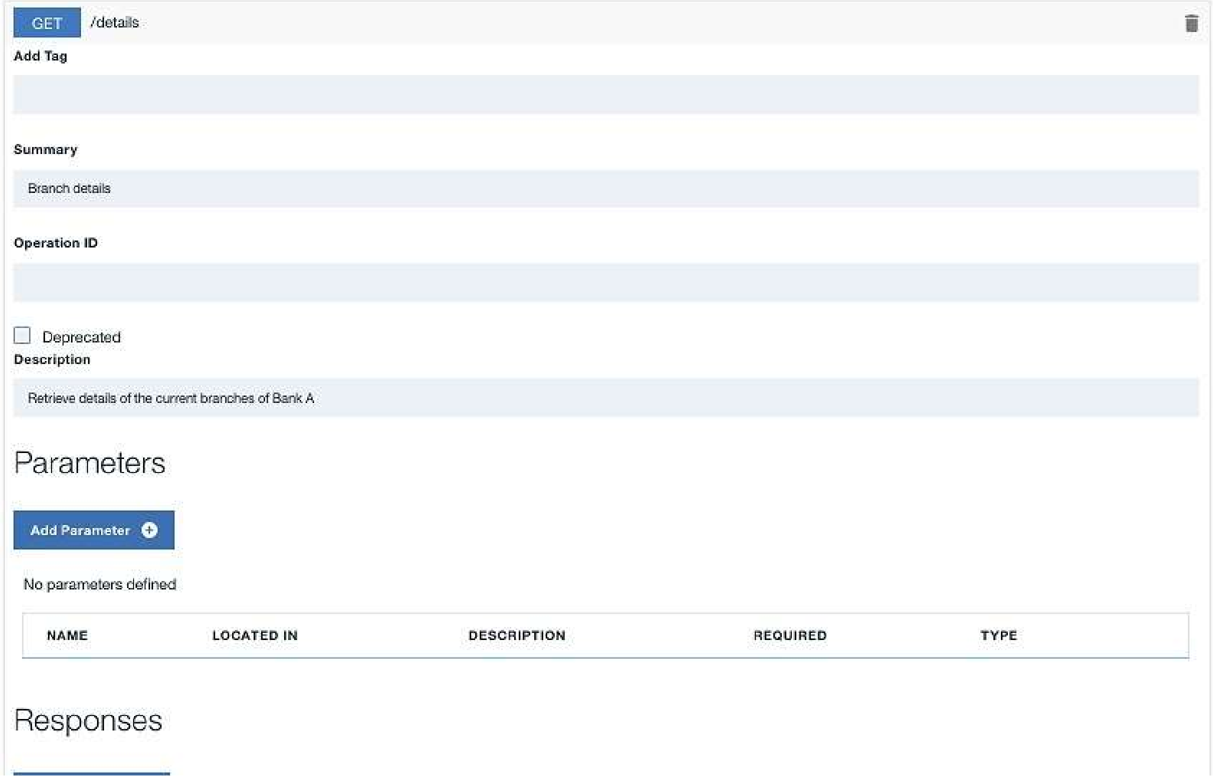
NAME	LOCATED IN	DESCRIPTION	REQUIRED	TYPE

GET /details

12. By default, a single GET operation is already in your Path. Click the **GET** button to expand the setting dialog.
13. For the operation, provide a summary and a description as in the following table.

Table 1. Operation definition values

Field	Value
Summary	`Branch details`
Description	`Retrieve details of the current branches of BankA`



GET /details

Add Tag

Summary

Branch details

Operation ID

□ Deprecated

Description

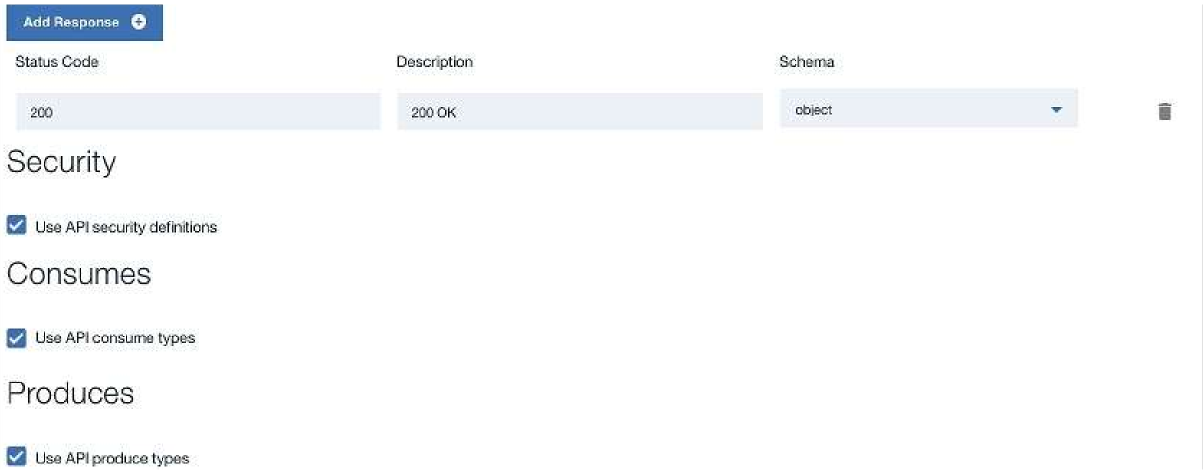
Retrieve details of the current branches of Bank A

Parameters

No parameters defined

NAME	LOCATED IN	DESCRIPTION	REQUIRED	TYPE

Responses



Add Response ⊕

Status Code	Description	Schema
200	200 OK	object

Security

☑ Use API security definitions

Consumes

☑ Use API consume types

Produces

☑ Use API produce types

14. In the side bar, click **Definitions** to display the **Definitions** panel. Add a Definition by clicking the **Add Definition** icon ⊕.
15. Expand your new definition by clicking **new-definition-1**. For the **Name** field, enter `address`, and a **Description** of `The format of the address object`.
16. Using the same **Definitions** panel, configure the **Properties** definition according to the following table. Edit the default property and then create new properties by clicking **Add Property** and editing the default values.

Table 2. Properties

Property Name	Description	Type	Example
street1	The first line of the address	string	4660 La Jolla Village Drive
street2	The second line of the address	string	Suite 300
city	The city of the address	string	San Diego
state	The state of the address	string	CA
zip_code	The zip code of the address	string	92122

This is an OpenAPI (Swagger 2.0) schema definition and is presented to developers in the Developer Portal to provide them with information about the type of data to expect in their response.

The **Required** * column indicates whether a property is required for success if a rest-validate policy uses the definition to perform validation. In this tutorial, no validation is performed and so none of your properties need to be marked as required.

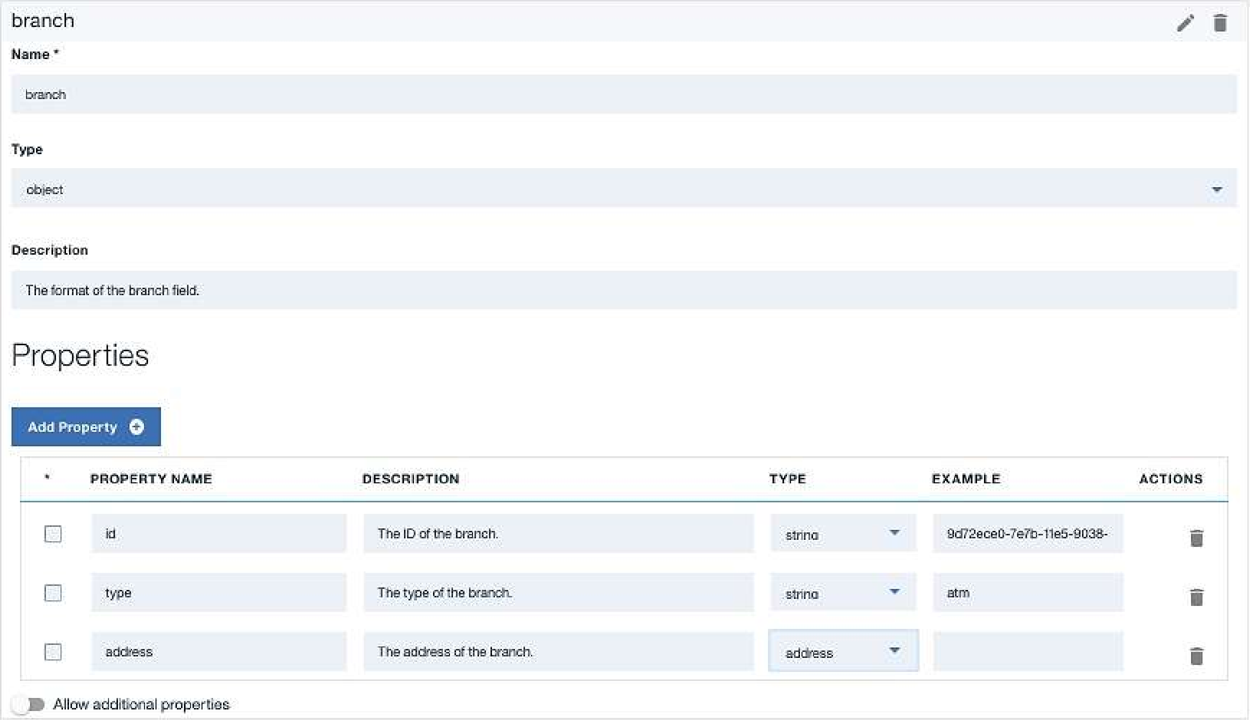
17. Create a second definition by clicking the **Add** icon ⊕ in the **Definitions** panel.
18. Name the definition `branch` and, in the **Description** field, enter `The format of the branch field`.
19. Configure the **branch** definition to have the properties listed in the following table by creating new properties and editing the default property. Create new properties by clicking **Add Property**.

Table 3. Properties

Property Name	Description	Type	Example
address	The address of the branch	address	
type	The type of branch	string	atm
id	The ID of the branch	string	9d72ece0-7e7b-11e5-9038-55f9f9c08c06

Notice that for the **address** property, the type of the property references another definition within your API and the example is left blank. In this manner, you can create complex data structures.

Table of contents



branch

Name *

branch

Type

object

Description

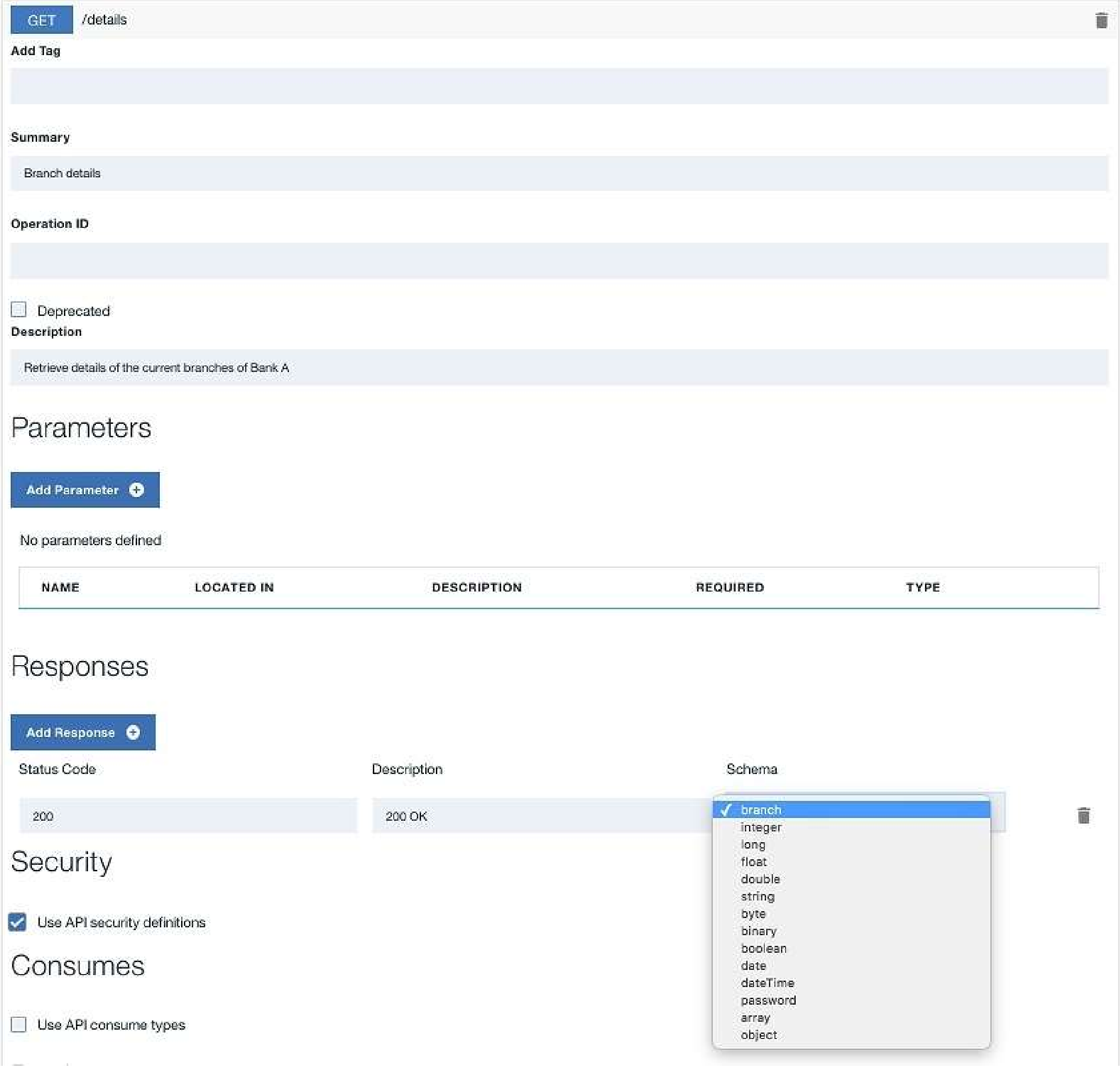
The format of the branch field.

Properties

*	PROPERTY NAME	DESCRIPTION	TYPE	EXAMPLE	ACTIONS
□	id	The ID of the branch.	string	9d72ece0-7e7b-11e5-9038-	
□	type	The type of the branch.	string	atm	
□	address	The address of the branch.	address		

Allow additional properties

20. In the side bar, select **Paths** to display the **Paths** panel. For the **/details** Path, click **GET** to expand the available settings. Include the **branch** definition in the **GET** operation **Status Code 200** response by clicking the **Schema** field and selecting **branch** from the drop-down list.



GET /details

Add Tag

Summary

Branch details

Operation ID

□ Deprecated

Description

Retrieve details of the current branches of Bank A

Parameters

No parameters defined

NAME	LOCATED IN	DESCRIPTION	REQUIRED	TYPE

Responses

Status Code	Description	Schema
200	200 OK	branch

Security

☑ Use API security definitions

Consumes

□ Use API consume types

Table of contents

21. In the submenu navigation bar, click the **Assemble** tab to open the assemble view.
22. Access the **invoke** policy property sheet by clicking the **invoke** label.
23. Populate the **Title**, **Description**, and **URL** fields according to the following table. When called, your API now invokes the existing Branches API and uses its response. In this tutorial, no transformations are applied to the response of this API and so the entirety of the response is returned to the caller. You can see this response at https://apim-services.mybluemix.net/banka/v1/branches.

    Table 4. invoke fields

Field	Value
Title	`Branches Invoke`
Description	`Invoke an API to retrieve the status of all branches in the BankA system`
URL	`https://apim-services.mybluemix.net/banka/v1/branches`

    Leave the remaining fields with their default values.

24. Click the **Save** icon 💾 to save your changes.
25. Click the **Source** tab to view the OpenAPI (Swagger 2.0) definition of your API. All the configuration you have performed is included in this definition, either as part of the standard OpenAPI (Swagger 2.0) schema, or as part of the `x-ibm-configuration` extension.

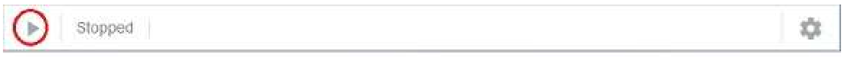
Your REST API is defined. This example helped you to configure the REST API invocation through the Assembly tool. No coding was required. The definitions help developers who are creating applications and integrating with the BankA Branches REST API for the first time.

Testing the REST API

Test your REST API to ensure that it is defined and implemented correctly.

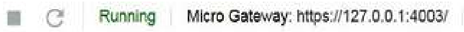
To test the REST API, complete the following steps:

1. Start the local test servers by completing the following steps:
    a. In the test console at the bottom of the screen, click the **Start the servers** icon:

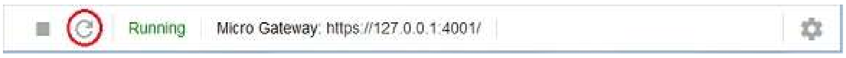
    Stopped

    b. Wait until the `Running` message is displayed:

    Running | Micro Gateway: https://127.0.0.1:4003/

    Depending on your project configuration and whether other processes are running, a different port number might be displayed.

> **Note**
>
> If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking the **Restart the servers** icon
>
> 
>
> Running | Micro Gateway: https://127.0.0.1:4001/
>
> .

2. Click the **Assemble** tab.
3. Click the **Test** icon ▶. The test tool opens, overlaying the palette.
4. In the **Operation** section, use the drop-down menu to select the **get /details** operation.
5. At the bottom of the section, click **Invoke**. The operation is called by the test tool. The response of your API is shown in the test tool.

In other tutorials you get a chance to test the API by using the Developer Portal and API Manager test tools, both of which run online.

Creating a Product and a Plan for the REST API

Create a Product and a Plan so that you can later stage or publish your APIs.

Table of contents

create a Product, complete the following steps:

Click **All APIs** and then click the **Products** tab.

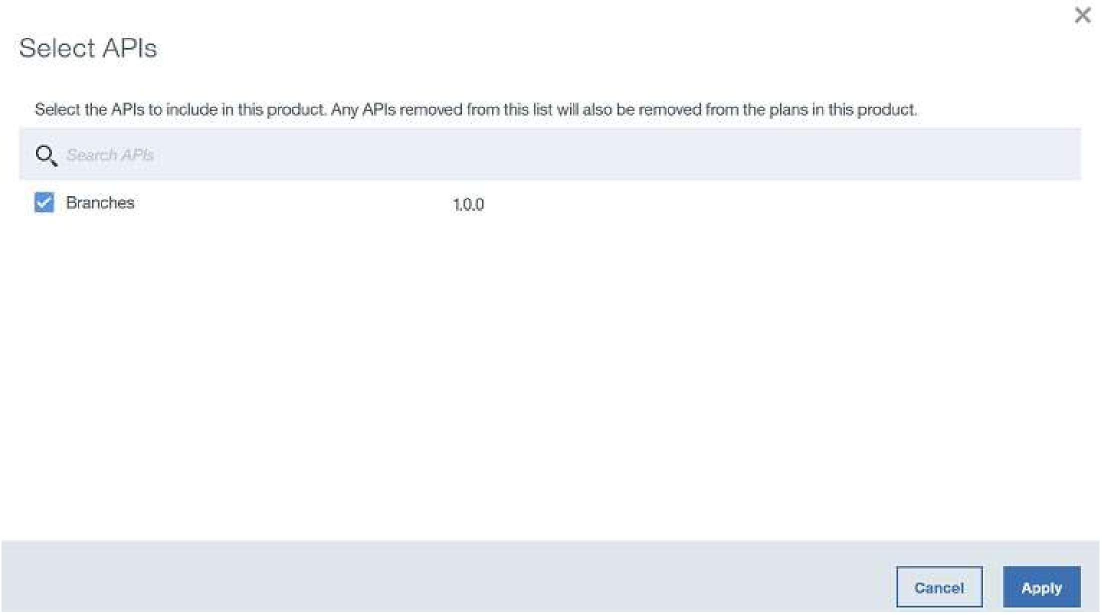
Click **Add** and then click **New Product**. The "New Product" window opens.

Complete the fields as shown in the following table and then click **Create product**.

Table 5. Product fields

Field Name	Value
Title	`Banking Services`
Name	`banking-services`
Version	`1.0.0`

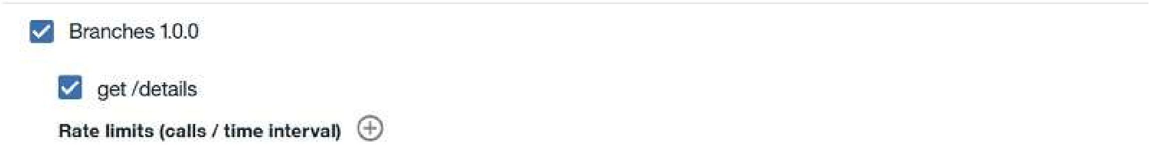
4. In the **Visibility** section you can control who the Product is visible to and who can subscribe to its Plans. The Product visibility is set to "Public" and so anybody will be able to see the Product when it is published to the Developer Portal. When published, the Plans can be subscribed to by "Authenticated users", which refers to users who have accounts in the Developer Portal.
5. In the **APIs** section, click the **Add API** icon ⊕. The Select APIs window opens.
6. Select the **Branches** API.

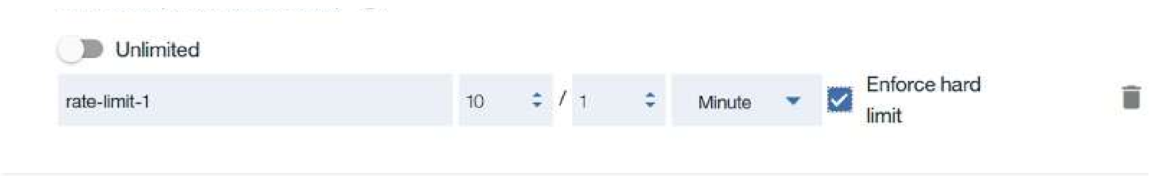


Select APIs

Select the APIs to include in this product. Any APIs removed from this list will also be removed from the plans in this product.

7. Click **Apply**.
8. Expand the Plan titled **Default** that has been automatically created. Because no APIs have been excluded, the only API in the Product is included in the Plan.
9. Enter `Basic` for the **Title** field and the **Name** field.
10. Add a rate limit to your **/details** operation by completing the following steps:
    a. Expand the **Branches** API.
    b. Click **Override rate limit** beside the **get /details** operation.
    c. Use the controls to set the rate limit as 10 requests per 1 minute against **rate-limit-1**, and select **Enforce hard limit**.





Unlimited

rate-limit-1 10 / 1 Minute ☑ Enforce hard limit

Table of contents

. Click the **Save** icon 💾 to save your changes.

u have created the Banking Services Product with the Basic Plan within it, you added the Branches API to the Basic Plan, added a e limit to the `rate-limit-1` operation, and staged your Banking Services Product to your development environment.

> **Note**
>
> These steps are not necessary to test your APIs offline, but a Product is needed when making your APIs externally available.

Publishing your Product

Publish your Product and the API it contains to make them externally available for later tutorials.

1. In the API Designer, click **Publish** and then click **Add and Manage Targets**.
2. Click **Add IBM Bluemix target**.
3. Provide credentials for your Bluemix account and then click **Sign in**.
4. In the **Region** field, select the region you want to publish in.
5. In the **Organization** field, select the provider organization that you want to publish with.
6. Select **Sandbox** from the list of Catalogs. If you have a large number of Catalogs, use the **Search** field to refine the list of Catalogs.
7. Click **Save**.
8. Click **Publish** and then click your newly created target.
9. Select **Select specific products** and then select your **Banking Services** Product.
10. Click **Publish**. Your Product is now available through your gateway server and visible in both API Manager and the Developer Portal,

Your Product and the API it contains are published to your specified target.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a REST API definition.
- Tested a REST API.
- Created a Product that contains a Plan.
- Published a Product to a Catalog.

What to do next

- Discover and use your API in the Developer Portal.
- Secure your API with a client ID and secret.
- Secure your API by using an LDAP registry.
- Secure your API by using OAuth.

Parent topic:

Tutorials for working with API definitions that call an existing endpoint

Table of contents

Sign In

0 Comments IBM Knowledge Center

♡ Recommend Share Sort by Best

Nothing in this discussion yet.

Subscribe Add Disqus to your siteAdd DisqusAdd Privacy

Contact Privacy Terms of use Accessibility Feedback Cookie preferences

English