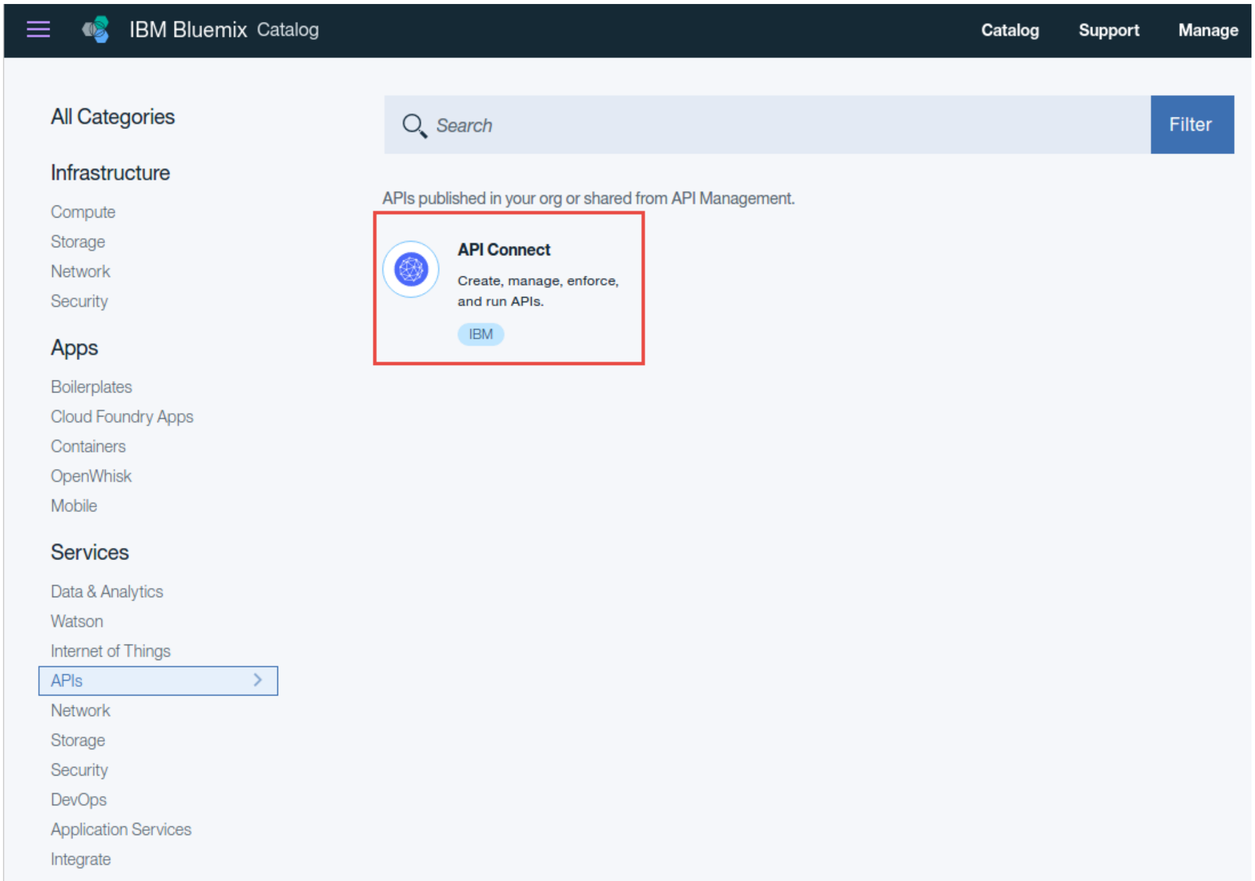
APIC with Secure Gateway Notes

Create API Connect Service if not already deployed

From Catalog

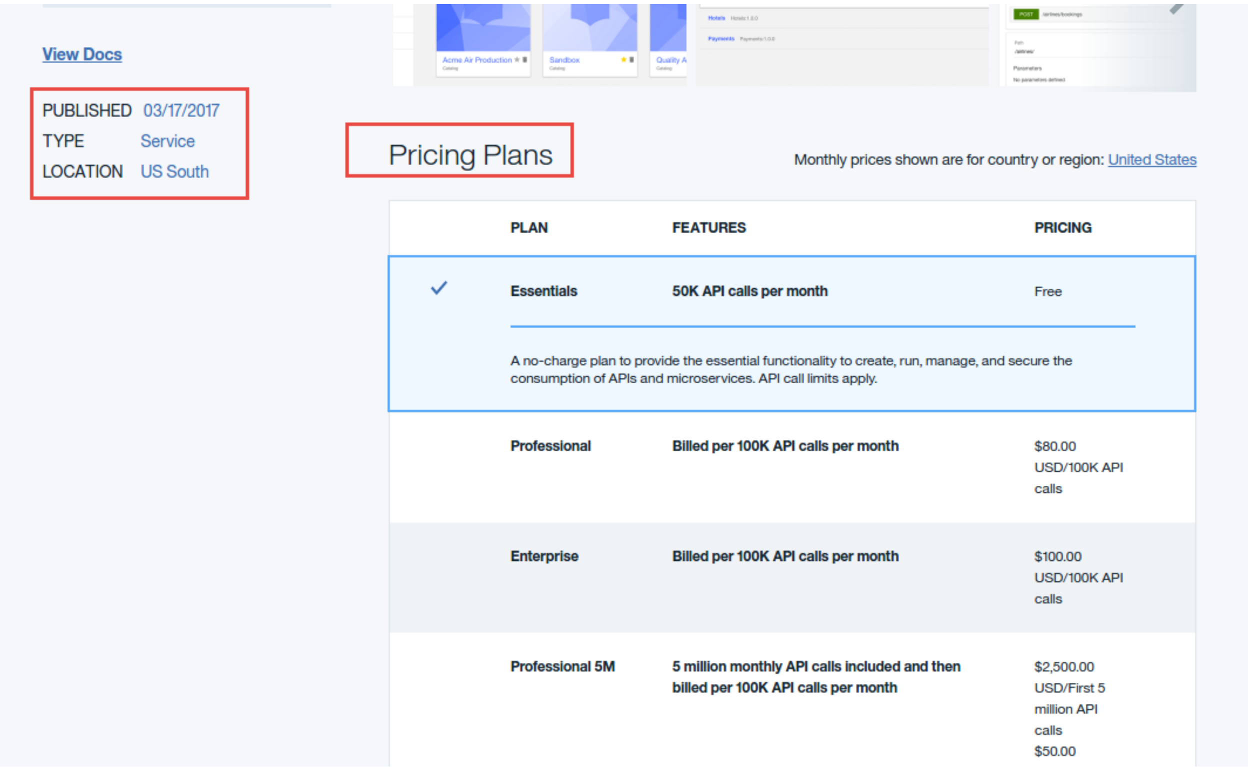


APIs published in your org or shared from API Management.

API Connect

Create, manage, enforce, and run APIs.

Review Plans



View Docs

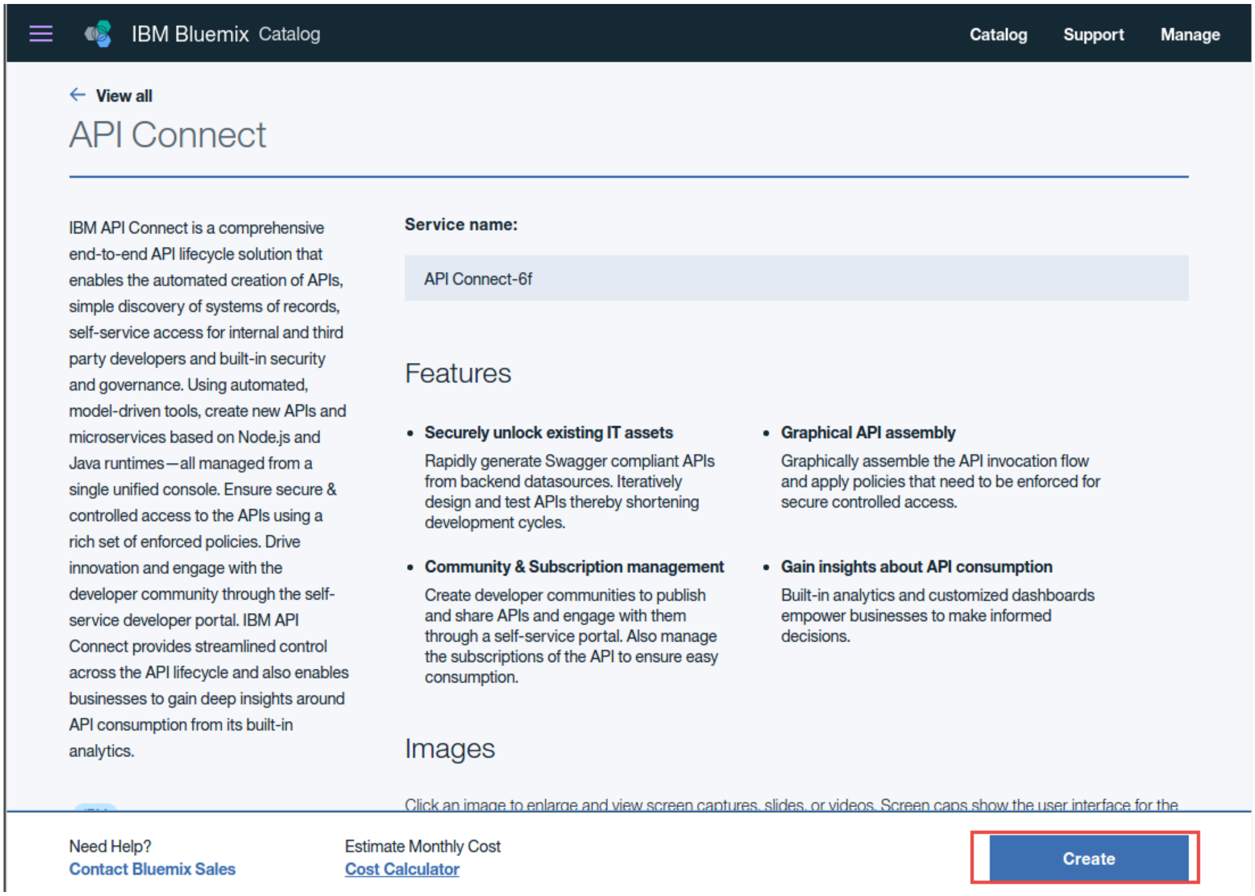
PUBLISHED 03/17/2017
TYPE Service
LOCATION US South

Pricing Plans

Monthly prices shown are for country or region: United States

	PLAN	FEATURES	PRICING
✓	**Essentials**	**50K API calls per month**	Free
	A no-charge plan to provide the essential functionality to create, run, manage, and secure the consumption of APIs and microservices. API call limits apply.		
	Professional	**Billed per 100K API calls per month**	$80.00 USD/100K API calls
	Enterprise	**Billed per 100K API calls per month**	$100.00 USD/100K API calls
	Professional 5M	**5 million monthly API calls included and then billed per 100K API calls per month**	$2,500.00 USD/First 5 million API calls $50.00

Create



IBM Bluemix Catalog

Catalog Support Manage

← View all

API Connect

IBM API Connect is a comprehensive end-to-end API lifecycle solution that enables the automated creation of APIs, simple discovery of systems of records, self-service access for internal and third party developers and built-in security and governance. Using automated, model-driven tools, create new APIs and microservices based on Node.js and Java runtimes—all managed from a single unified console. Ensure secure & controlled access to the APIs using a rich set of enforced policies. Drive innovation and engage with the developer community through the self-service developer portal. IBM API Connect provides streamlined control across the API lifecycle and also enables businesses to gain deep insights around API consumption from its built-in analytics.

Service name:

API Connect-6f

Features

- **Securely unlock existing IT assets**
  Rapidly generate Swagger compliant APIs from backend datasources. Iteratively design and test APIs thereby shortening development cycles.
- **Graphical API assembly**
  Graphically assemble the API invocation flow and apply policies that need to be enforced for secure controlled access.
- **Community & Subscription management**
  Create developer communities to publish and share APIs and engage with them through a self-service portal. Also manage the subscriptions of the API to ensure easy consumption.
- **Gain insights about API consumption**
  Built-in analytics and customized dashboards empower businesses to make informed decisions.

Images

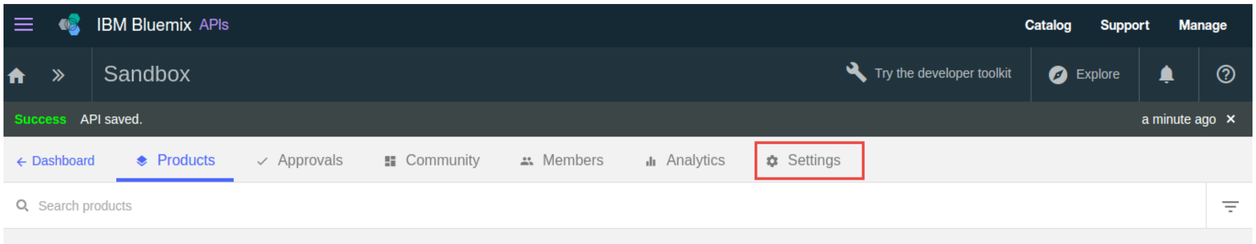
Click an image to enlarge and view screen captures, slides, or videos. Screen caps show the user interface for the

Need Help?
Contact Bluemix Sales

Estimate Monthly Cost
Cost Calculator

Create

Instance Created. Select Sandbox Catalog then Settings



IBM Bluemix APIs

Catalog Support Manage

Sandbox

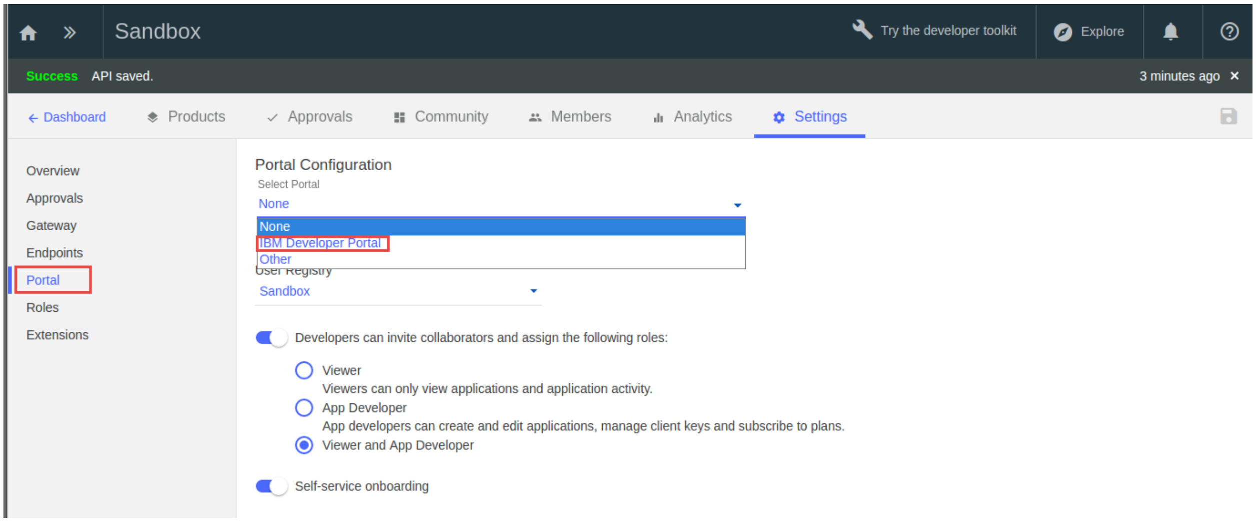
Try the developer toolkit Explore

Success API saved. a minute ago ×

← Dashboard Products Approvals Community Members Analytics Settings

Search products

Select Portal. In Portal Configuration select IBM Developer Portal.



Success API saved.

3 minutes ago ×

Portal Configuration

Select Portal: None

User Registry: Sandbox

Developers can invite collaborators and assign the following roles:

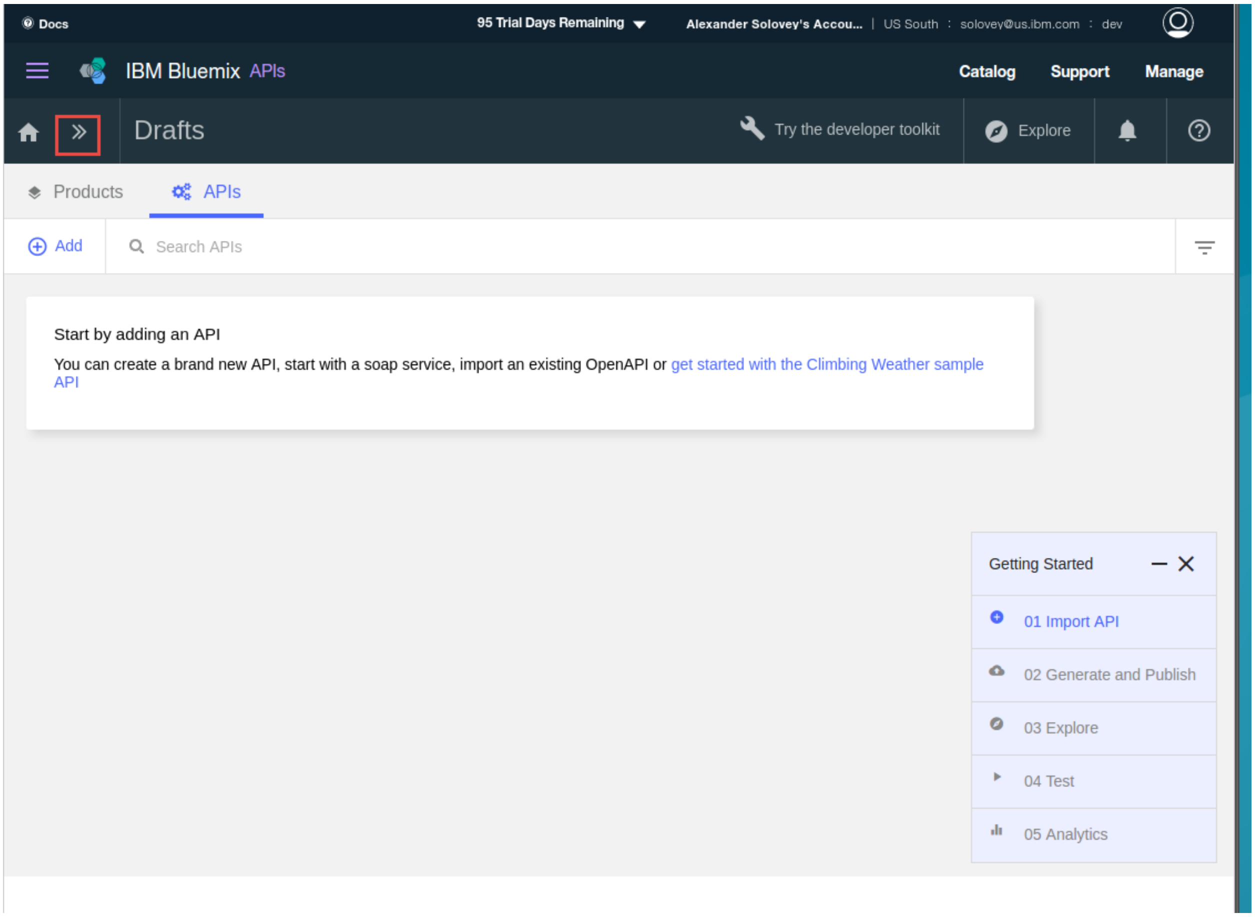
○ Viewer
Viewers can only view applications and application activity.

○ App Developer
App developers can create and edit applications, manage client keys and subscribe to plans.

● Viewer and App Developer

Self-service onboarding

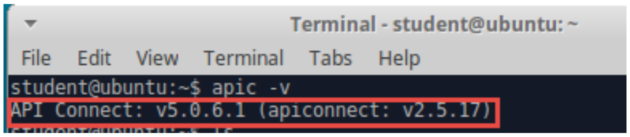
Select Menu -> Admin



Start by adding an API

You can create a brand new API, start with a soap service, import an existing OpenAPI or get started with the Climbing Weather sample API

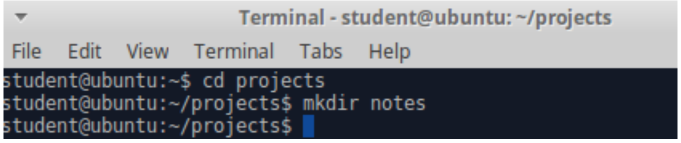
Used APIC PoT Ubuntu image


Terminal - student@ubuntu: ~
File Edit View Terminal Tabs Help

```
student@ubuntu:~$ apic -v
API Connect: v5.0.6.1 (apiconnect: v2.5.17)
student@ubuntu:~$ ls
```

Follow article part 1

Create empty dir projects then notes (forgot)


Terminal - student@ubuntu: ~/projects
File Edit View Terminal Tabs Help

```
student@ubuntu:~$ cd projects
student@ubuntu:~/projects$ mkdir notes
student@ubuntu:~/projects$ 
```

1.1 - Creating a notes Application

See Lab 1 sections 1.1 – 1.3 from PoT Lab instructions at:

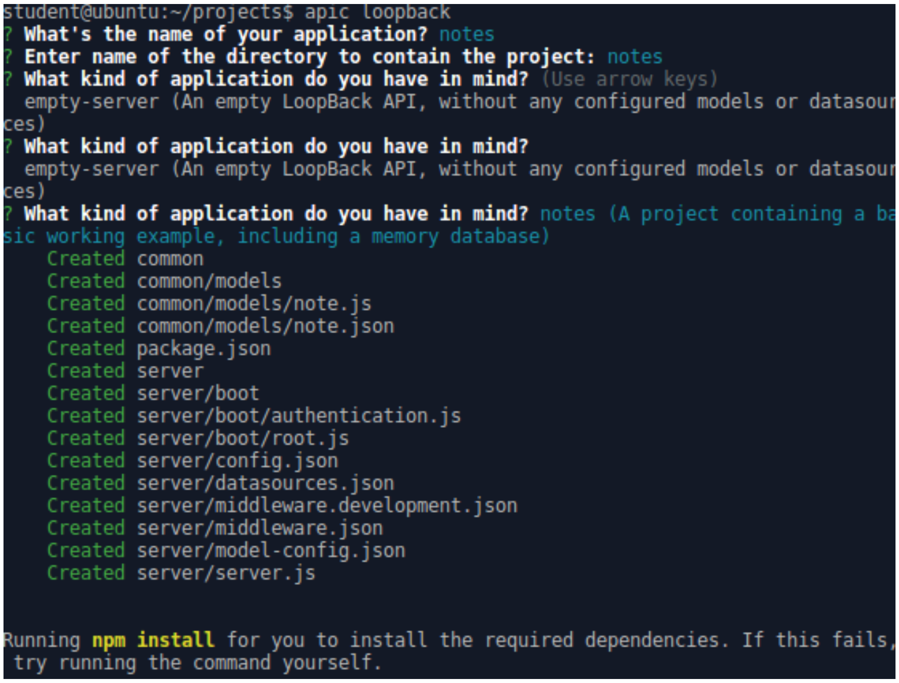
https://ibm-apiconnect.github.io/pot/lab1.html#creating-a-notes-application

We will use the API Connect Developer Toolkit command line interface to create the initial application and explore the created artifacts.

From the terminal command line type:

apic loopback

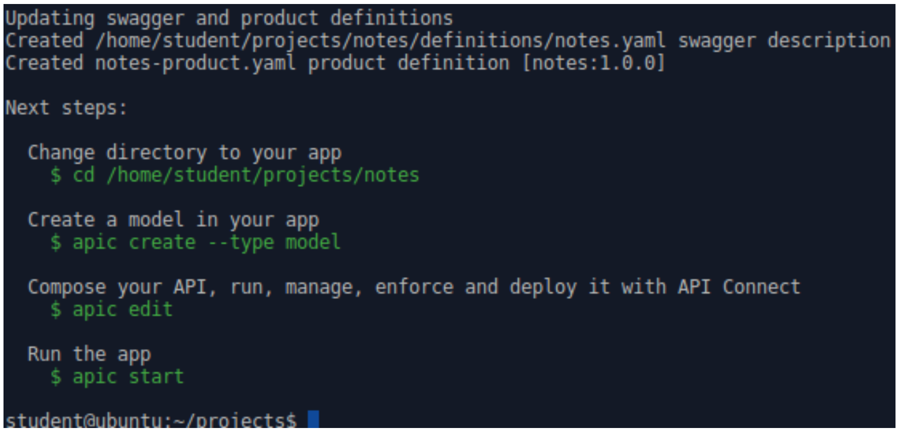
This command starts the application generator, Yeoman, to help scaffold the new project. Just press enter for first two questions and then select notes for the kind of application.



```
student@ubuntu:~/projects$ apic loopback
? What's the name of your application? notes
? Enter name of the directory to contain the project: notes
? What kind of application do you have in mind? (Use arrow keys)
  empty-server (An empty LoopBack API, without any configured models or datasour
ces)
? What kind of application do you have in mind?
  empty-server (An empty LoopBack API, without any configured models or datasour
ces)
? What kind of application do you have in mind? notes (A project containing a ba
sic working example, including a memory database)
    Created common
    Created common/models
    Created common/models/note.js
    Created common/models/note.json
    Created package.json
    Created server
    Created server/boot
    Created server/boot/authentication.js
    Created server/boot/root.js
    Created server/config.json
    Created server/datasources.json
    Created server/middleware.development.json
    Created server/middleware.json
    Created server/model-config.json
    Created server/server.js


Running npm install for you to install the required dependencies. If this fails,
 try running the command yourself.
```

This creates an application named “notes” in a directory of the same name. The application is a basic Notes application. You will see a lot of messages printed to the command line window. It is creating a few resources for you and installing the various node modules. Once the node modules are loaded you’ll notice that the process creates swagger and product definitions for you. Finally, the process displays some hints about what to do next. Since we’ve been given such lovely suggestions about what to do next, we may as well follow the first one at least.



```
Updating swagger and product definitions
Created /home/student/projects/notes/definitions/notes.yaml swagger description
Created notes-product.yaml product definition [notes:1.0.0]

Next steps:

  Change directory to your app
    $ cd /home/student/projects/notes

  Create a model in your app
    $ apic create --type model

  Compose your API, run, manage, enforce and deploy it with API Connect
    $ apic edit

  Run the app
    $ apic start

student@ubuntu:~/projects$
```

Change directories to the project directory:

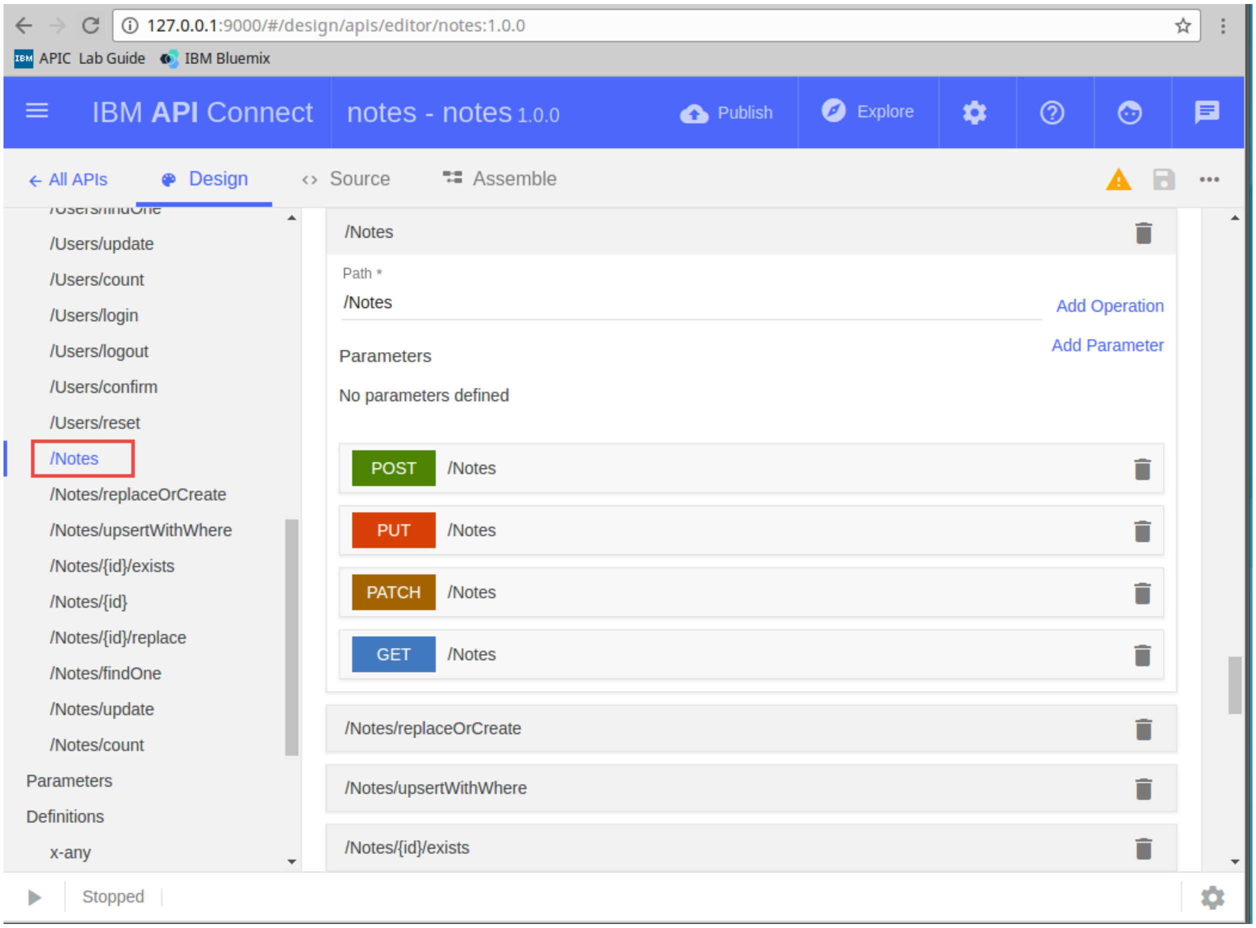
cd notes

1.2 - Launching the notes Application

Now that we’ve explored what is created by the application generator, let’s move on to the API Designer. From the command line:

apic edit

Next, click on the start button located on the bottom panel of the API Designer to launch the notes application.



127.0.0.1:9000/#/design/apis/editor/notes:1.0.0

Path *

/Notes

Parameters

No parameters defined

On a Windows environment, you might see 2 node windows pop up on your screen. Minimize, but do not close these windows.

Once start completes, you should see a screen similar to this:

Notice that once the application is up and running, stop and restart buttons will appear on the bottom side of the screen:

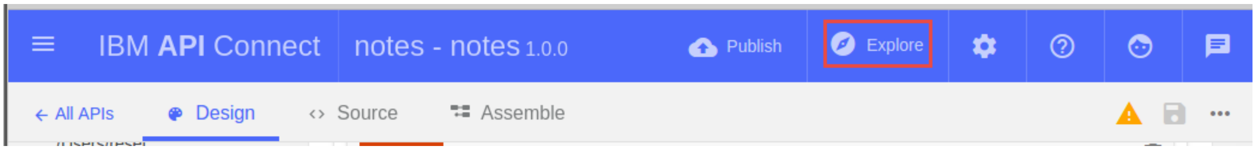
At this point we're ready to Explore and test our services.

Note:

We used the web-based editor to launch the application. There's also a command provided with the API Connect Developer Toolkit that can be utilized from the terminal to lauch the application: apic start

1.3 - Testing the notes Application

Click the Explore button to switch to the API Explorer view.



You will see all the exposed service paths displayed.

Now we're going to test the services using the GUI presented on the explore screen. You'll notice that on the left several REST services are defined for us. In particular, take a look POST /notes and GET /notes.

If you're not familiar with GET and POST, they are HTTP methods (sometimes called verbs). The POST method is used for creation calls to the service. The GET method is used to retrieve information from a service. In this case, we see that both methods are used against the /notes path. So, POST will create a note and GET will retrieve all the notes that have been created.

Start by creating a couple of notes. Click the POST /notes link in the left column. The other columns will scroll so that you're looking at information and controls pertaining to the POST /notes service.

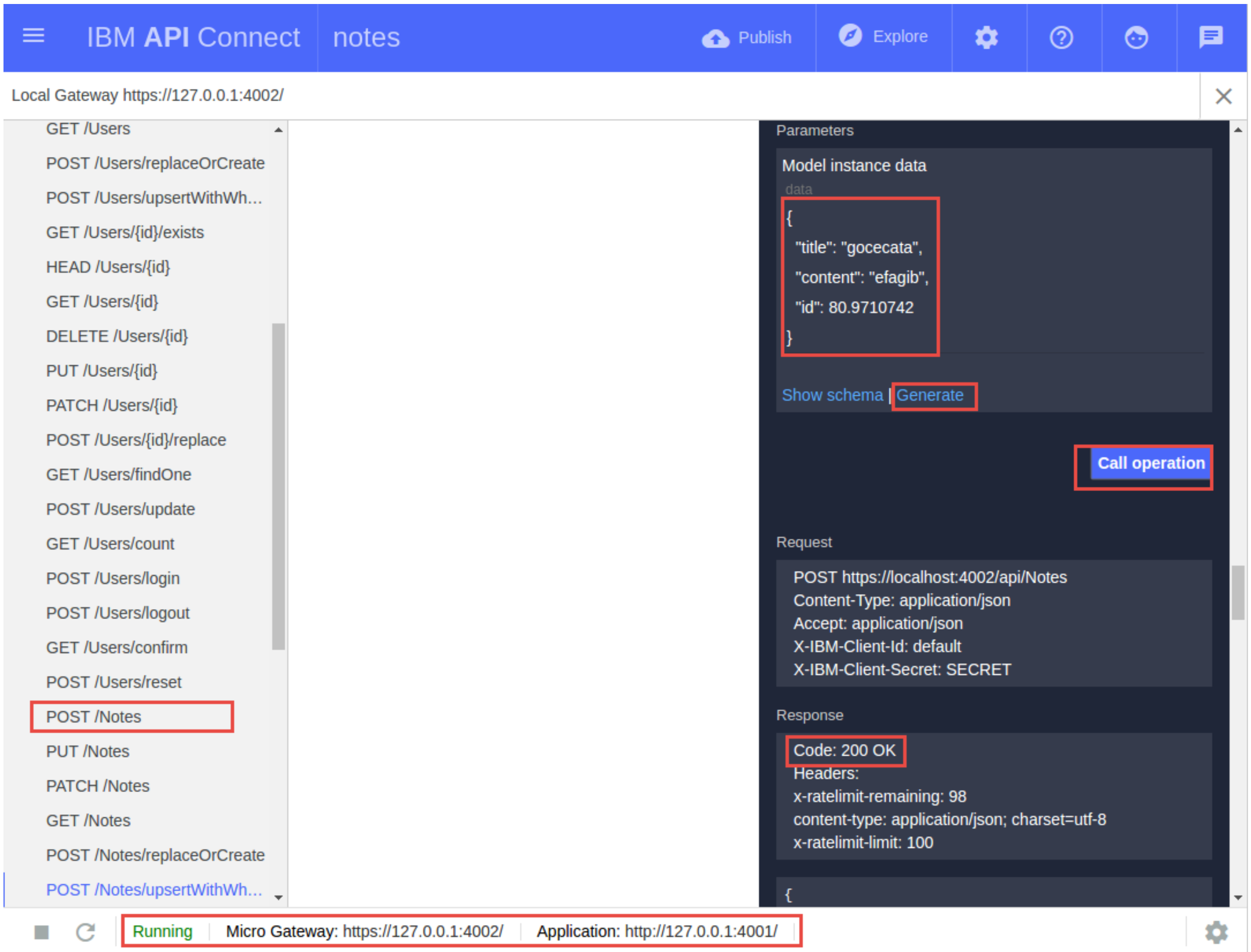
To test creation of a note, scroll down in the right hand column until the Call operation button is visible at the bottom. Just above the Call operation button you'll see a Generate link. This link will generate dummy data for you to create a test call to this service.

Press the Generate link to generate some sample data.

Your data will look different, but you're ready to test the service.

Go ahead and press the Call operation button and scroll down to the Response section to see the results.

In the results, you should see a Code: 200 OK which indicates that a new note was created. If you received a different response, see the troubleshooting steps below.



```
{
 "title": "gocecata",
 "content": "efagib",
 "id": 80.9710742
}
```

```
POST https://localhost:4002/api/Notes
Content-Type: application/json
Accept: application/json
X-IBM-Client-Id: default
X-IBM-Client-Secret: SECRET
```

```
Code: 200 OK
Headers:
x-ratelimit-remaining: 98
content-type: application/json; charset=utf-8
x-ratelimit-limit: 100
```

```
{
```

Troubleshooting:

You may see an error displayed that mentions a CORS issue. This has to do with certificates in your browser. Go ahead an click the given link to rectify this, accept any certificate, close the opened tab, and press the Call operation button again. Additionally, be sure not to skip step 5, as doing a POST operation without generating payload will cause an error.

Troubleshooting:

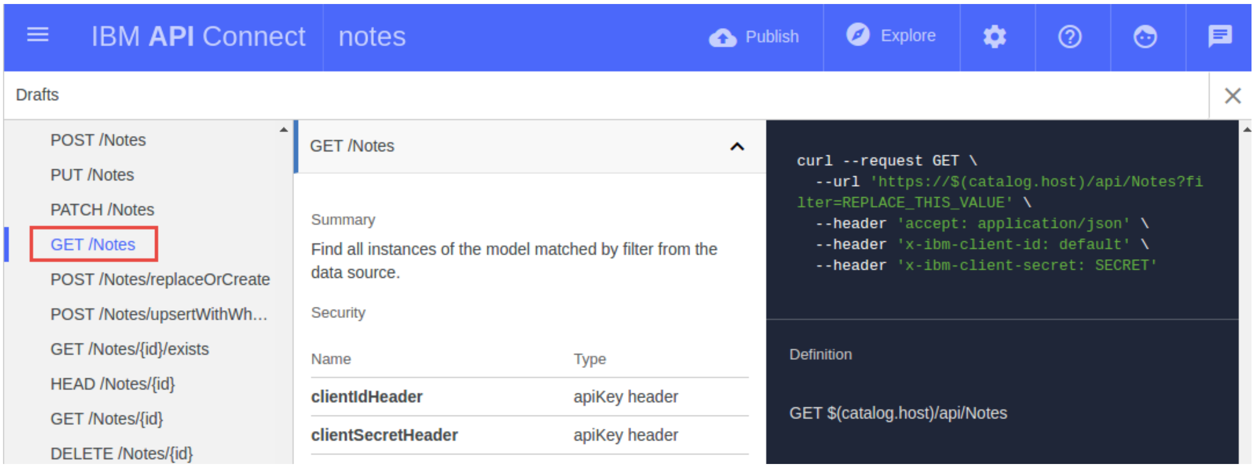
If you see a 500 error like the one below, make sure you press the Generate button before you press the Call operation button again. Otherwise, you're trying to create a duplicate note.

Once you have created one note, go ahead and create another one or two.

Note:

It should be noted that you don't need to use the Generate link. You can type data directly into the Parameters. You can also use Generate to create a template for you to use and then change the generated parameters. You may also notice that not all the parameters are always generated. This is because only the title parameter is required. Try pressing Generate several times to get a feel for how it works.

Finally, let's test the GET /notes service. We should have two, three, or more notes created at this point. In the left hand column click the GET /notes link.



Summary

Find all instances of the model matched by filter from the data source.

Security

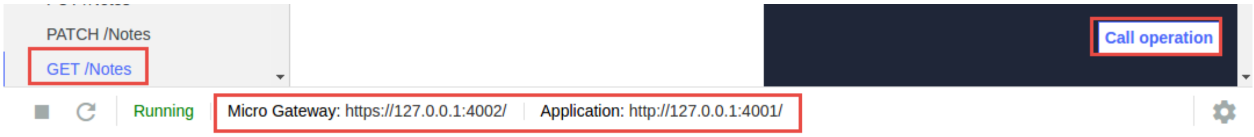
Name	Type
clientIdHeader	apiKey header
clientSecretHeader	apiKey header

```
curl --request GET \
  --url 'https://$(catalog.host)/api/Notes?fi
lter=REPLACE_THIS_VALUE' \
  --header 'accept: application/json' \
  --header 'x-ibm-client-id: default' \
  --header 'x-ibm-client-secret: SECRET'
```

Definition

GET $(catalog.host)/api/Notes

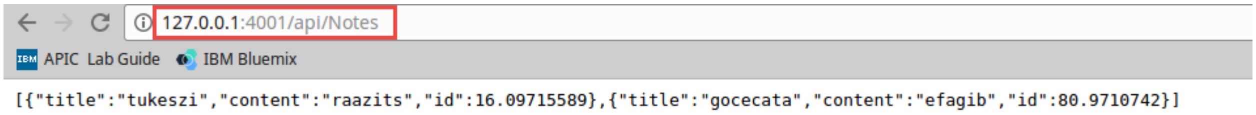
Scroll down to the Call operation button and press it. Then scroll down to the results.



Micro Gateway: https://127.0.0.1:4002/ Application: http://127.0.0.1:4001/

You should see all the notes that you generated in the result set.

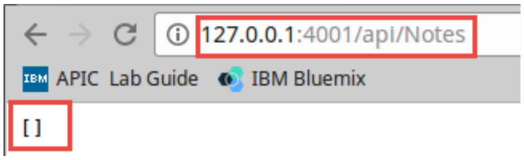
Or use this url http://127.0.0.1:4001/api/Notes in the browser



```
[{"title":"tukeszi","content":"raazits","id":16.09715589},{"title":"gocecata","content":"efagib","id":80.9710742}]
```

Important:

If you see an empty array, [], as your result, then you've not successfully created any notes. This is also true if you stop the application and restart it. With the notes example, we're using an in-memory database which means that nothing is persisted to disk. So, it is lost when the server is stopped and restarted



```
[]
```

.

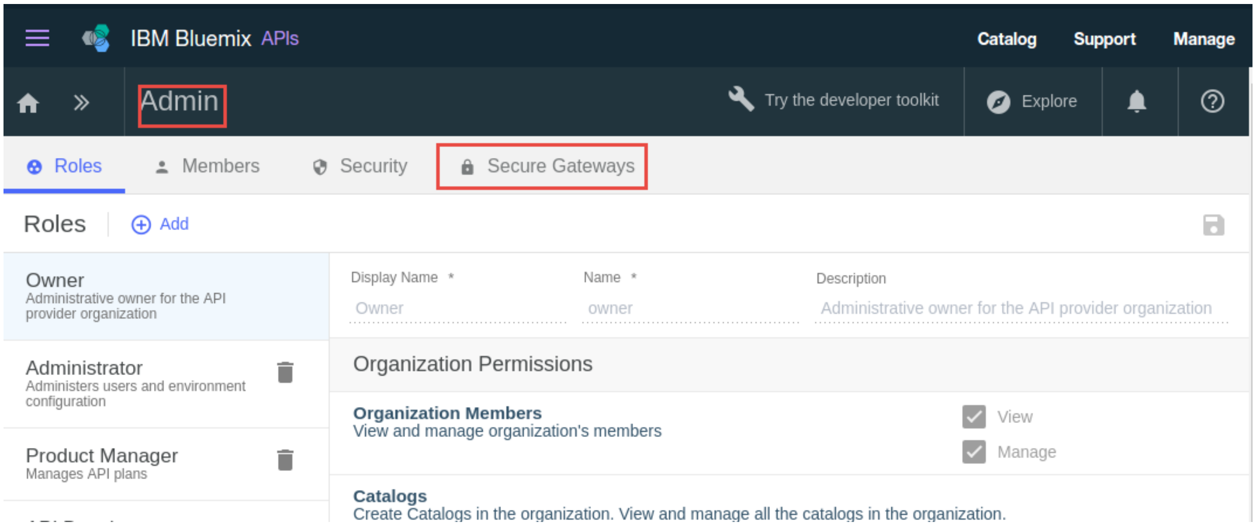
At this point, we are done testing the app locally.

API Connect Secure Gateway Integration Part 2 : Leveraging on-premise API resources

https://developer.ibm.com/apiconnect/2017/01/13/api-connect-secure-gateway-integration-part-2-leveraging-on-premise-api-resources/

In part 2, we will expose this newly created on-premise API securely using the Secure Gateway functionality integrated into API Connect.

We first log into our Bluemix Region where API Connect is provisioned and navigate to the Admin section. From there we will select the Secure Gateways tab.



Owner
Administrative owner for the API provider organization

Administrator
Administers users and environment configuration

Product Manager
Manages API plans

Display Name * Owner

Name * owner

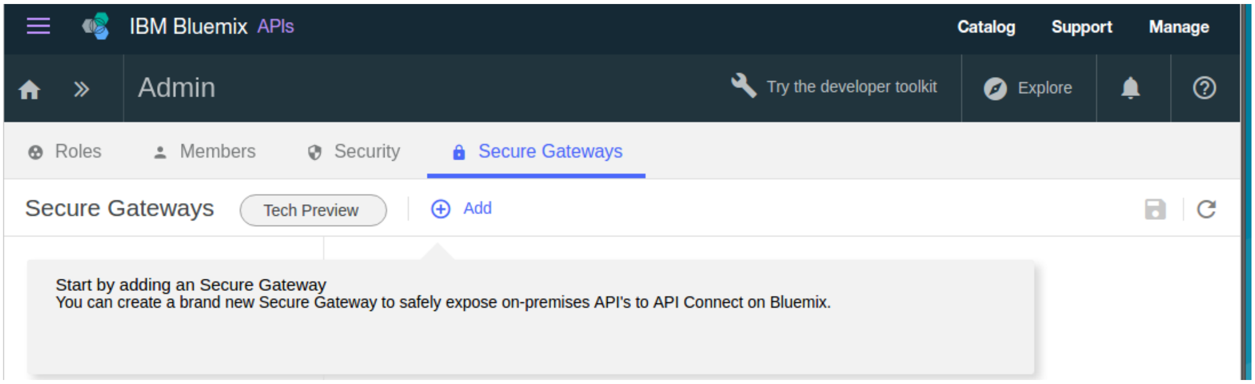
Description Administrative owner for the API provider organization

Organization Permissions

Organization Members
View and manage organization's members
☑ View
☑ Manage

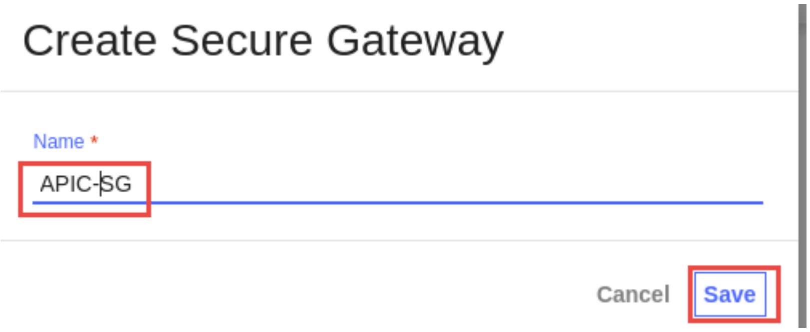
Catalogs
Create Catalogs in the organization. View and manage all the catalogs in the organization.

Click the “Add” link to create our first Secure Gateway.



Start by adding an Secure Gateway
You can create a brand new Secure Gateway to safely expose on-premises API's to API Connect on Bluemix.

Give it a name and click Save

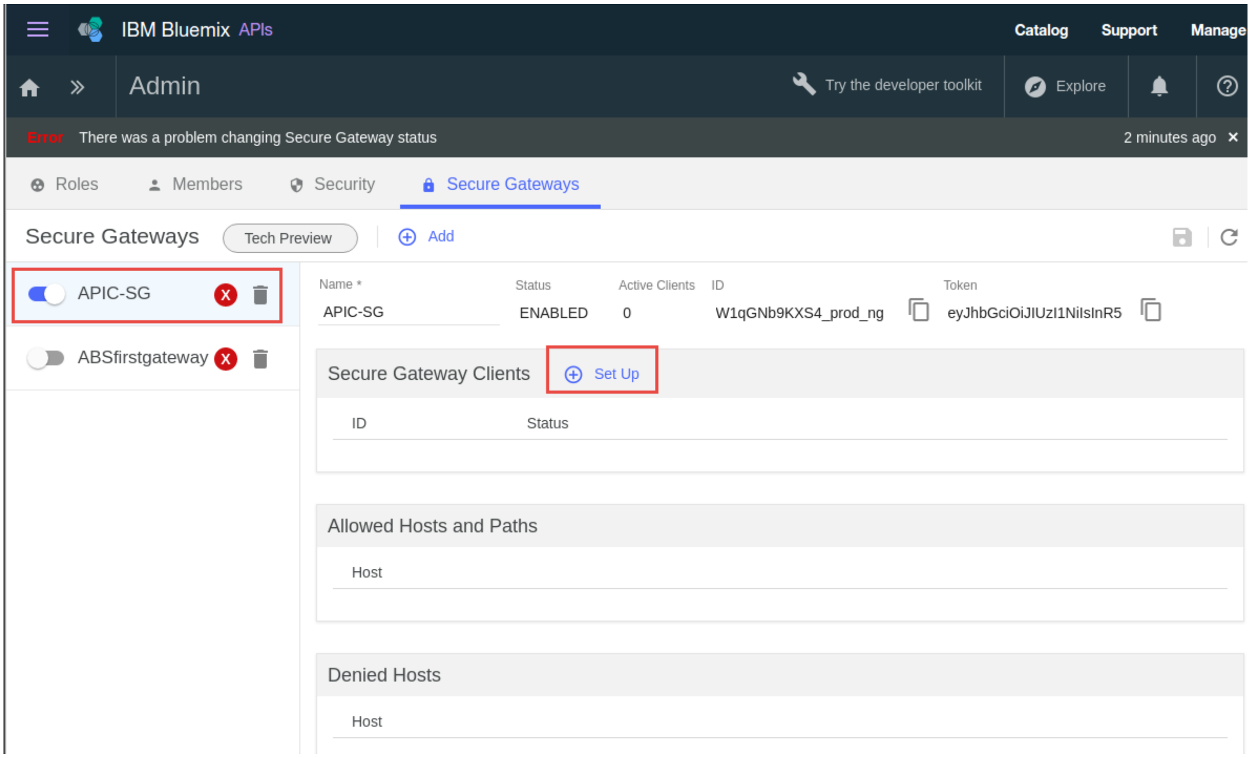


Create Secure Gateway

Name *

APIC-SG

After creating the gateway we need to set up the Secure Gateway client. This will be our connection between the on-premise API and API Connect. From the Secure Gateway Tab we will select the "Set Up" link.



Error There was a problem changing Secure Gateway status 2 minutes ago ×

Secure Gateways

Name *	Status	Active Clients	ID	Token
APIC-SG	ENABLED	0	W1qGNb9KXS4_prod_ng	eyJhbGciOiJIUzI1NiIsInR5

Secure Gateway Clients

ID	Status

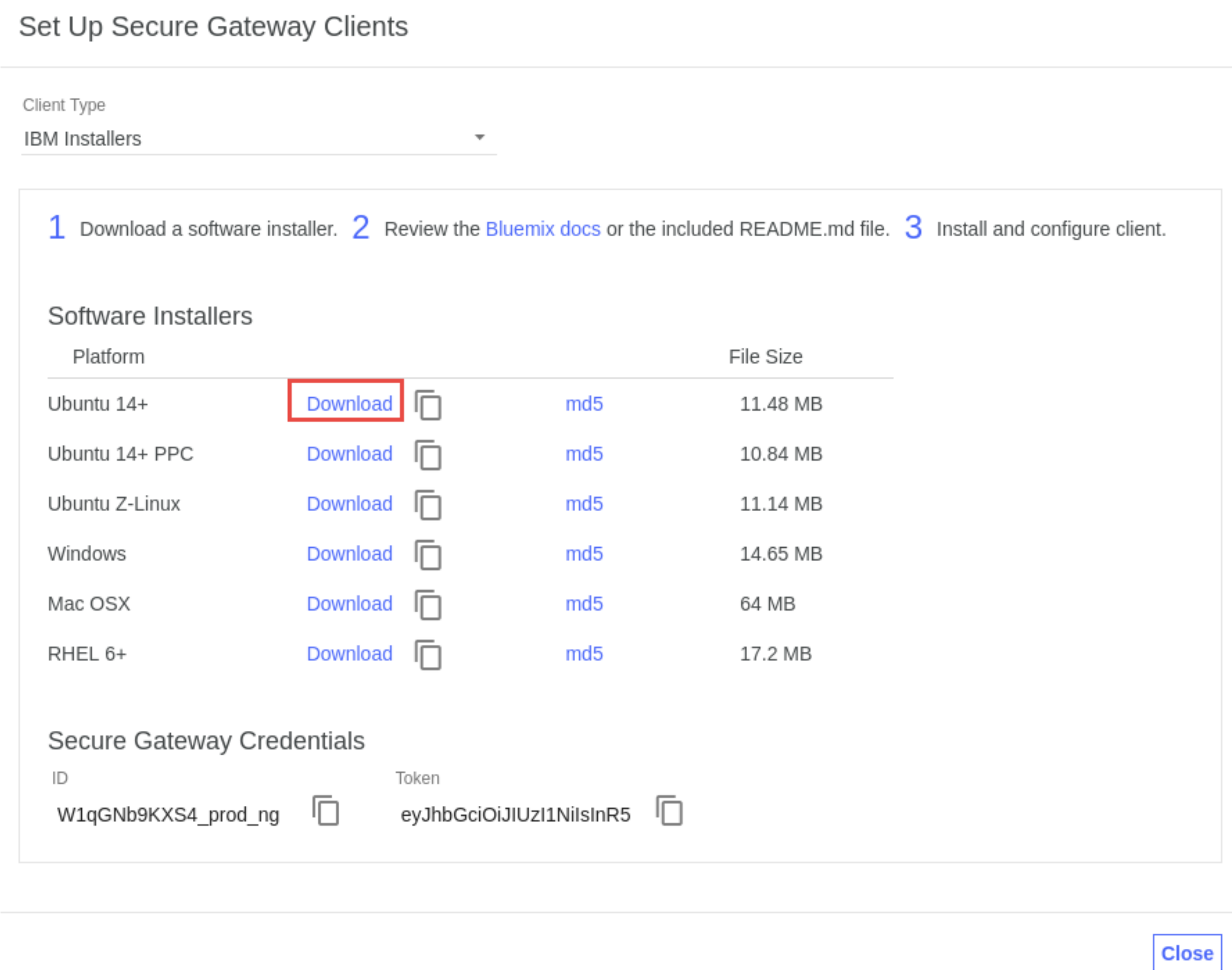
Allowed Hosts and Paths

Host

Denied Hosts

Host

This will bring up the page to download the Secure Gateway Client.



Set Up Secure Gateway Clients

Client Type
IBM Installers

1 Download a software installer. 2 Review the Bluemix docs or the included README.md file. 3 Install and configure client.

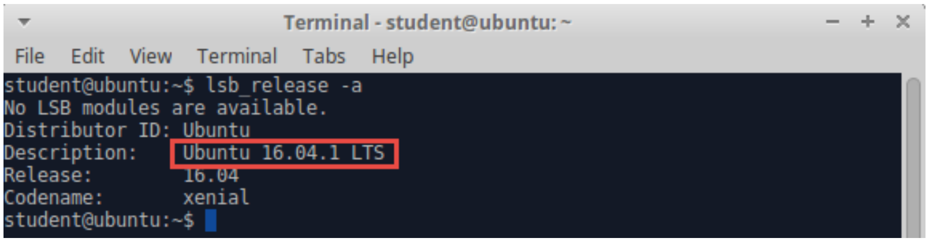
Software Installers

Platform			File Size
Ubuntu 14+	Download	md5	11.48 MB
Ubuntu 14+ PPC	Download	md5	10.84 MB
Ubuntu Z-Linux	Download	md5	11.14 MB
Windows	Download	md5	14.65 MB
Mac OSX	Download	md5	64 MB
RHEL 6+	Download	md5	17.2 MB

Secure Gateway Credentials

ID	Token
W1qGNb9KXS4_prod_ng	eyJhbGciOiJIUzI1NiIsInR5

Check Ubuntu version of PoT VM

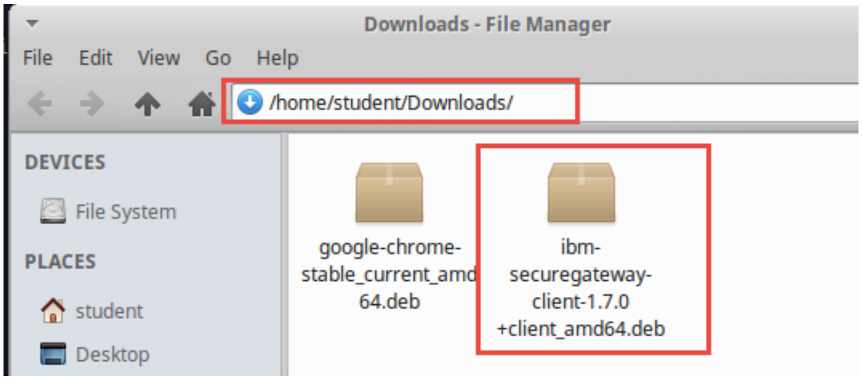


Terminal - student@ubuntu: ~
File Edit View Terminal Tabs Help

```
student@ubuntu:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.1 LTS
Release:        16.04
Codename:       xenial
student@ubuntu:~$
```

Select the installer you wish to use and copy the ID and token for the gateway. You can use the following guide for a more detail explanation of how to setup a Secure Gateway Client: https://console.ng.bluemix.net/docs/services/SecureGateway/sg_021.html#sg_021

Show downloaded file in finder:



/home/student/Downloads/

google-chrome-stable_current_amd64.deb

ibm-securegateway-client-1.7.0+client_amd64.deb

Installing the client - Ubuntu

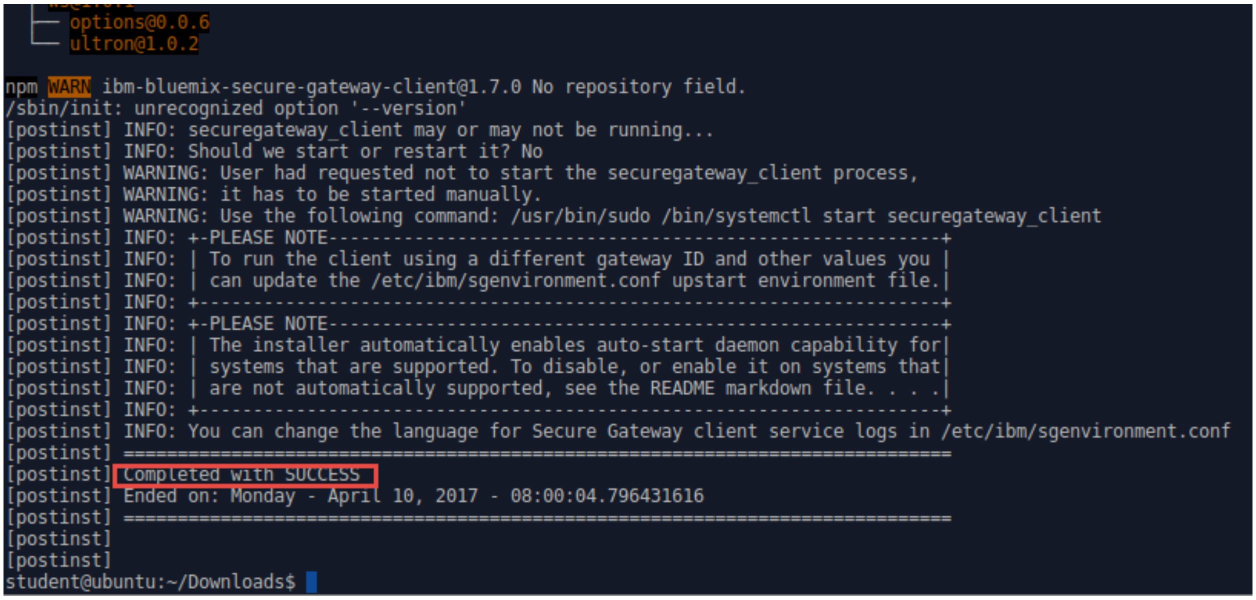
You require root or administrative privileges to install the client on an Ubuntu operating system.

Install the Secure Gateway client. For example, using a Debian package manager, if you are trying to install version 1.4.1 issue the following command.

sudo dpkg -i ibm-securegateway-client-1.4.1+client_amd64.deb

When the client installer starts, you are prompted for the following information:

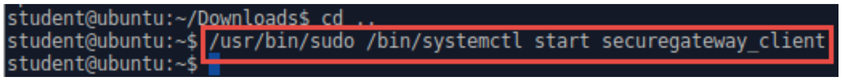
Note: You do not have to answer the prompts. They will all take the defined default, or be left blank in thesgenvironment.conf file. This allows the installation process to run without user interaction.



```
  ├── options@0.0.6
  └── ultron@1.0.2

npm WARN ibm-bluemix-secure-gateway-client@1.7.0 No repository field.
/sbin/init: unrecognized option '--version'
[postinst] INFO: securegateway_client may or may not be running...
[postinst] INFO: Should we start or restart it? No
[postinst] WARNING: User had requested not to start the securegateway_client process,
[postinst] WARNING: it has to be started manually.
[postinst] WARNING: Use the following command: /usr/bin/sudo /bin/systemctl start securegateway_client
[postinst] INFO: +-PLEASE NOTE---------------------------------------------------------+
[postinst] INFO: | To run the client using a different gateway ID and other values you |
[postinst] INFO: | can update the /etc/ibm/sgenvironment.conf upstart environment file.|
[postinst] INFO: +---------------------------------------------------------------------+
[postinst] INFO: +-PLEASE NOTE---------------------------------------------------------+
[postinst] INFO: | The installer automatically enables auto-start daemon capability for|
[postinst] INFO: | systems that are supported. To disable, or enable it on systems that|
[postinst] INFO: | are not automatically supported, see the README markdown file. . . .|
[postinst] INFO: +---------------------------------------------------------------------+
[postinst] INFO: You can change the language for Secure Gateway client service logs in /etc/ibm/sgenvironment.conf
[postinst] ==============================================================================
[postinst] Completed with SUCCESS
[postinst] Ended on: Monday - April 10, 2017 - 08:00:04.796431616
[postinst] ==============================================================================
[postinst]
[postinst]
student@ubuntu:~/Downloads$
```

Start the client

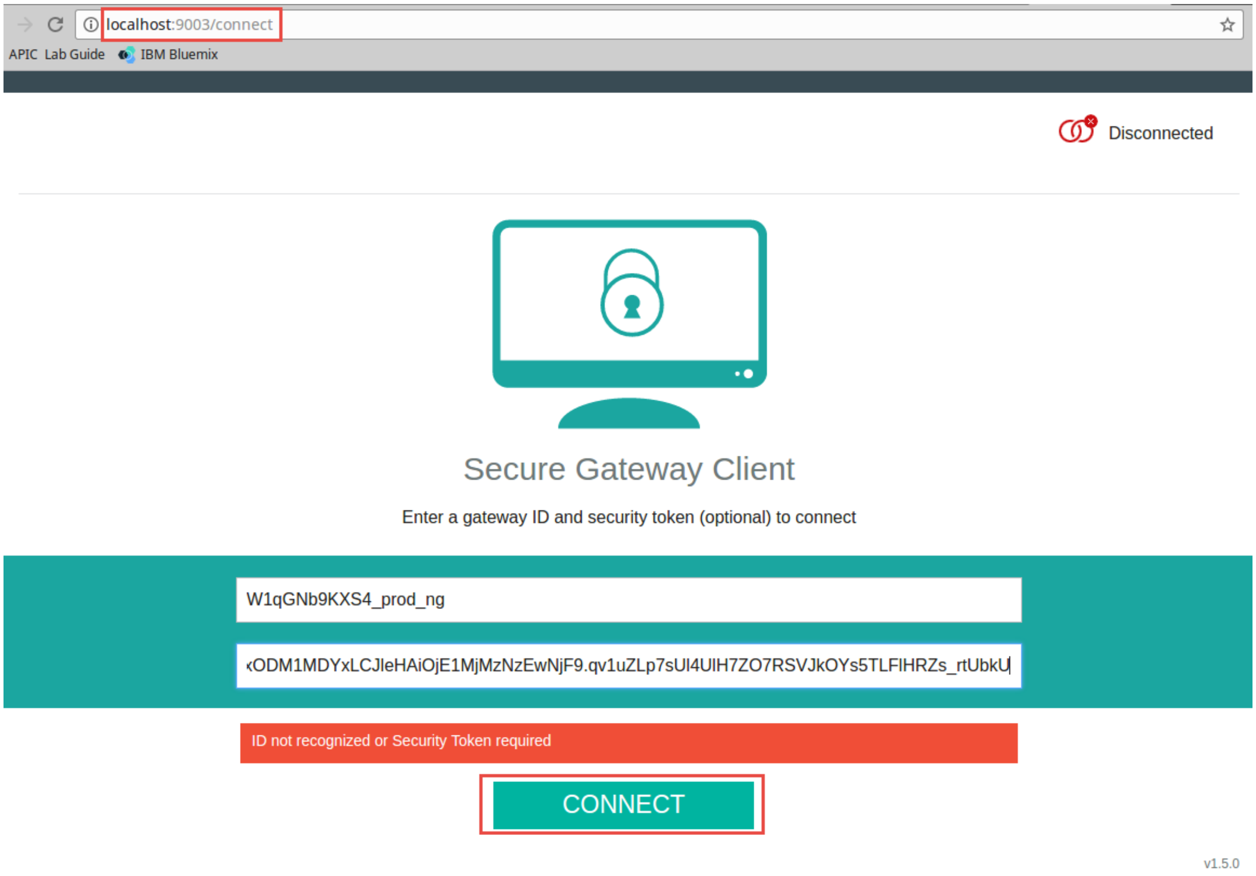


```
student@ubuntu:~/Downloads$ cd ..
student@ubuntu:~$ /usr/bin/sudo /bin/systemctl start securegateway_client
student@ubuntu:~$
```

Configure client ACL

Once the install of the client is finished, you'll need to ensure the hostname and port for the on-premise API is added to the Access Control List of the Secure Gateway Client. This can be accomplished via the Secure Gateway Client's UI:

http://localhost:9003/dashboard



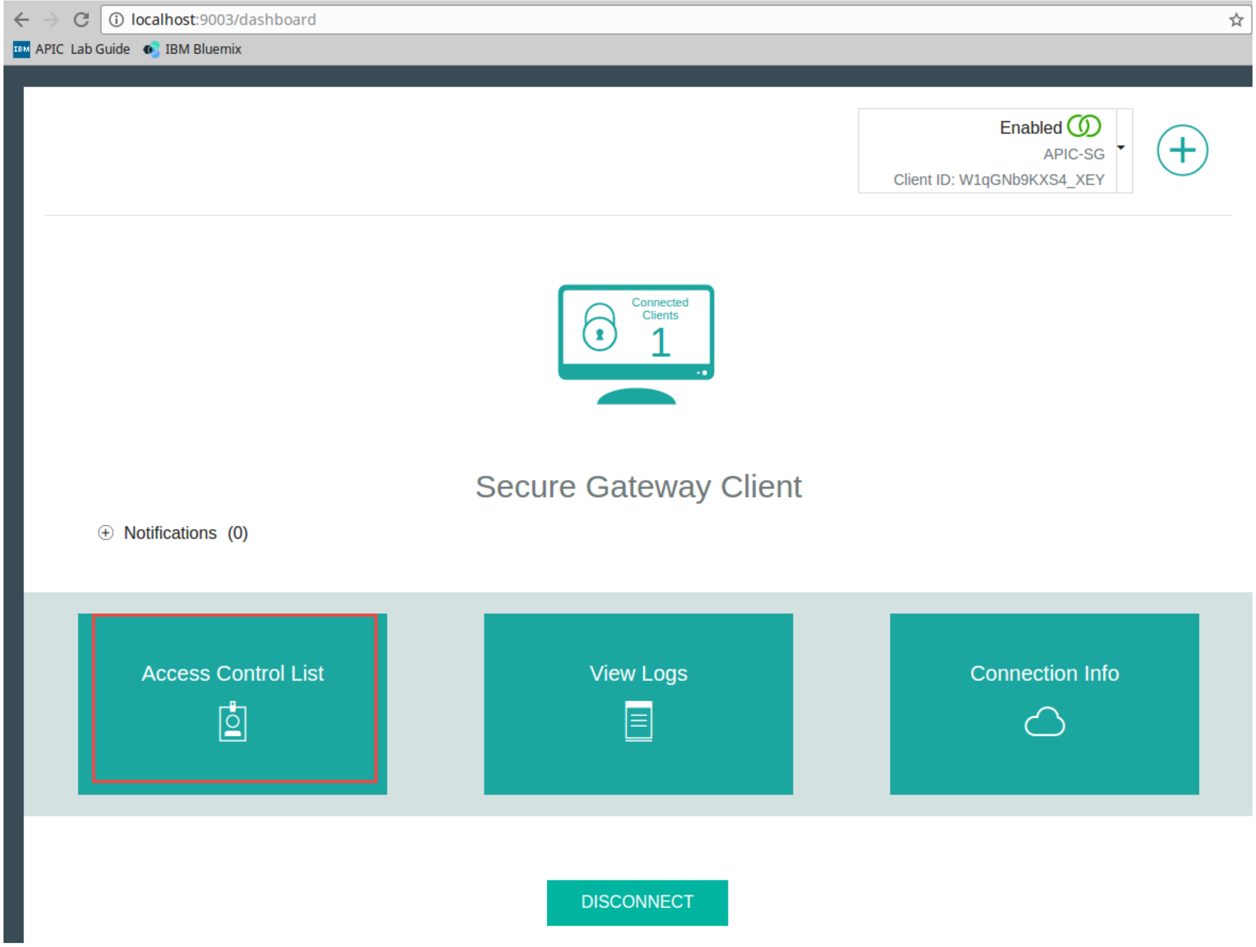
Enter a gateway ID and security token (optional) to connect

W1qGNb9KXS4_prod_ng

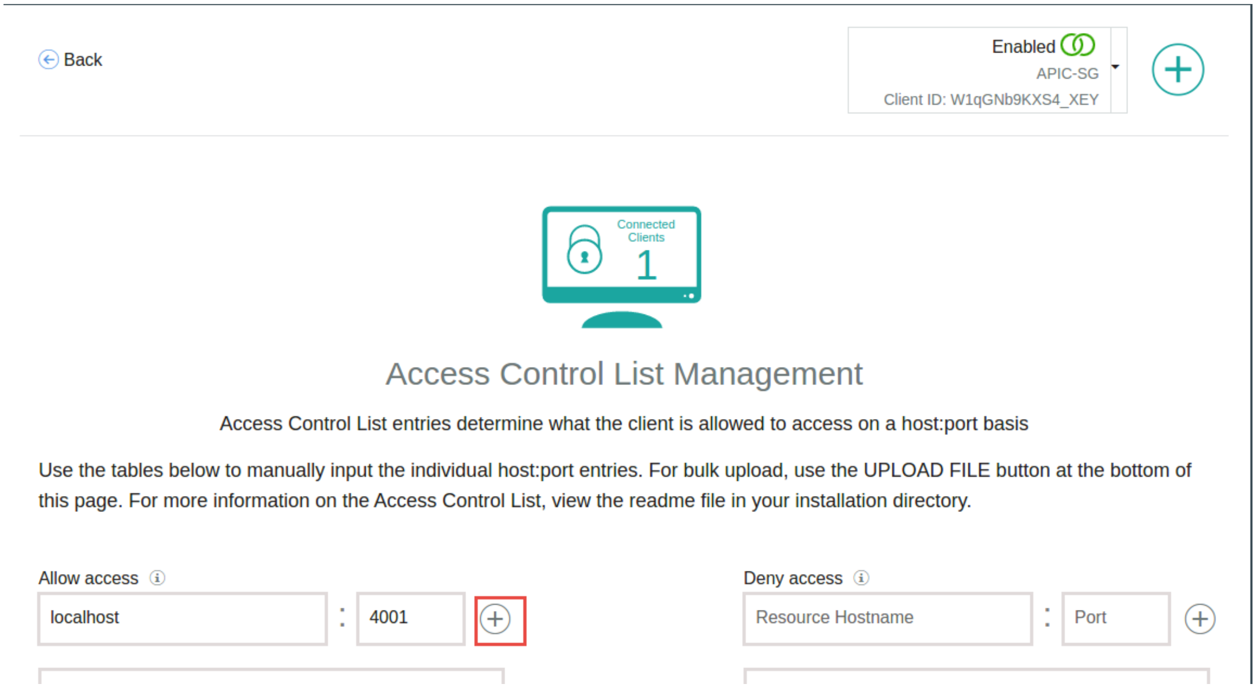
xODM1MDYxLCJleHAiOjE1MjMzNzEwNjF9.qv1uZLp7sUl4UlH7ZO7RSVJkOYs5TLFlHRZs_rtUbkU

ID not recognized or Security Token required

On Dashboard click ACL



Add localhost:4001

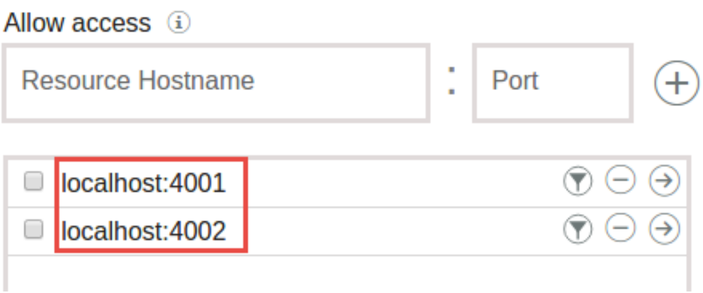


Access Control List Management

Access Control List entries determine what the client is allowed to access on a host:port basis

Use the tables below to manually input the individual host:port entries. For bulk upload, use the UPLOAD FILE button at the bottom of this page. For more information on the Access Control List, view the readme file in your installation directory.

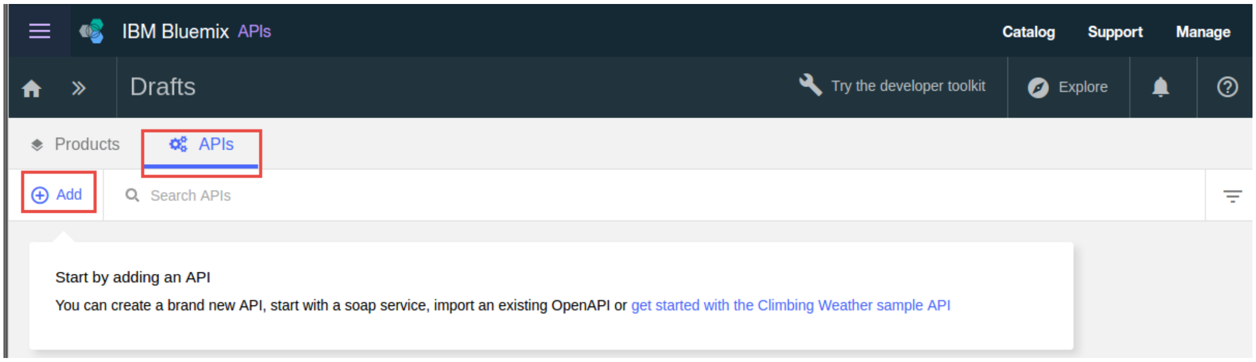
Add localhost:4002. See results:



□ localhost:4001

□ localhost:4002

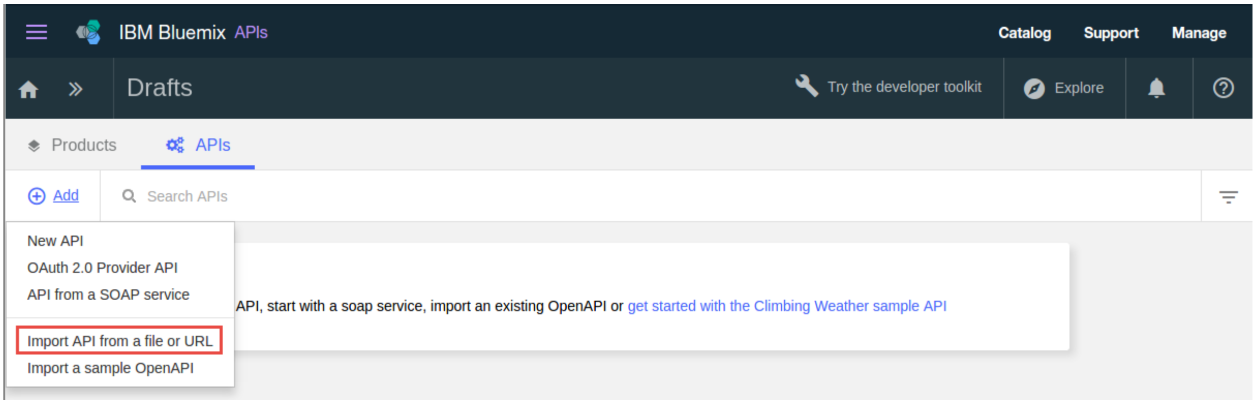
After the client is successfully configured, we can import the on-premise API into API Connect. First we go into the Drafts mode and select the APIs tab.



Start by adding an API

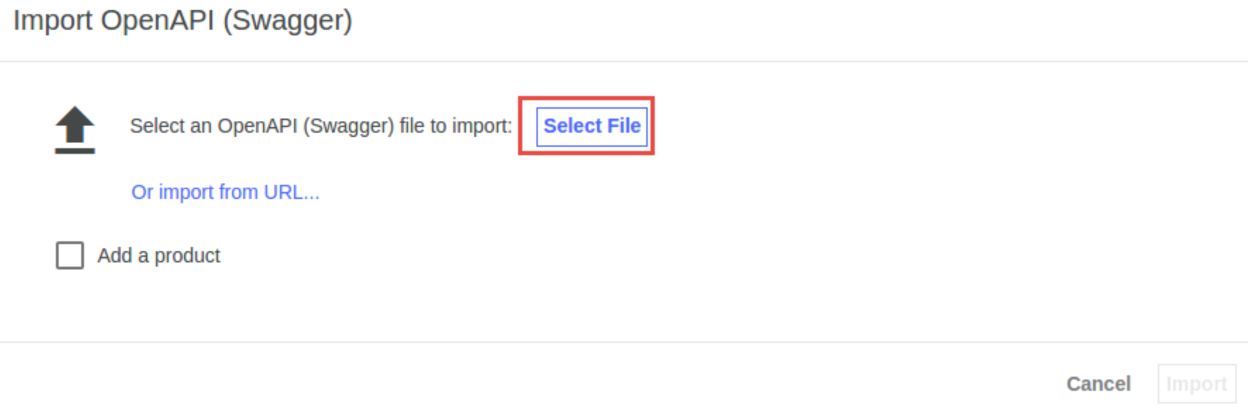
You can create a brand new API, start with a soap service, import an existing OpenAPI or get started with the Climbing Weather sample API

Then click the “Add” link and select “Import an existing OpenAPI” from the drop down.

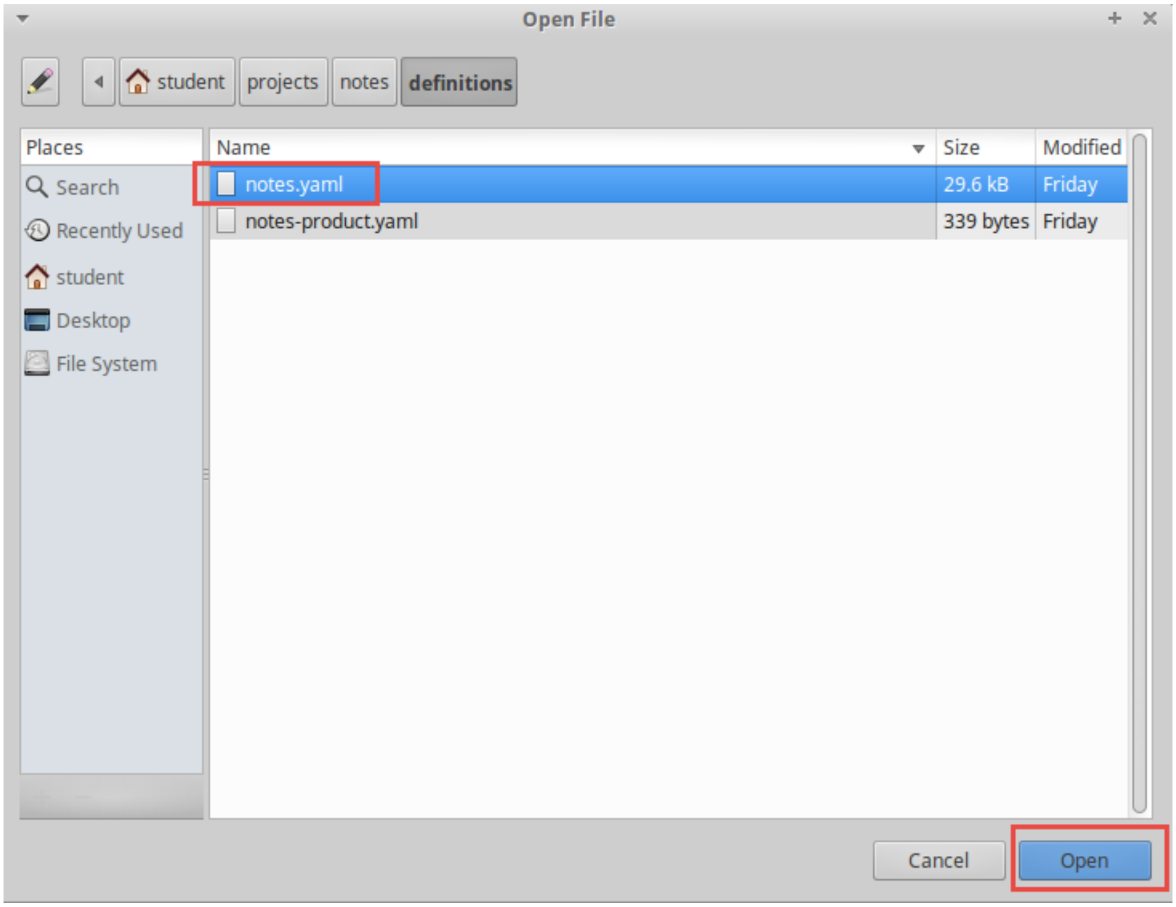


API, start with a soap service, import an existing OpenAPI or get started with the Climbing Weather sample API

The file we want to import is located in the $(project_workspace)/mysql-project/definitions directory from the API we created in the first part of the blog. There will be two files within that directory: the API definition file and the product definition file. Select the API definition file.

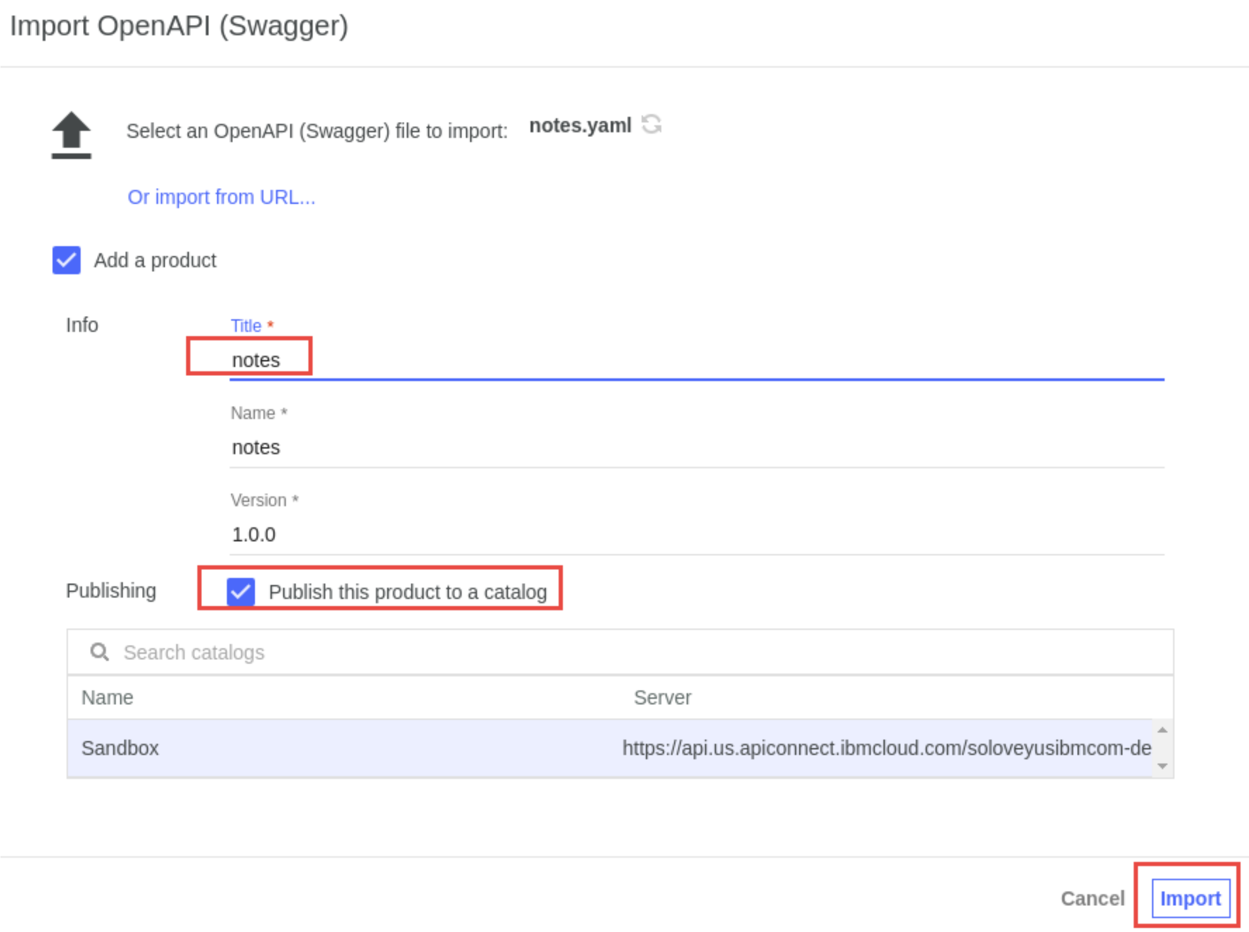


Select



Name	Size	Modified
notes.yaml	29.6 kB	Friday
notes-product.yaml	339 bytes	Friday

After selecting the file, make sure to check the box for “Add a product” to create a product for the imported API definition.



Import OpenAPI (Swagger)

Select an OpenAPI (Swagger) file to import: **notes.yaml**

Or import from URL...

☑ Add a product

Info

Title * notes

Name * notes

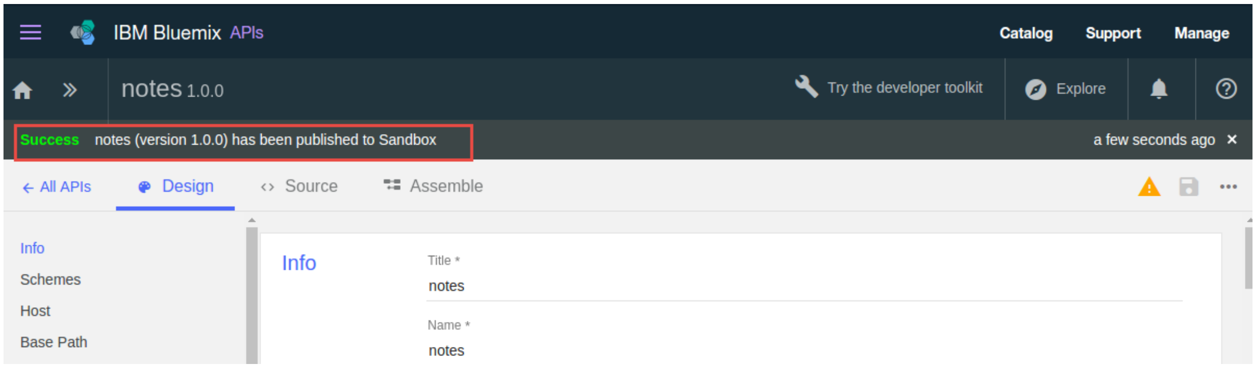
Version * 1.0.0

Publishing ☑ Publish this product to a catalog

Search catalogs

Name	Server
Sandbox	https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-de

Now we have the API and product imported into API Connect.



Success notes (version 1.0.0) has been published to Sandbox

a few seconds ago ×

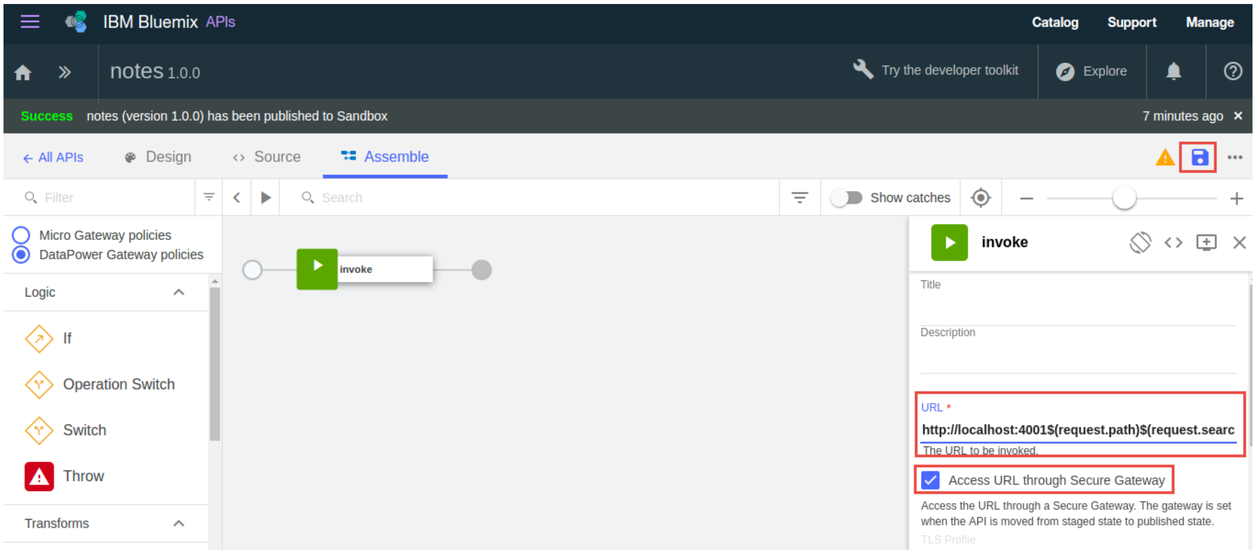
Info

Title * notes

Name * notes

Next we will make this API a Secure Gateway API. Within the Drafts mode, navigate to the Assemble tab of the API and select the invoke policy. In the right handle menu, check the box to “Access URL through Secure Gateway” and change the URL to the hostname and port of the machine hosting the on-premise API.

For our example, the Secure Gateway client is running on the same machine as the on-premises API, so the URL is http://localhost:4001$(request.path)$(request.search)



Success notes (version 1.0.0) has been published to Sandbox 7 minutes ago

invoke

Title

Description

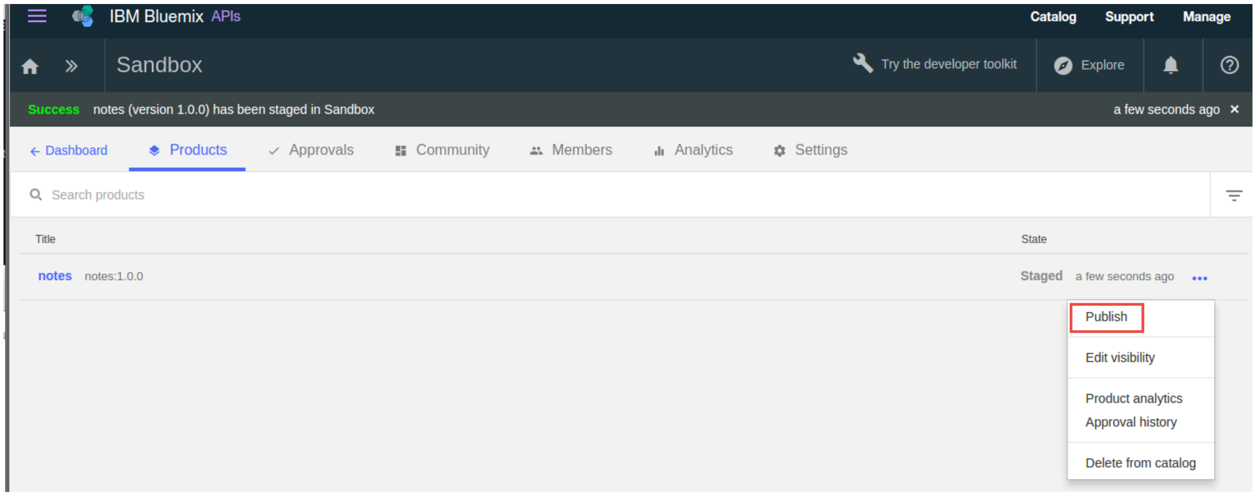
URL *
http://localhost:4001$(request.path)$(request.searc
The URL to be invoked.

☑ Access URL through Secure Gateway

Access the URL through a Secure Gateway. The gateway is set when the API is moved from staged state to published state.

TLS Profile

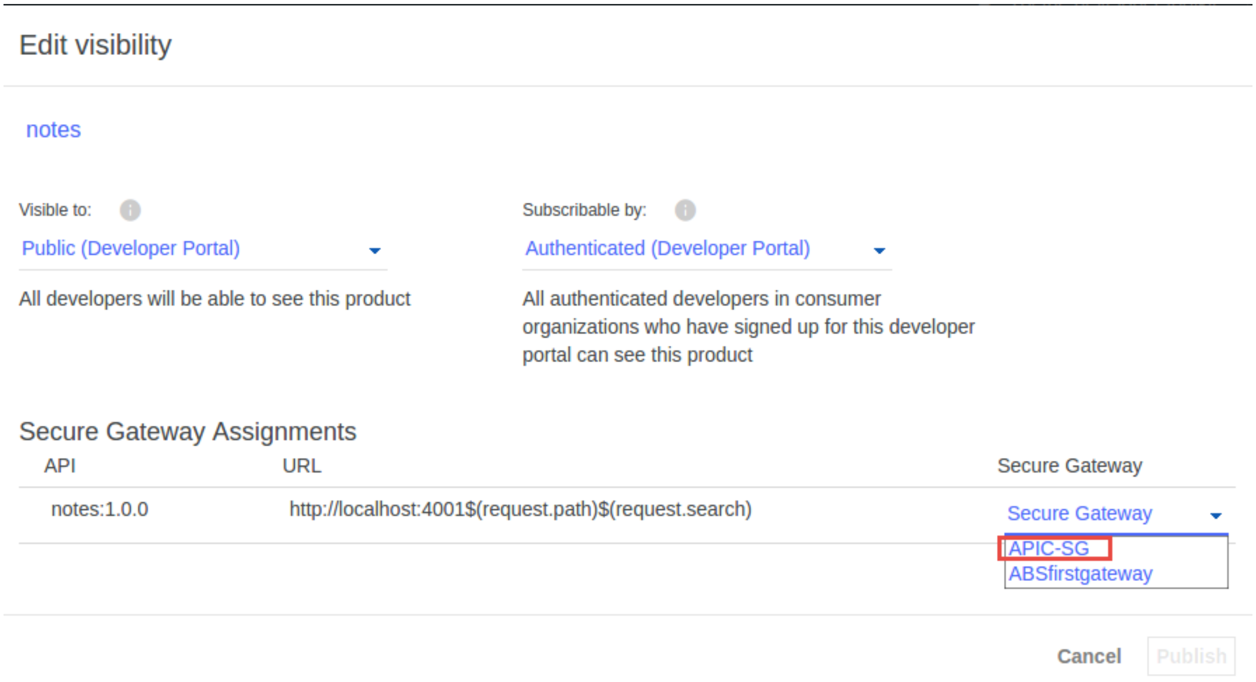
After saving the changes, you can stage the product to a catalog and publish the product.



Success notes (version 1.0.0) has been staged in Sandbox a few seconds ago

Title	State
notes notes:1.0.0	Staged a few seconds ago

When you publish the product, a new section will be present to configure which gateway to use for the product.



Edit visibility

notes

Visible to: Public (Developer Portal)

All developers will be able to see this product

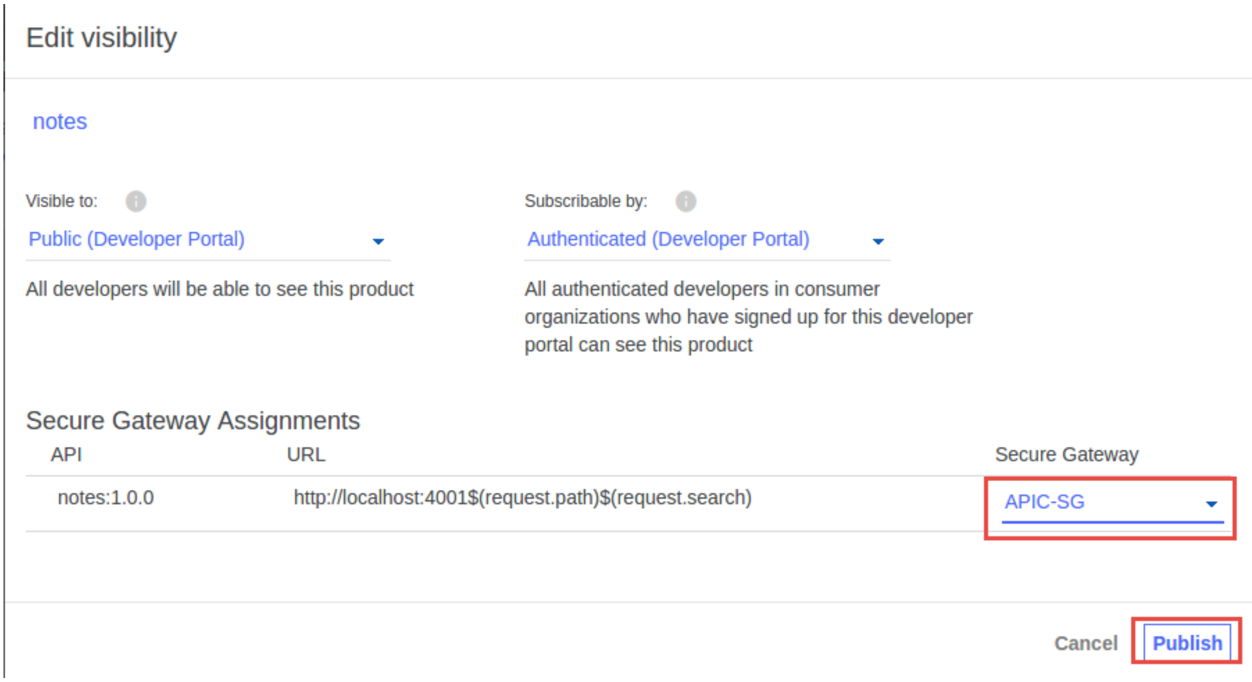
Subscribable by: Authenticated (Developer Portal)

All authenticated developers in consumer organizations who have signed up for this developer portal can see this product

Secure Gateway Assignments

API	URL	Secure Gateway
notes:1.0.0	http://localhost:4001$(request.path)$(request.search)	Secure Gateway

Select the gateway created earlier and publish the API.



Edit visibility

notes

Visible to: Public (Developer Portal)

All developers will be able to see this product

Subscribable by: Authenticated (Developer Portal)

All authenticated developers in consumer organizations who have signed up for this developer portal can see this product

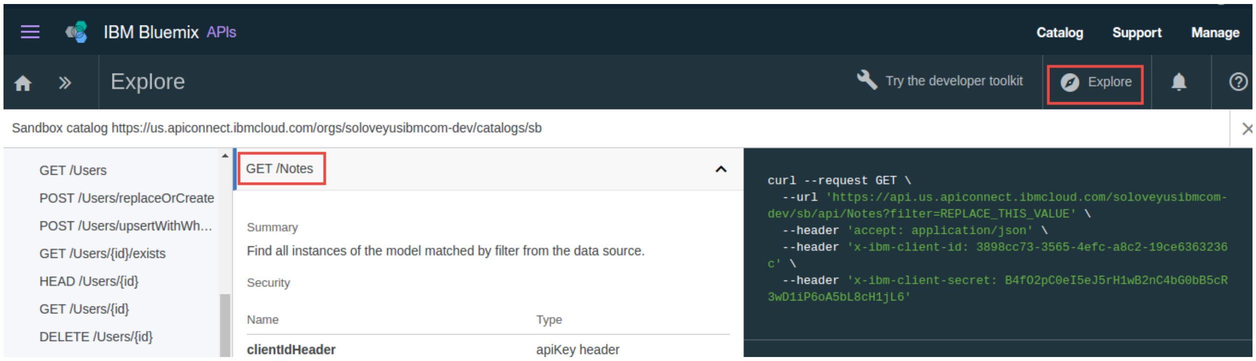
Secure Gateway Assignments

API	URL	Secure Gateway
notes:1.0.0	http://localhost:4001$(request.path)$(request.search)	APIC-SG

Testing API from APIC Management server

All that is left to do is test the newly published Secure Gateway API. You can use curl to make a request to the API Connect endpoint or use the built in test tool available within API Connect.

Select notes API and click Explore then select GET /Notes operation on the left



Sandbox catalog https://us.apiconnect.ibmcloud.com/orgs/soloveyusibmcom-dev/catalogs/sb

Summary

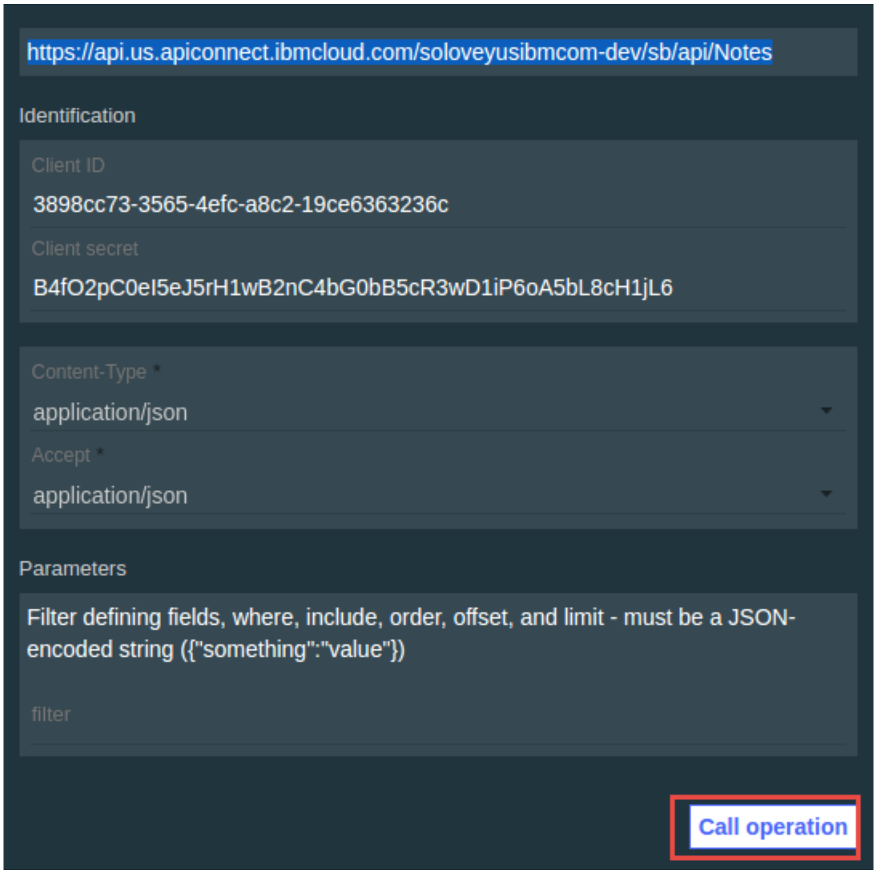
Find all instances of the model matched by filter from the data source.

Security

Name	Type
clientIdHeader	apiKey header

```
curl --request GET \
  --url 'https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-
dev/sb/api/Notes?filter=REPLACE_THIS_VALUE' \
  --header 'accept: application/json' \
  --header 'x-ibm-client-id: 3898cc73-3565-4efc-a8c2-19ce6363236
c' \
  --header 'x-ibm-client-secret: B4fO2pC0eI5eJ5rH1wB2nC4bG0bB5cR
3wD1iP6oA5bL8cH1jL6'
```

Scroll down on the right section and click “Call operation”



https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-dev/sb/api/Notes

Identification

Client ID
3898cc73-3565-4efc-a8c2-19ce6363236c

Client secret
B4fO2pC0eI5eJ5rH1wB2nC4bG0bB5cR3wD1iP6oA5bL8cH1jL6

Content-Type *
application/json

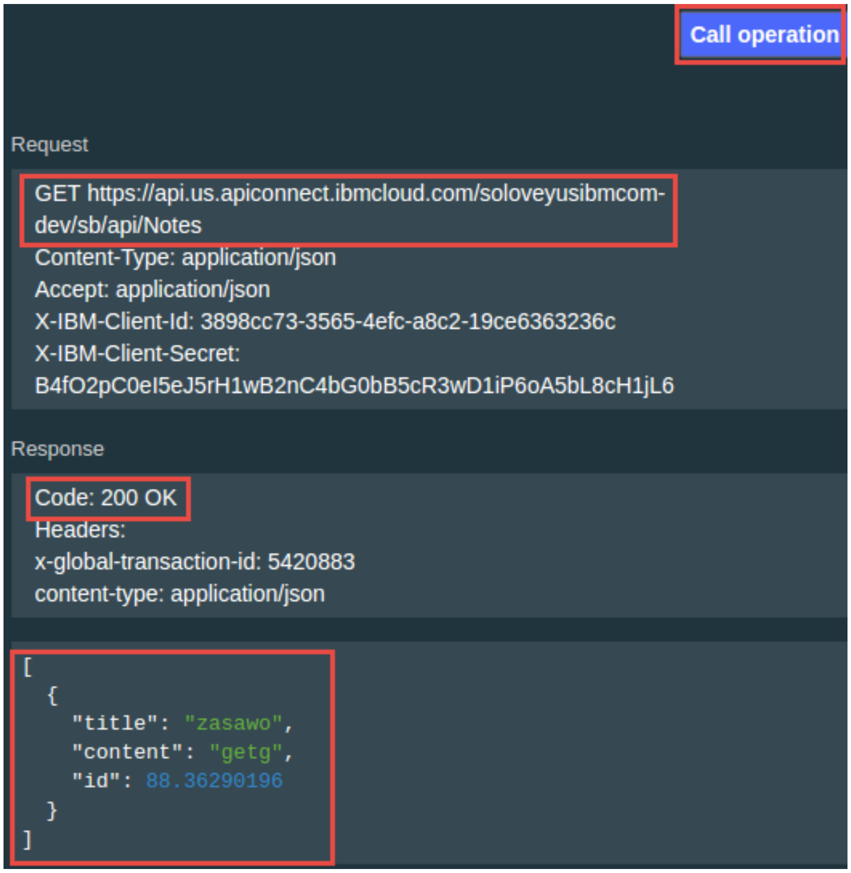
Accept *
application/json

Parameters

Filter defining fields, where, include, order, offset, and limit - must be a JSON-encoded string ({"something":"value"})

filter

Review Request URL and result (same as running localhost on VM):



Request

```
GET https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-
dev/sb/api/Notes
Content-Type: application/json
Accept: application/json
X-IBM-Client-Id: 3898cc73-3565-4efc-a8c2-19ce6363236c
X-IBM-Client-Secret:
B4fO2pC0eI5eJ5rH1wB2nC4bG0bB5cR3wD1iP6oA5bL8cH1jL6
```

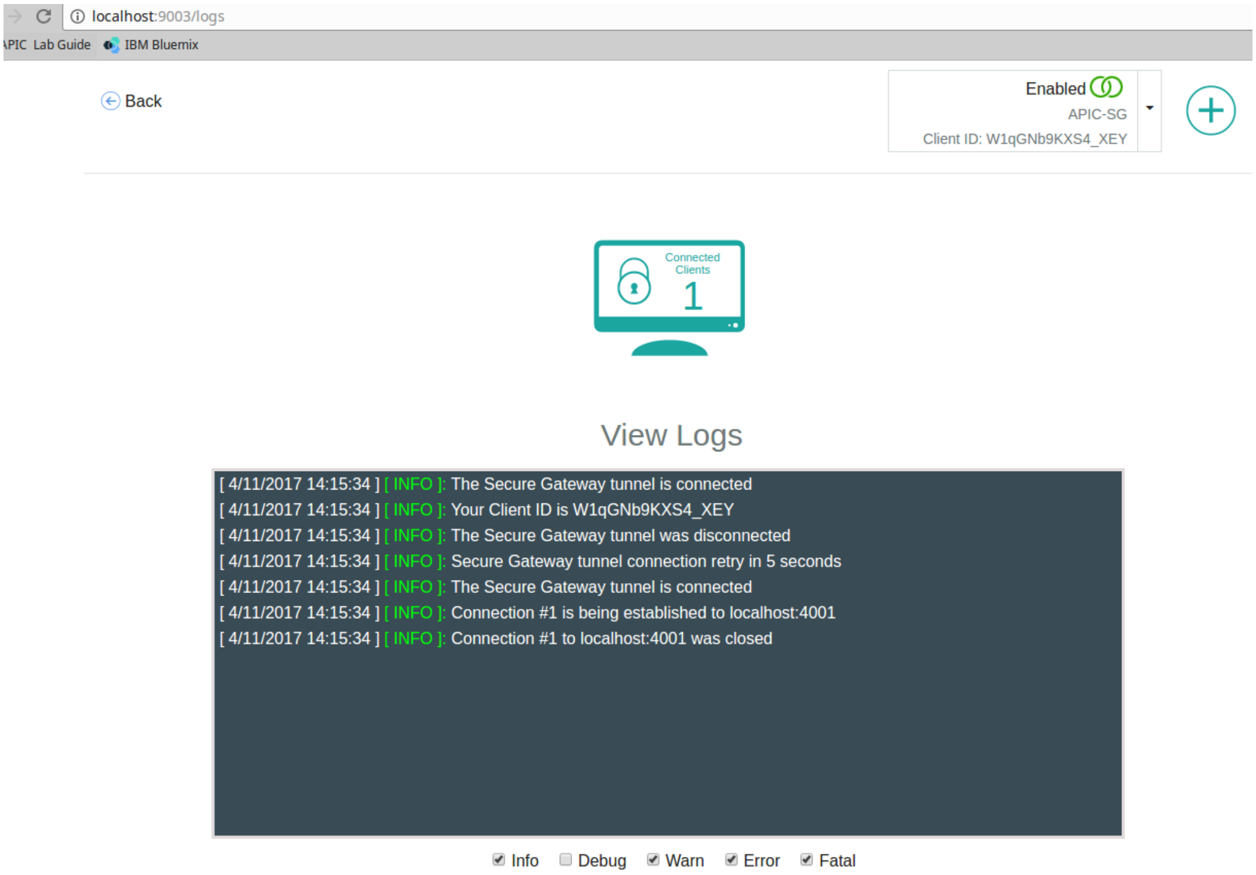
Response

```
Code: 200 OK
Headers:
x-global-transaction-id: 5420883
content-type: application/json
```

```
[
  {
    "title": "zasawo",
    "content": "getg",
    "id": 88.36290196
  }
]
```

Now you have successfully published your first Secure Gateway enabled API. With the new Secure Gateway Integration within API Connect, you will be able to quickly and securely create APIs to access on-premises resources.

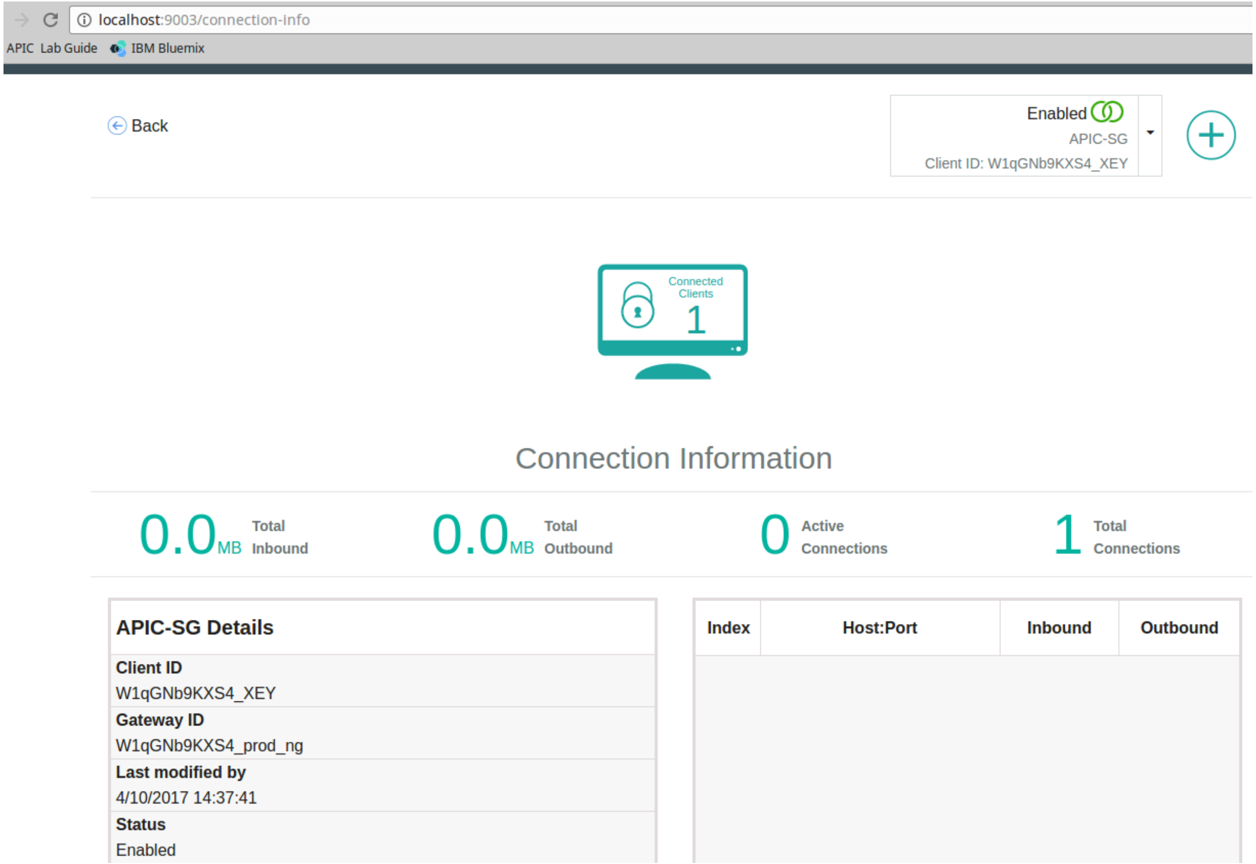
Review SG client logs



```
[ 4/11/2017 14:15:34 ] [ INFO ]: The Secure Gateway tunnel is connected
[ 4/11/2017 14:15:34 ] [ INFO ]: Your Client ID is W1qGNb9KXS4_XEY
[ 4/11/2017 14:15:34 ] [ INFO ]: The Secure Gateway tunnel was disconnected
[ 4/11/2017 14:15:34 ] [ INFO ]: Secure Gateway tunnel connection retry in 5 seconds
[ 4/11/2017 14:15:34 ] [ INFO ]: The Secure Gateway tunnel is connected
[ 4/11/2017 14:15:34 ] [ INFO ]: Connection #1 is being established to localhost:4001
[ 4/11/2017 14:15:34 ] [ INFO ]: Connection #1 to localhost:4001 was closed
```

☑ Info ☐ Debug ☑ Warn ☑ Error ☑ Fatal

Client connection info



Enabled
APIC-SG
Client ID: W1qGNb9KXS4_XEY

Connection Information

0.0MB Total Inbound

0.0MB Total Outbound

0 Active Connections

1 Total Connections

APIC-SG Details

Client ID
W1qGNb9KXS4_XEY

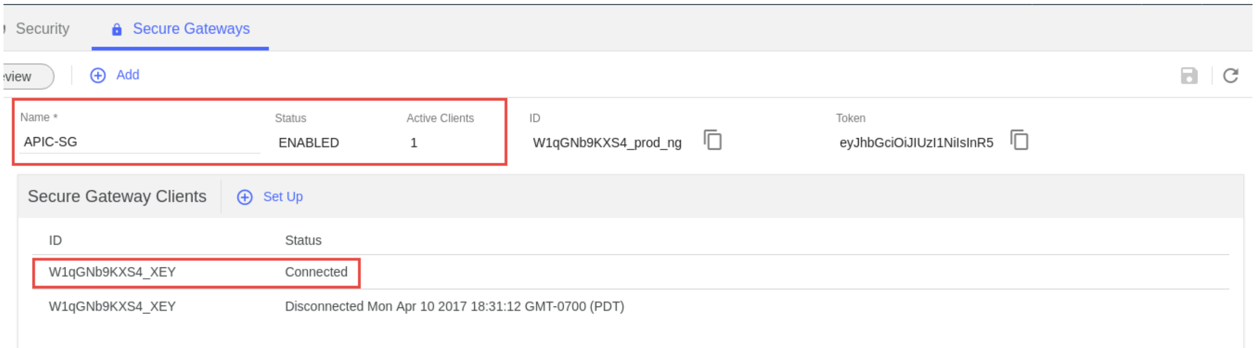
Gateway ID
W1qGNb9KXS4_prod_ng

Last modified by
4/10/2017 14:37:41

Status
Enabled

Index	Host:Port	Inbound	Outbound

Review SG server in APIC Admin Secure Gateways tab



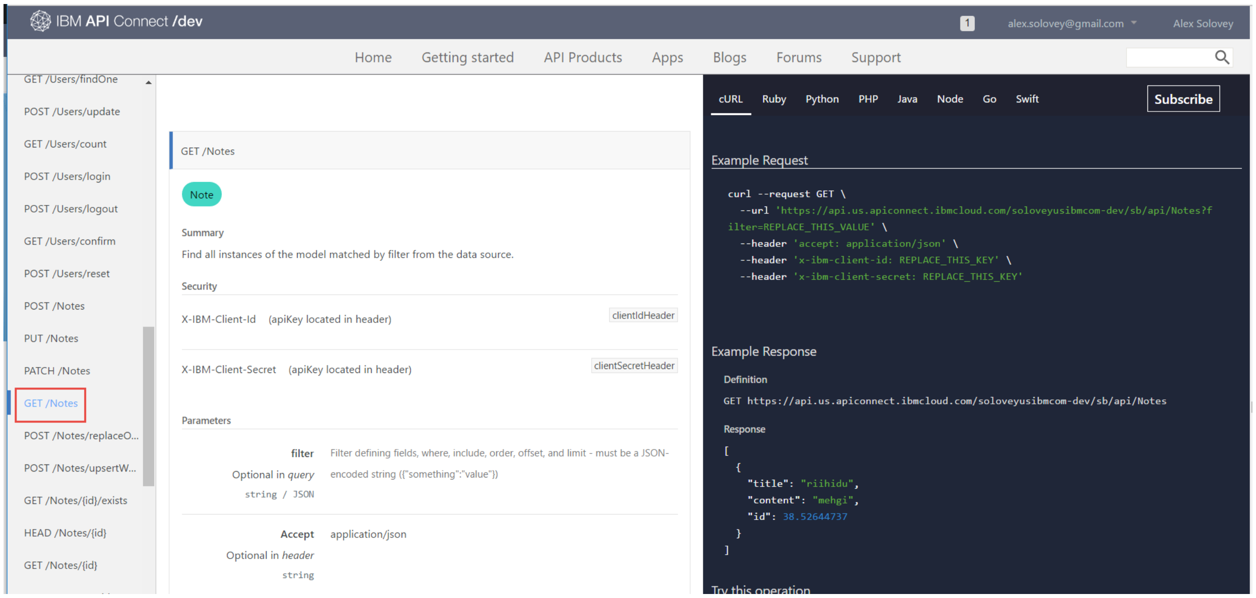
Name *	Status	Active Clients	ID	Token
APIC-SG	ENABLED	1	W1qGNb9KXS4_prod_ng	eyJhbGciOiJIUzI1NiIsInR5

Secure Gateway Clients Set Up

ID	Status
W1qGNb9KXS4_XEY	Connected
W1qGNb9KXS4_XEY	Disconnected Mon Apr 10 2017 18:31:12 GMT-0700 (PDT)

Testing from APIC Developer Portal

Sign into Dev Portal, create an app and subscribe to notes API. Select GET /Notes



GET /Notes

Note

Summary

Find all instances of the model matched by filter from the data source.

Security

X-IBM-Client-Id (apiKey located in header) clientIdHeader

X-IBM-Client-Secret (apiKey located in header) clientSecretHeader

Parameters

filter — Optional in *query* — string / JSON — Filter defining fields, where, include, order, offset, and limit - must be a JSON-encoded string ({"something":"value"})

Accept — Optional in *header* — string — application/json

Example Request

```
curl --request GET \
  --url 'https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-dev/sb/api/Notes?f
ilter=REPLACE_THIS_VALUE' \
  --header 'accept: application/json' \
  --header 'x-ibm-client-id: REPLACE_THIS_KEY' \
  --header 'x-ibm-client-secret: REPLACE_THIS_KEY'
```

Example Response

Definition

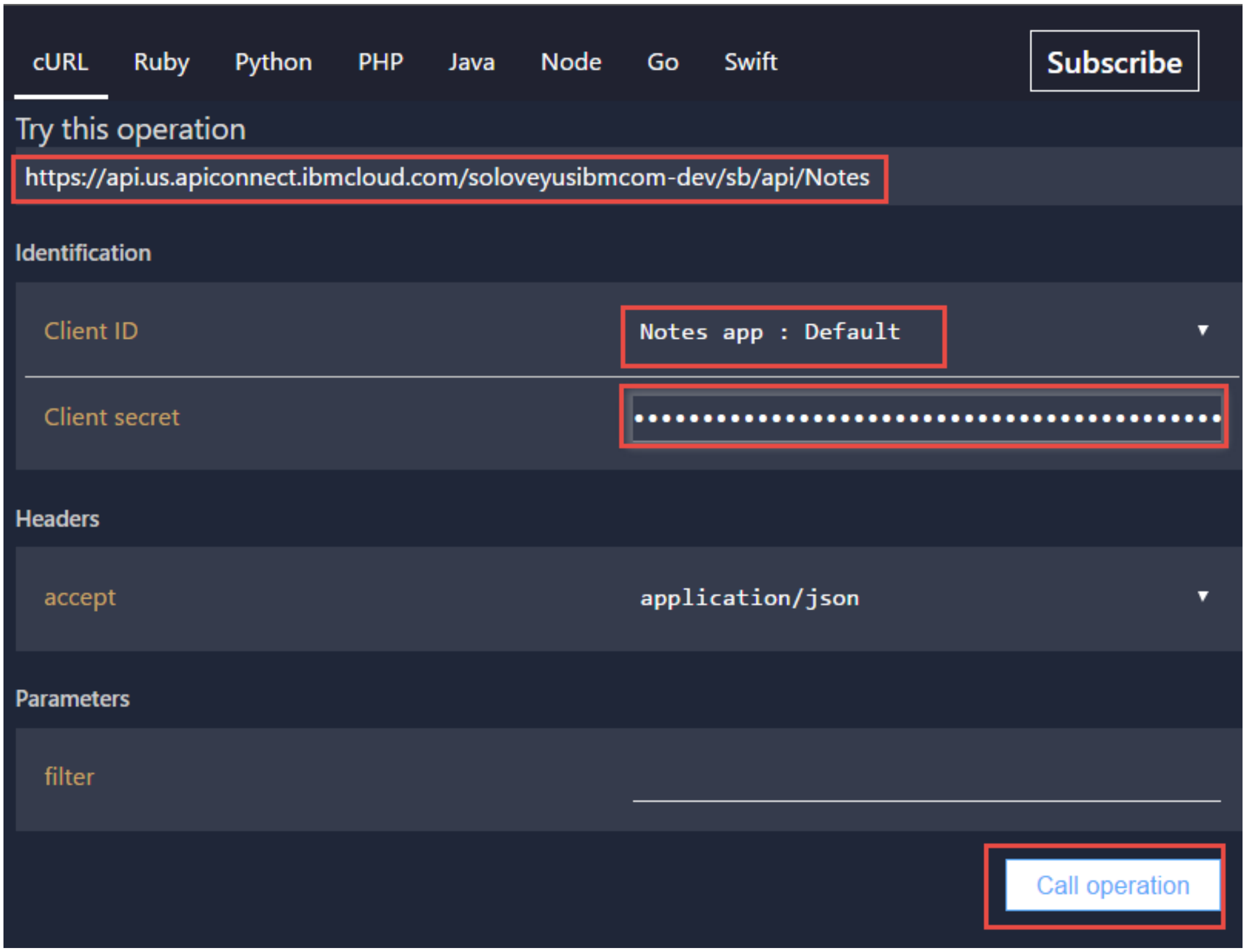
```
GET https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-dev/sb/api/Notes
```

Response

```
[
  {
    "title": "riihidu",
    "content": "mehgi",
    "id": 38.52644737
  }
]
```

Try this operation

Copy and paste Client secret and select Notes App for Client Id. Call Operation.



Try this operation

https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-dev/sb/api/Notes

Identification

Client ID: Notes app : Default

Client secret: ••

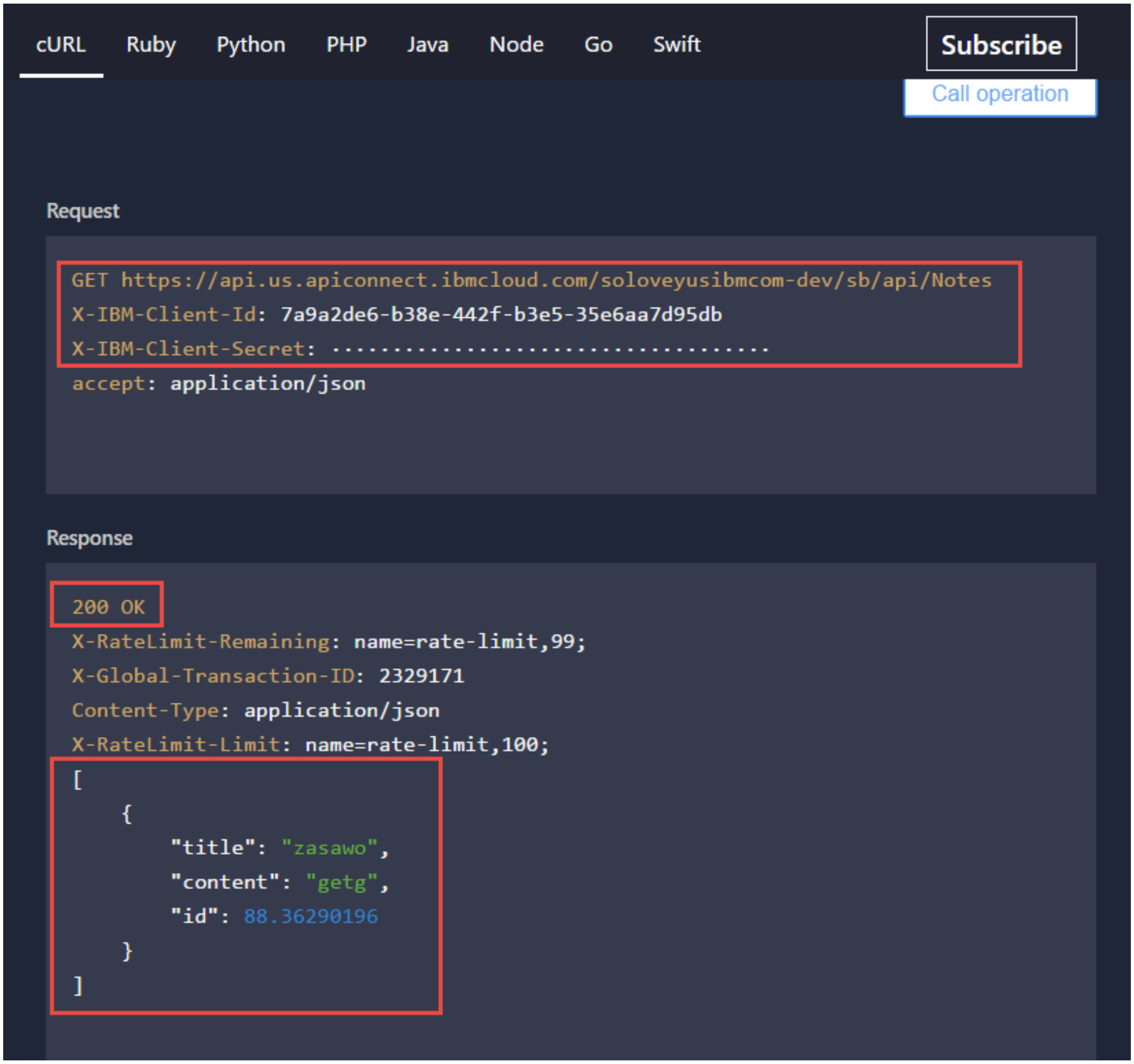
Headers

accept: application/json

Parameters

filter

Get the same result as before



Request

```
GET https://api.us.apiconnect.ibmcloud.com/soloveyusibmcom-dev/sb/api/Notes
X-IBM-Client-Id: 7a9a2de6-b38e-442f-b3e5-35e6aa7d95db
X-IBM-Client-Secret: ····································
accept: application/json
```

Response

```
200 OK
X-RateLimit-Remaining: name=rate-limit,99;
X-Global-Transaction-ID: 2329171
Content-Type: application/json
X-RateLimit-Limit: name=rate-limit,100;
[
    {
        "title": "zasawo",
        "content": "getg",
        "id": 88.36290196
    }
]
```