

In this lab, you will explore the how to define GraphQL APIs that proxy to a backend GraphQL server. GraphQL is a query language for APIs that gives an application client greater control over what data it retrieves in an API request when compared with a REST API request.

API Connect GraphQL support also allows rate limiting and other controls using API Gateway.

In this tutorial, you will explore the following key capabilities:

- Create a GraphQL Proxy API
- Learn about GraphQL introspection and Schema Editor
- Test APIs by using the built-in GraphQL editor
- Review API Assembly Policies for the cost end point

APIC Dev Jam Series

The APIC Dev Jam Series is a hands-on workshop with lab exercises that walk you through designing, publishing, and securing APIs. This workshop is for API developers, architects, and line of business people who want to create a successful API strategy. There are 8 labs and each is 30 minutes long. Make sure you choose enough time in your reservation to get through all the labs!

[NOTE:]**[This demo environment contains a full API Connect installation in Cloud Pak for Integration. The login information to the APIC cluster will be sent in a separate email when you reserve the instance. Use Google Chrome, Firefox or Microsoft Edge to access the cluster using the credentials supplied. Make sure you login using API Manager User Registry not Common Services registry.]**

Prerequisites: None

Create a GraphQL Proxy API

1. Select Develop API and click Add->API

The screenshot shows the 'Develop' page in IBM Cloud Pak for Integration. At the top right, there is an 'Add' button. A dropdown menu is open below it, with 'API' selected and highlighted by a red box. Below the dropdown is a search bar and a table of API services. The table has columns for Title, Version, Type, and Last modified. The services listed are:

| Title | Version | Type | Last modified |
|---------------------|---------|------------|---------------|
| Accessories Product | 1.0.0 | Product | 5 days ago |
| financing | 1.0.0 | API (REST) | 5 days ago |
| inventory | 1.0.0 | API (REST) | 5 days ago |
| inventory | 2.0.0 | API (REST) | 5 days ago |
| Inventory Product | 1.0.0 | Product | 5 days ago |
| logistics | 2.0.0 | API (REST) | 5 days ago |

At the bottom of the table, there are controls for 'Items per page' (set to 10) and '1-6 of 6 items'. On the right side, there are pagination controls showing '1 of 1 pages'.

2. Enter the following values and click Next. You To create a GraphQL proxy definition, select From Existing GraphQL service. When you point to an existing GraphQL server, API Connect introspects the service and automatically creates a GraphQL proxy service.

The screenshot shows the 'Select API Type' page in IBM Cloud Pak for Integration. The page is titled 'Develop / Select API Type'. Under the 'Create' section, there are several options:

- From target service: Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service: Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy): Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy): Create a REST proxy based upon a WSDL described target service
- From existing GraphQL service (GraphQL proxy): Create a GraphQL proxy based on a GraphQL service
- New OpenAPI: Compose a new REST proxy by defining paths and operations

To the right of the options is a diagram illustrating the flow of data. It shows an 'App' on the left, a 'GraphQL API proxy' in the center, and a 'GraphQL endpoint' on the right. The 'GraphQL API proxy' is connected to the 'App' and the 'GraphQL endpoint'. Below the 'GraphQL API proxy' is a 'GraphQL schema' icon, and below the 'GraphQL endpoint' is another 'GraphQL schema' icon. Arrows indicate the flow of data between the App, the proxy, and the endpoint.

3. Enter the following values and click Next. You can use default values for the rest.

Title: accounts

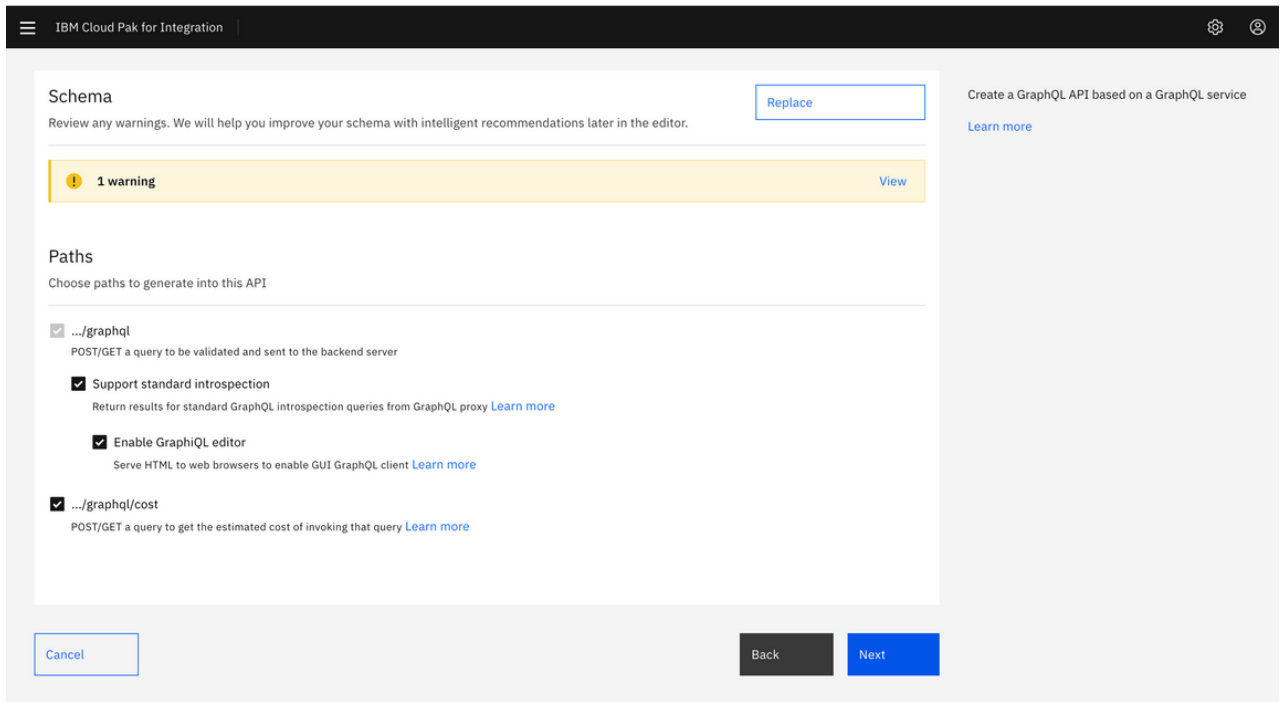
GraphQL server URL: <https://graphql-test-server.us-east.cf.appdomain.cloud/accounts/graphql>

The screenshot shows the IBM Cloud Pak for Integration console. The top navigation bar includes a hamburger menu, the text 'IBM Cloud Pak for Integration', and a user profile icon. Below the navigation bar, the breadcrumb 'Develop / Select API Type /' is visible. The main heading is 'Create API from existing GraphQL service (GraphQL proxy)'. The form is divided into two main sections:

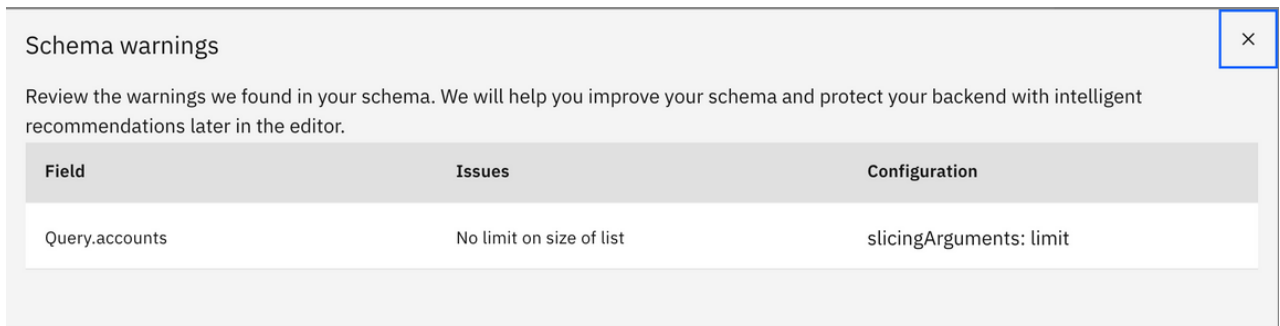
- Info:** A section titled 'Info' with the instruction 'Enter details of this API'. It contains five input fields:
 - Title: 'accounts'
 - Name: 'accounts'
 - Version: '1.0.0'
 - Base path (optional): '/accounts'
 - Description (optional): (empty)
- GraphQL Service:** A section titled 'GraphQL Service' with the instruction 'Enter the URL and schema name for the GraphQL service you would like to proxy'. It contains two input fields:
 - GraphQL server URL: 'https://graphql-test-server.us-east.cf.appdomain.cloud/accounts/graphql'
 - Schema name: 'graphql-test-server-us-east-cf-appdomain-cloud'

At the bottom of the form, there are two buttons: 'Cancel' and 'Next'. On the right side of the form, there is a sidebar with the text 'Create a GraphQL API based on a GraphQL service' and a 'Learn more' link.

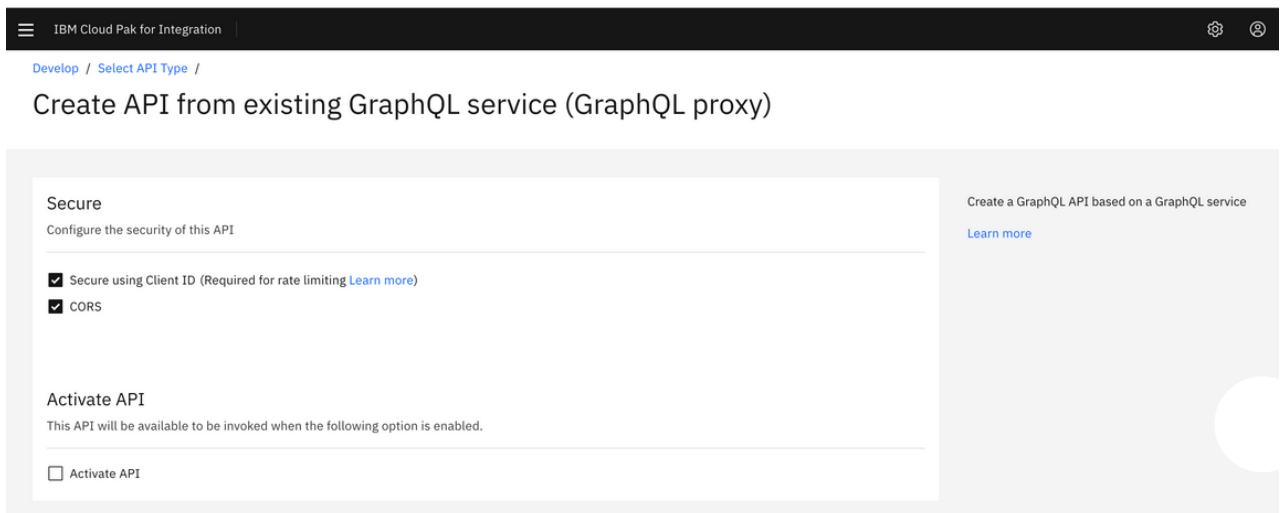
4. The schema validator reports warning and errors if found. There is one warning in the accounts schema imported from the server. You can review the schema warning details and also select the end points interested for this proxy. Select all the available end points.

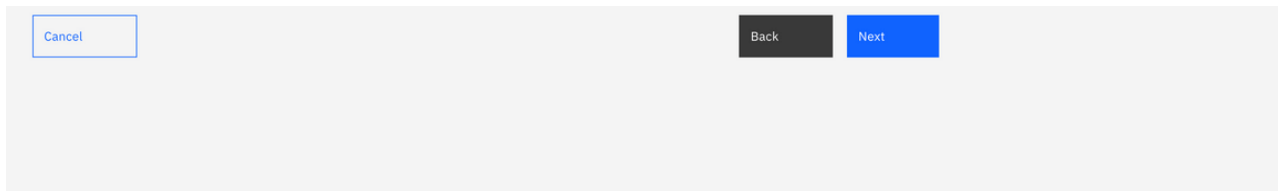


5. A review of the schema shows that no limit is set on the size of the list. You will fix this warning in later steps. Click Next to continue.

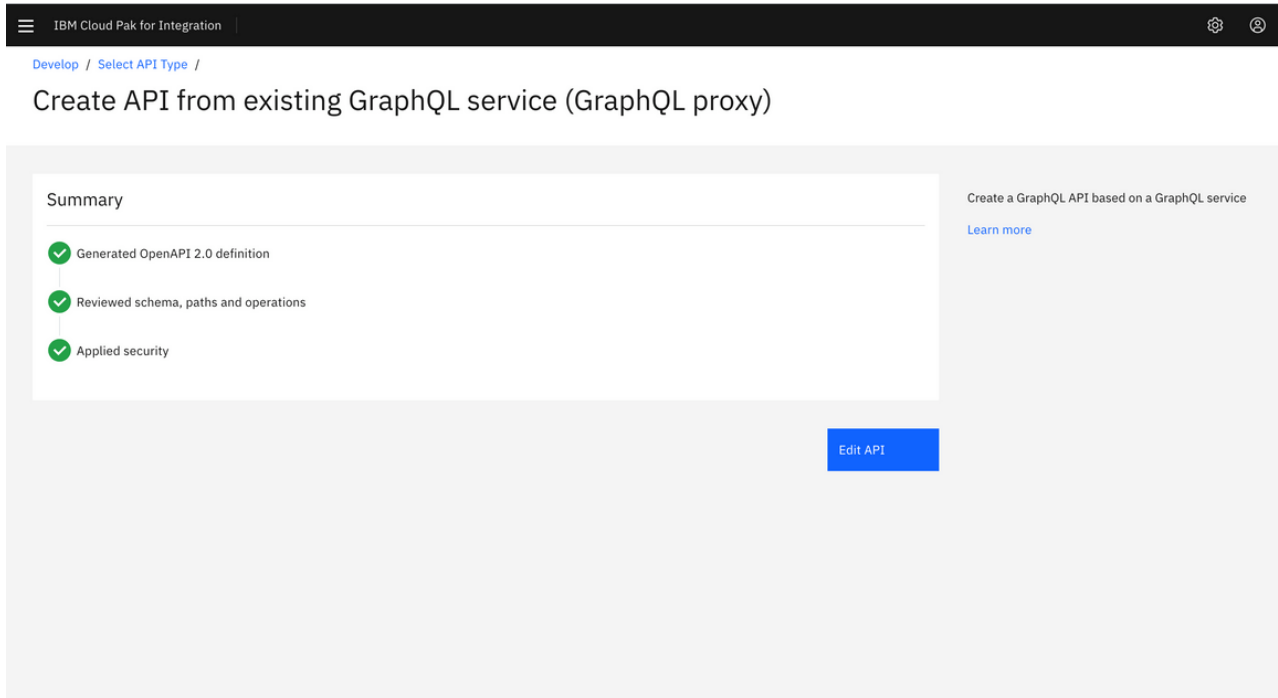


6. Keep both Secure Client ID and CORS checked and click Next

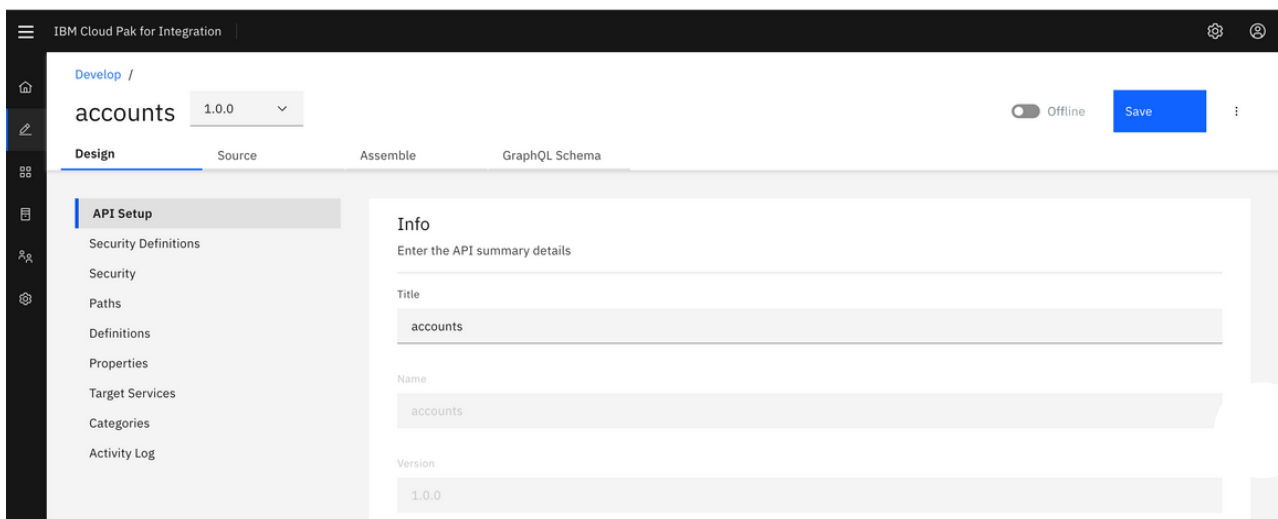


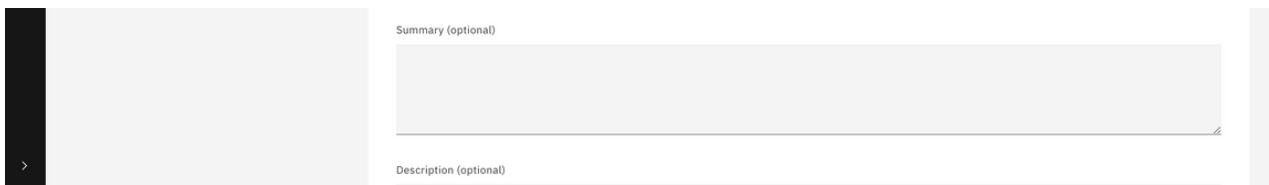


7. API Connect successfully created a GraphQL proxy API. You can edit the API to view more details on the proxy API just created. Click EDIT API to continue.

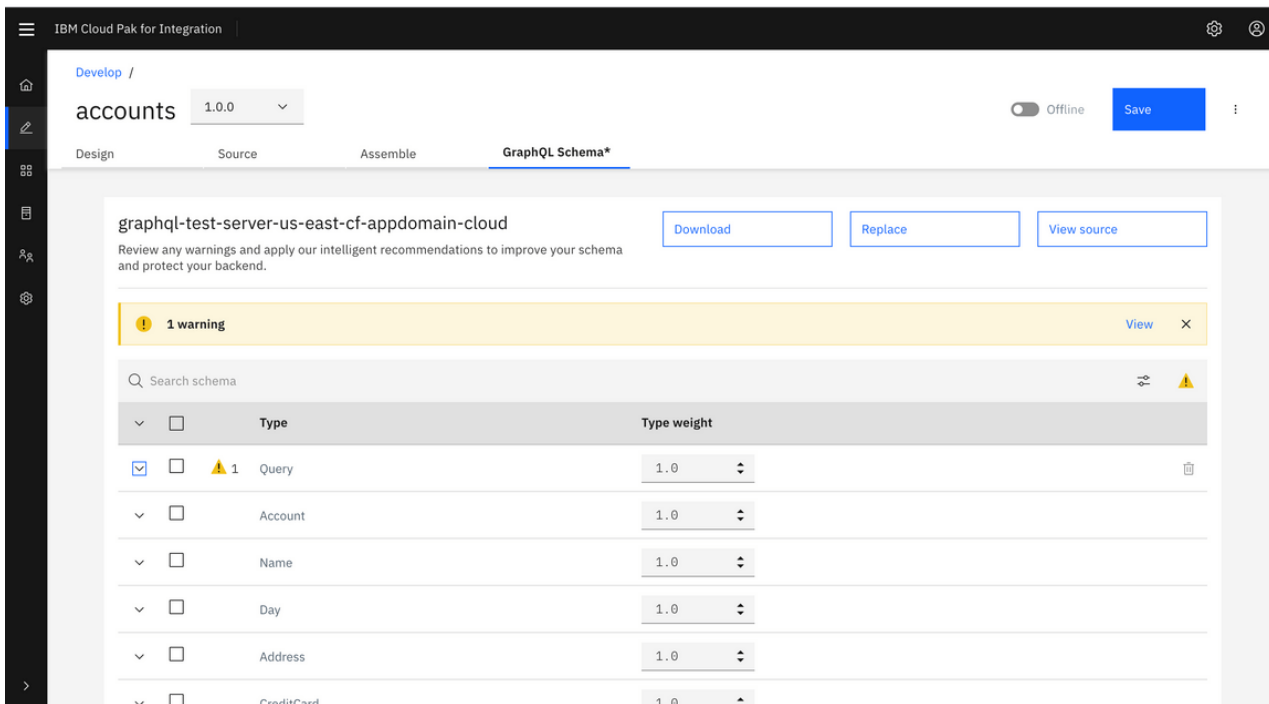


8. GraphQL APIs are structured the same as REST APIs with some additional options. Just like REST APIs, you will be able to add security definitions, properties and Activity logs from the left menu bar. You can also view the Source and Assemble tabs similar to REST APIs. You will notice a new tab for GraphQL APIs. Click on GraphQL Schema tab to view the schema.

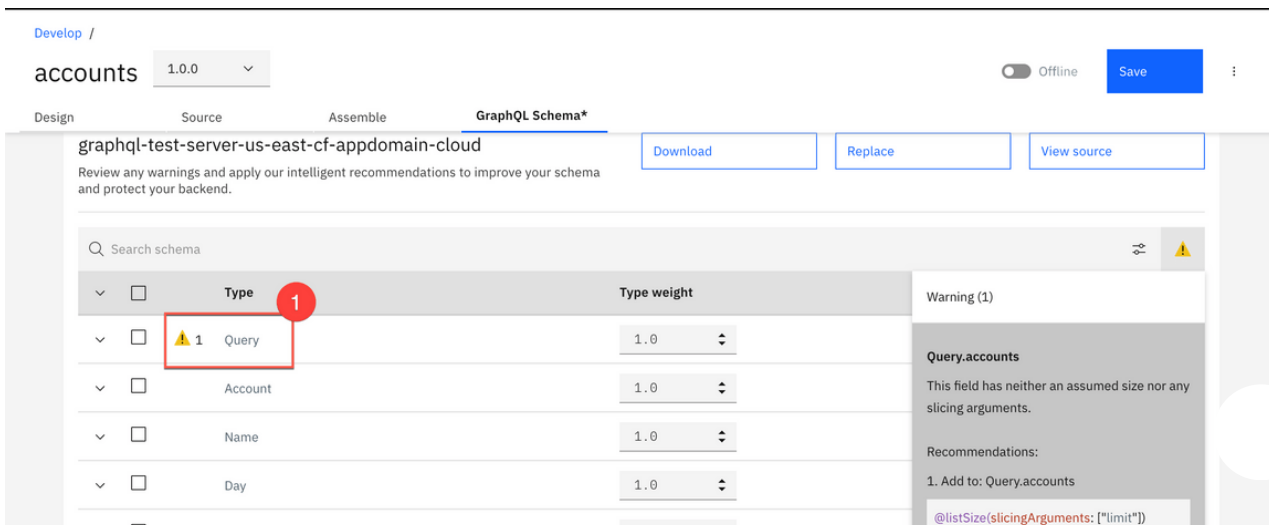




9. GraphQL Schema editor displays Type and Weight information. The weighting factor is used when calculating the type cost for a request to the GraphQL API. For example, a field that requires extensive CPU or memory use on the server to retrieve its value would be given a higher cost.

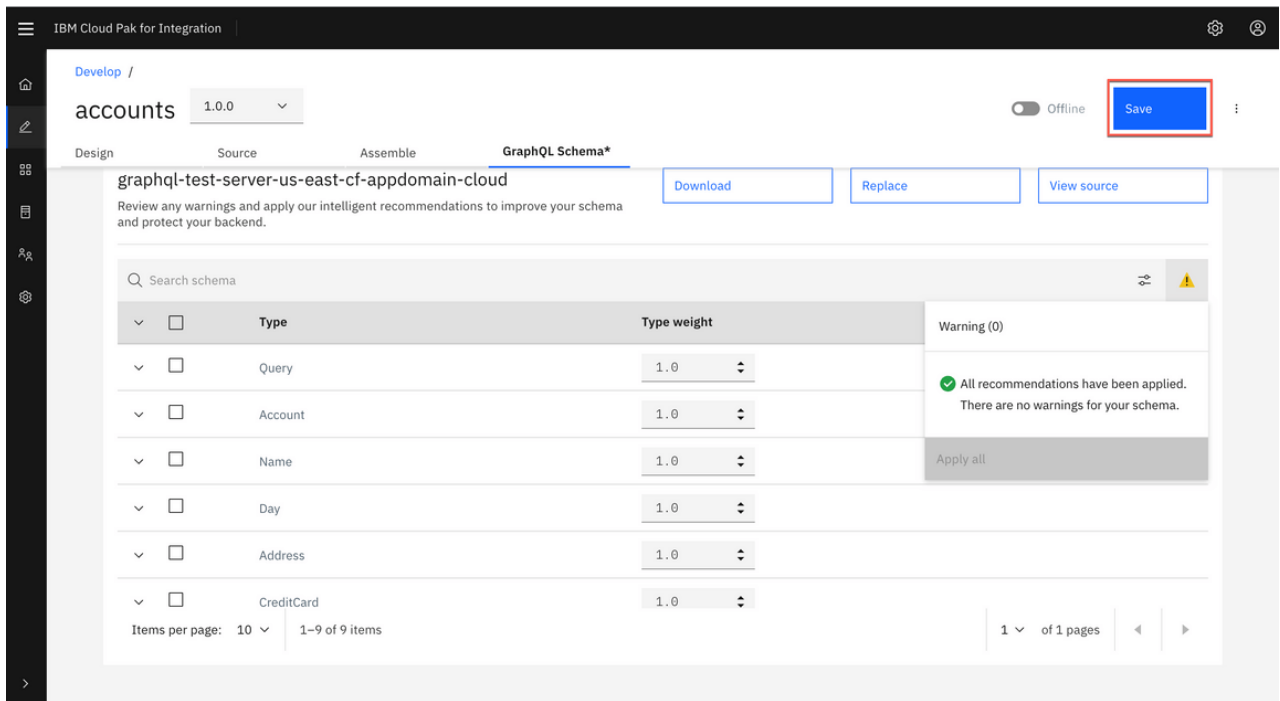


10. Select Query to review the warning details for this type. In addition to the warning details, the Warning window gives an option to fix the warning by apply the limits. Click Apply to continue.

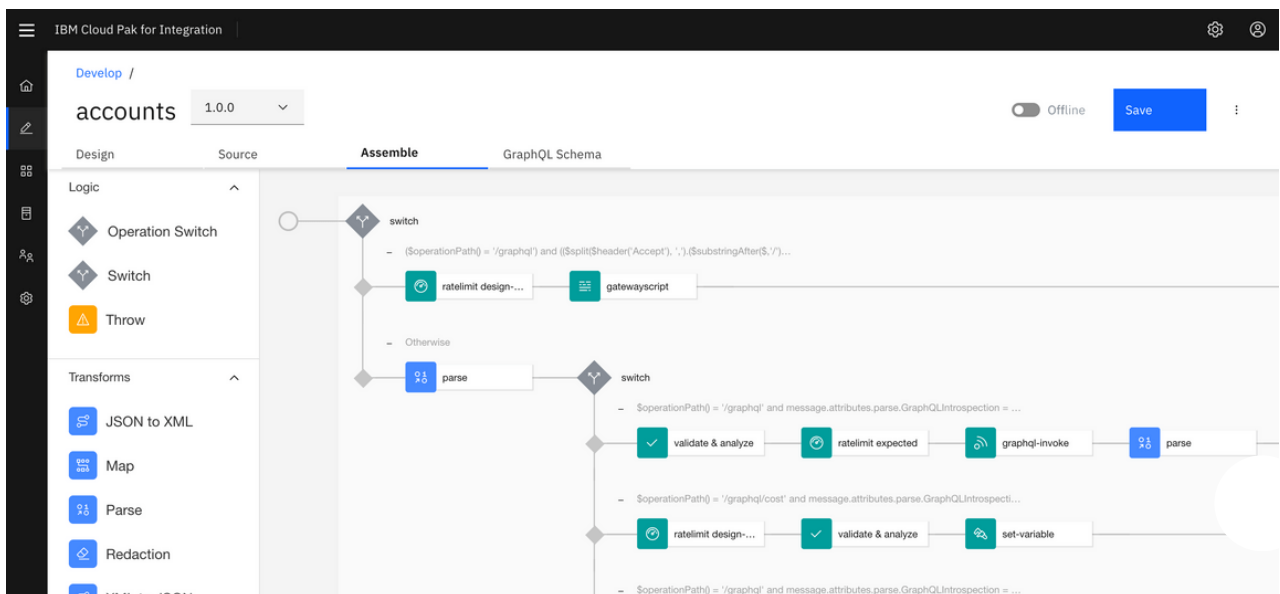




- After applying the recommendation the warning message is removed. Click Save to save the API definition .



- Select Assemble to view the API in Assembly view. The flow is automatically created as part of the initial API creation. You may apply addition logic by dragging and dropping items from the pallet to the canvas. In this tutorial, you are not adding additional logic.

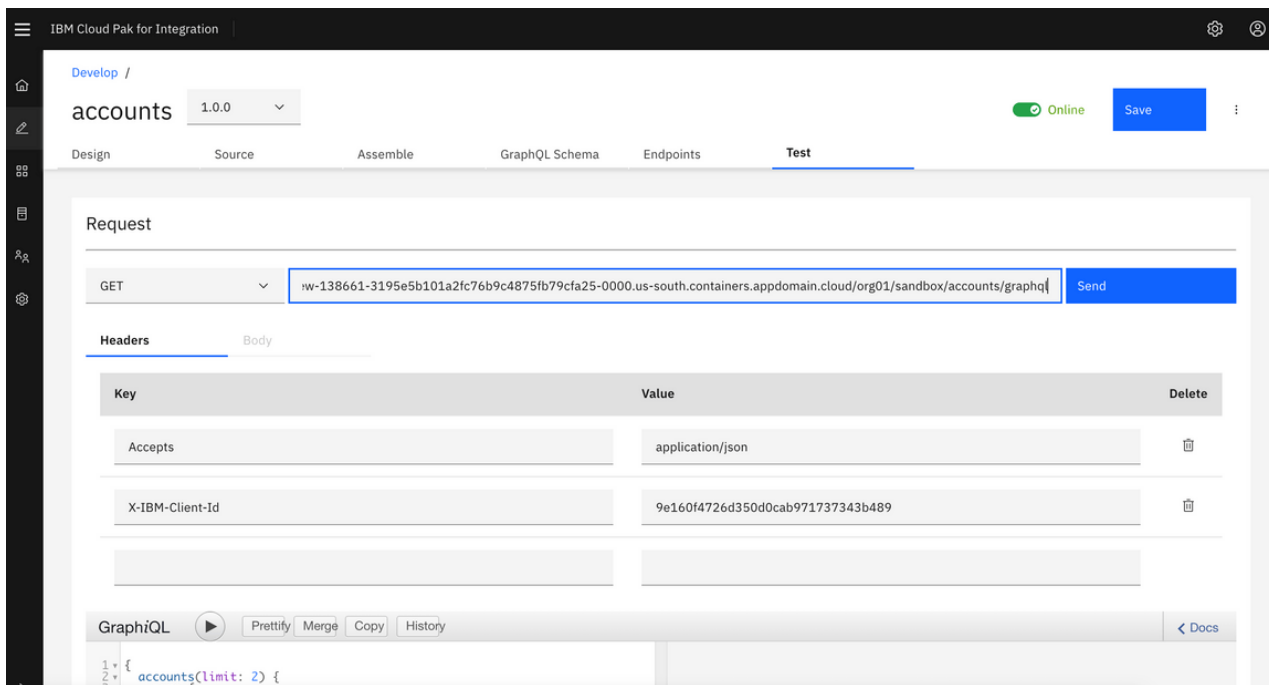




13. To publish the API toggle the button from offline to online.



14. After successfully publishing the API to API Gateway, additional tabs appear on the screen. Select Test tab to test the API.



15. To test the API select the end point by append graphql at the end of the request URL for GET method. The URL should look like this (your value may be different):

<https://apic-dev-gw-gateway-apic.cp4inew-138661->

3195e5b101a2fc76b9c4875fb79cfa25-0000.us-south.containers.appdomain.cloud/org01/sandbox/accounts/graphql

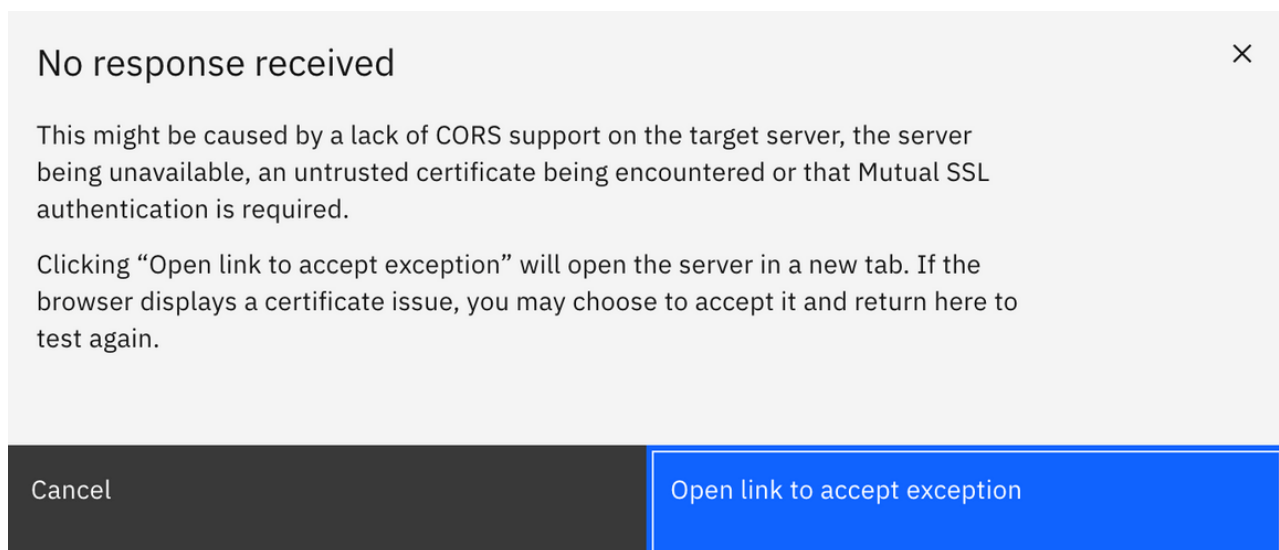
Click **Send** to test the API.

An empty string is returned since we have not specified query string yet.

16. API Connect Test feature also includes GraphQL Editor. Enter the following in in GraphQL Editor window. You may use Prettify option to view formatting string.

```
{
  accounts(limit: 2) {
    name {
      first
      last
    }
  }
}
```

17. Click Run button to send the GraphQL request to the end point. An exception window will appear when you use test the API for the first time.
18. Click accept Exception to continue.



You may see an invalid certificate issue page when you continue. This is because valid certificates are not applied in this test environment. If you are using Chrome

browser you can accept the exception by typing “thisisunsafe” anywhere in the window. Similar workarounds are available for other browsers. For firefox browser, follow the steps from here - <https://support.mozilla.org/en-US/kb/error-codes-secure-websites>



Your connection is not private

Attackers might be trying to steal your information from **apic-dev-gw-gateway-apic.cp4inew-138661-3195e5b101a2fc76b9c4875fb79cfa25-0000.us-south.containers.appdomain.cloud** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_INVALID

Help improve Chrome security by sending [URLs of some pages you visit, limited system information, and some page content](#) to Google. [Privacy policy](#)

Hide advanced

Reload

apic-dev-gw-gateway-apic.cp4inew-138661-3195e5b101a2fc76b9c4875fb79cfa25-0000.us-south.containers.appdomain.cloud normally uses encryption to protect your information. When Google Chrome tried to connect to apic-dev-gw-gateway-apic.cp4inew-138661-3195e5b101a2fc76b9c4875fb79cfa25-0000.us-south.containers.appdomain.cloud this time, the website sent back unusual and incorrect credentials. This may happen when an attacker is trying to pretend to be apic-dev-gw-gateway-apic.cp4inew-138661-3195e5b101a2fc76b9c4875fb79cfa25-0000.us-south.containers.appdomain.cloud, or a Wi-Fi sign-in screen has interrupted the connection. Your information is still secure because Google Chrome stopped the connection before any data was exchanged.

19. The results for GraphQL request are displayed on the right panel. In the request we limited the values to 2, so only two accounts are displayed. Change the value to 5 and submit the request again and notice that more accounts are returned in the results window.



20. You can request additional data fields (refer to the schema editor for details), GraphQL may reject if certain fields are locked or prohibited. To test this, enter CreditCard information in the request field.

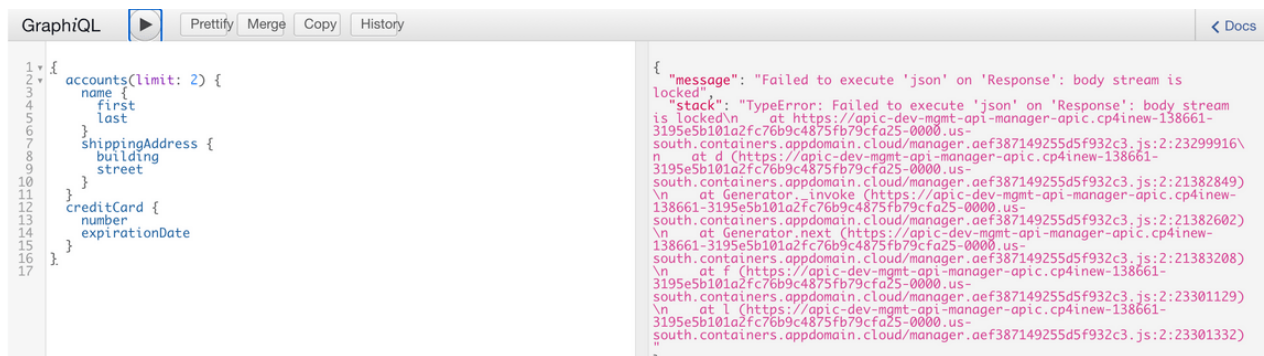
The complete request looks as follows:

```
{
  accounts(limit: 2) {
    name {
      first
      last
    }
    shippingAddress {
      building
      street
    }
  }
}
```

Show more

You will notice that an error message is displayed in the results window instead of the account values as the server restricts the query field creditCard.

(Note: The error message displayed may be different than what is shown in the screenshot).



- Submit another request to get account and shipping address details. This time the values are turned without errors.

The screenshot shows the GraphQL IDE interface. On the left, a query is defined with the following structure:

```

1 {
2   accounts(limit: 2) {
3     name {
4       first
5       last
6     }
7     shippingAddress {
8       building
9       street
10      state
11      zip
12    }
13  }
14 }
15

```

On the right, the JSON response is displayed:

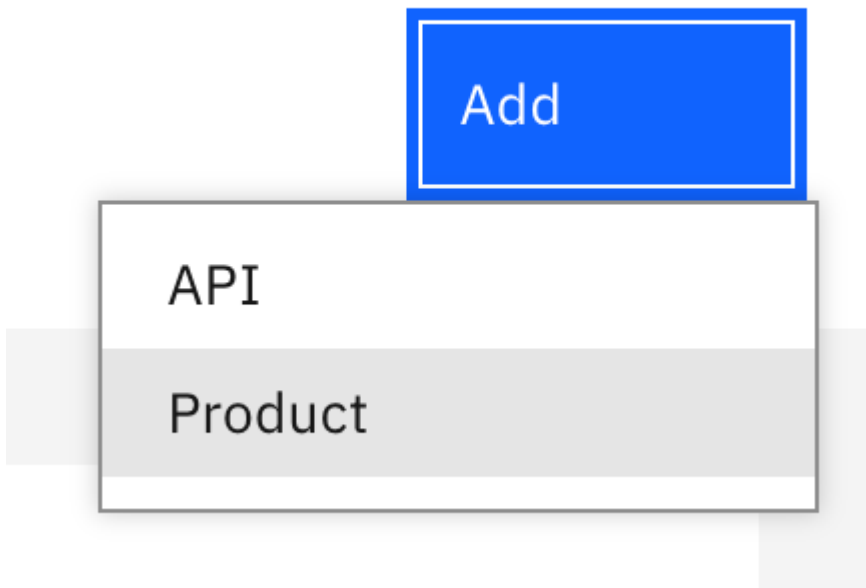
```

{
  "accounts": [
    {
      "name": {
        "first": "Gerson",
        "last": "Torp"
      },
      "shippingAddress": {
        "building": "26",
        "street": "Zuluaf Lakes",
        "state": "Maine",
        "zip": "26177"
      }
    },
    {
      "name": {
        "first": "Rodolfo",
        "last": "Kessler"
      },
      "shippingAddress": {
        "building": "3008",
        "street": "Keagan Stream",
        "state": "Ohio",
        "zip": "99148-2762"
      }
    }
  ]
}

```

Create a Product and apply GraphQL RateLimits

- To create a product Select Develop icon from left menu bar and click Add->Product



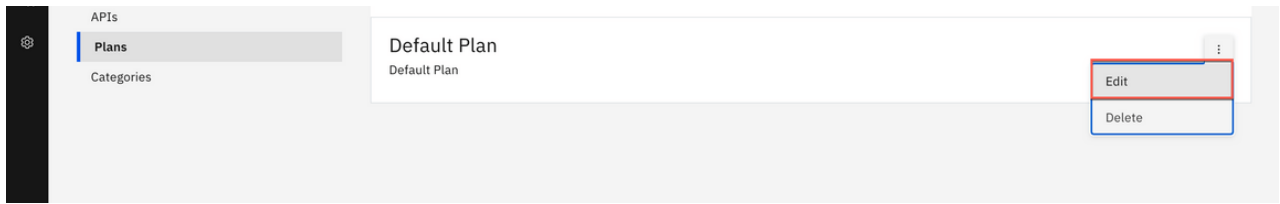
- Name the product as Accounts Product and click Next



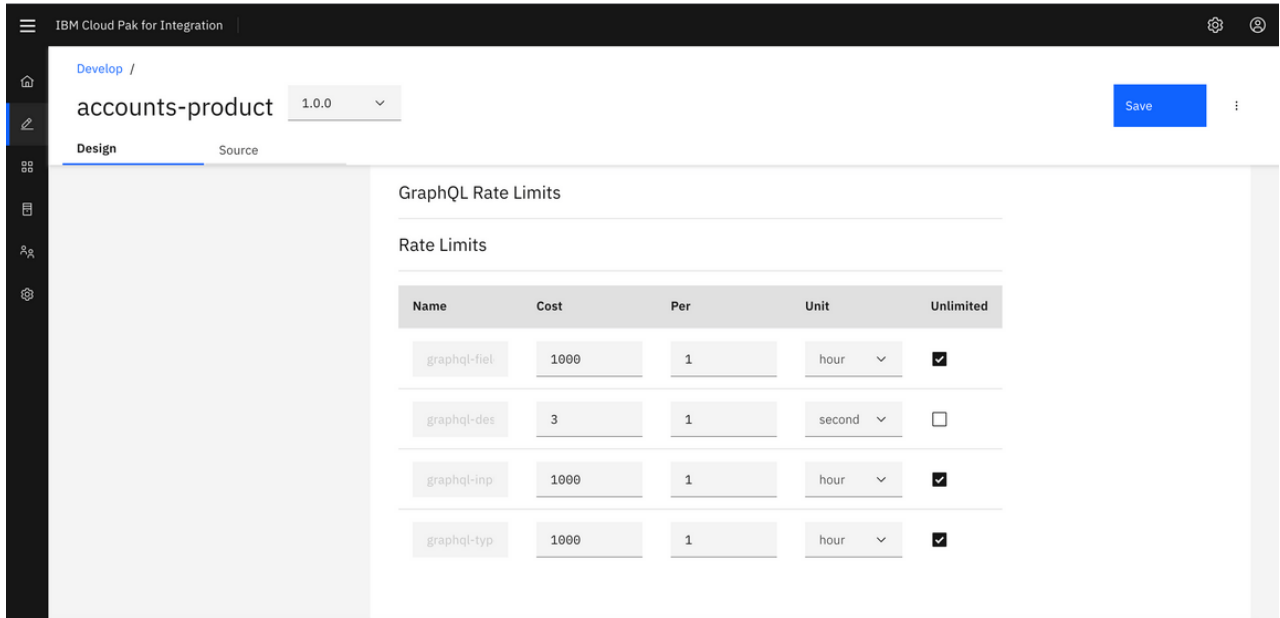
3. Add Accounts API to the product by selecting accounts. Click Next to continue to create the accounts-product.

| <input type="checkbox"/> | Title | Version | Description |
|-------------------------------------|-----------|---------|-------------|
| <input type="checkbox"/> | inventory | 1.0.0 | |
| <input checked="" type="checkbox"/> | accounts | 1.0.0 | |
| <input type="checkbox"/> | logistics | 2.0.0 | |
| <input type="checkbox"/> | financing | 1.0.0 | |
| <input type="checkbox"/> | inventory | 2.0.0 | |

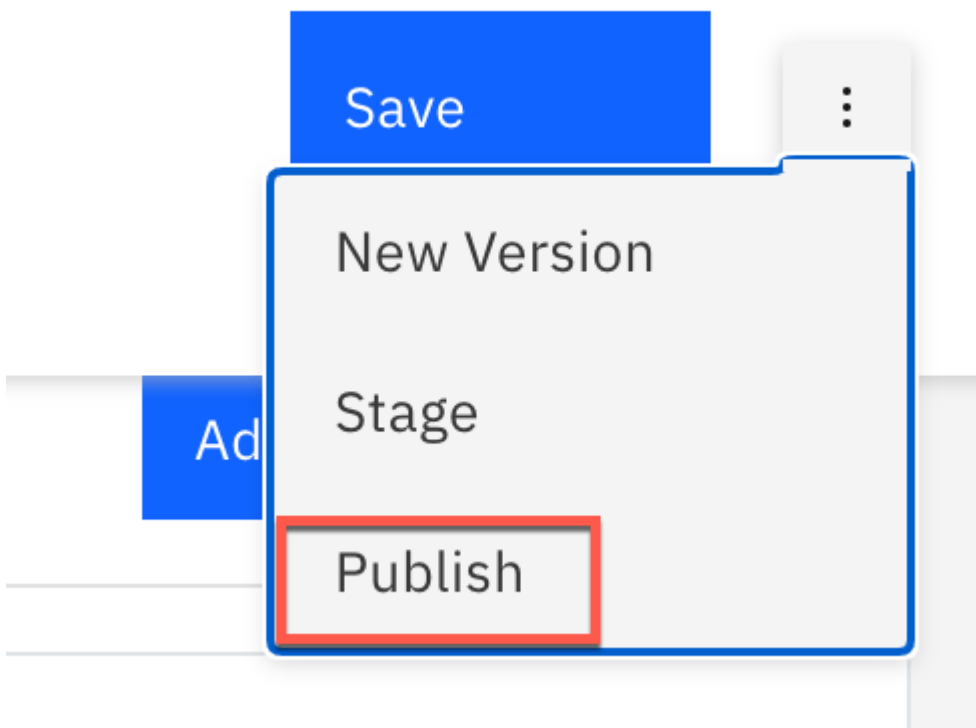
4. Edit the product accounts-product to review the details. The Default Plan contains traditional rate limiting options but you will notice another section that list GraphQL Rate Limits.



5. You will be able to adjust GraphQL cost parameters from this section. You can set a limit or make it unlimited. You will not be making any changes in this tutorial.



6. Publish the Product to Sandbox Catalog by selecting Publish



7. Accept the default for Visibility and Subscribability

Publish
Enable publishing of this product

Publish product

Visibility
Select the organizations or groups you would like to make this product visible to

Public
 Authenticated
 Custom

Subscribability
Select the organizations or groups you would like to subscribe to this product

Authenticated
 Custom

[Cancel](#) [Back](#) [Next](#)

Compose a new product by adding rate limits and plans
[Learn more](#)

8. Click Publish to complete publishing the product.

IBM Cloud Pak for Integration

Publish Product

Publish To

Catalog

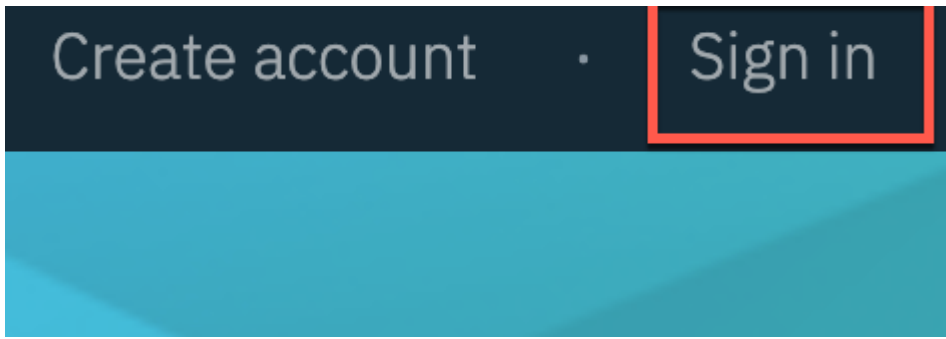
Sandbox

Publish to specific gateway services
By default, this product is published to all gateway services. You can also publish to specific gateway services by enabling this option.

[Cancel](#) [Publish](#)

Test GraphQL API from Developer Portal

1. Sign in to API Connect Developer Portal using the Developer Portal URL, username and password provided by the instructor.



API Developer Portal


Sign in

Sign in with Sandbox Catalog User Registry

Username

Password

Continue with

 admin

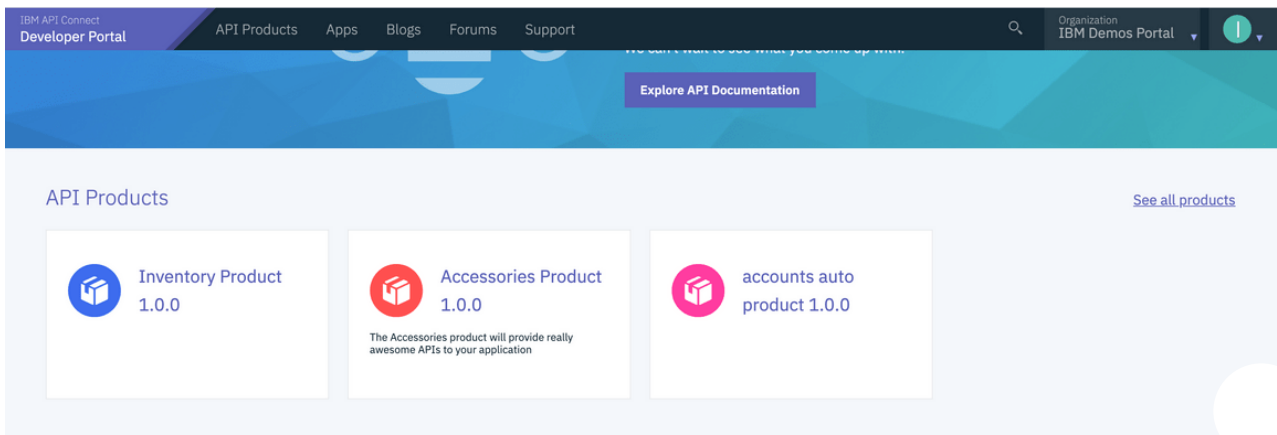
or


Sign in

[Forgot password?](#)

Don't have an account? [Sign up](#)


2. Developer Portal displays all the products that are currently published. Locate accounts-product and select the product.






Sign up

[Create a new account](#) and get started with our APIs. It's free to join.



Explore our APIs

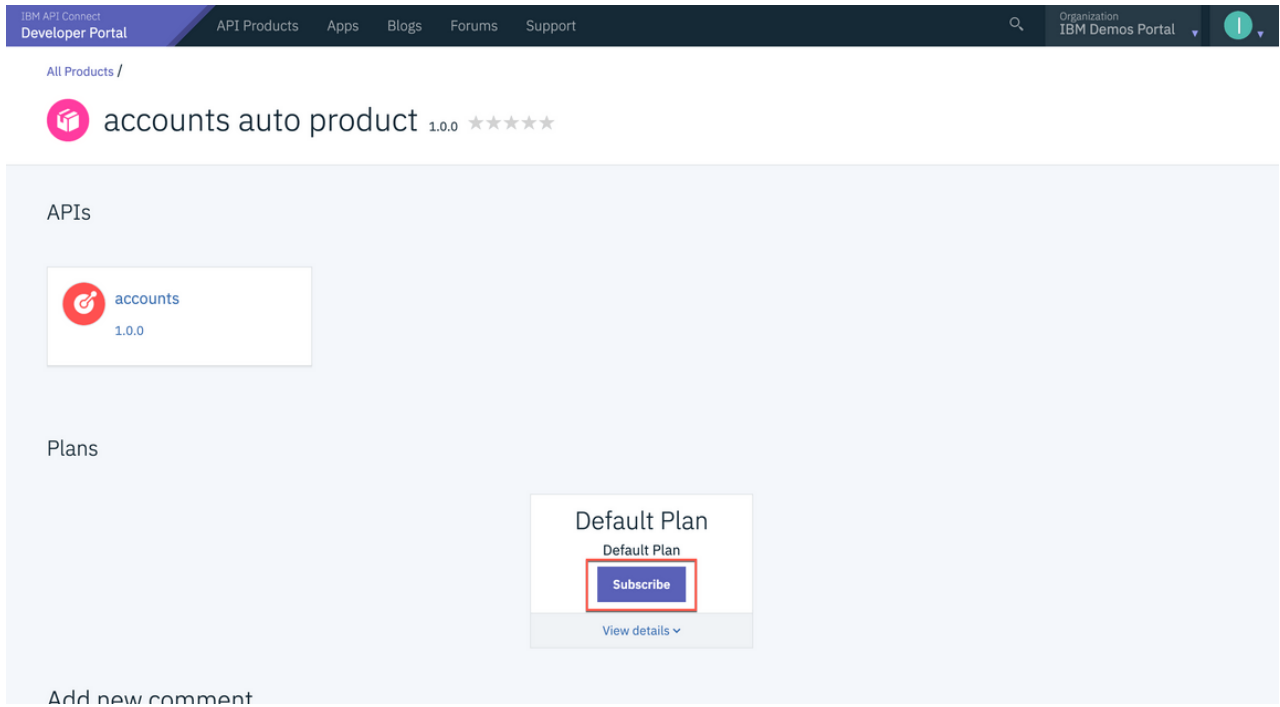
Take a look at our [API products](#) and quickly find APIs to construct a fully featured application



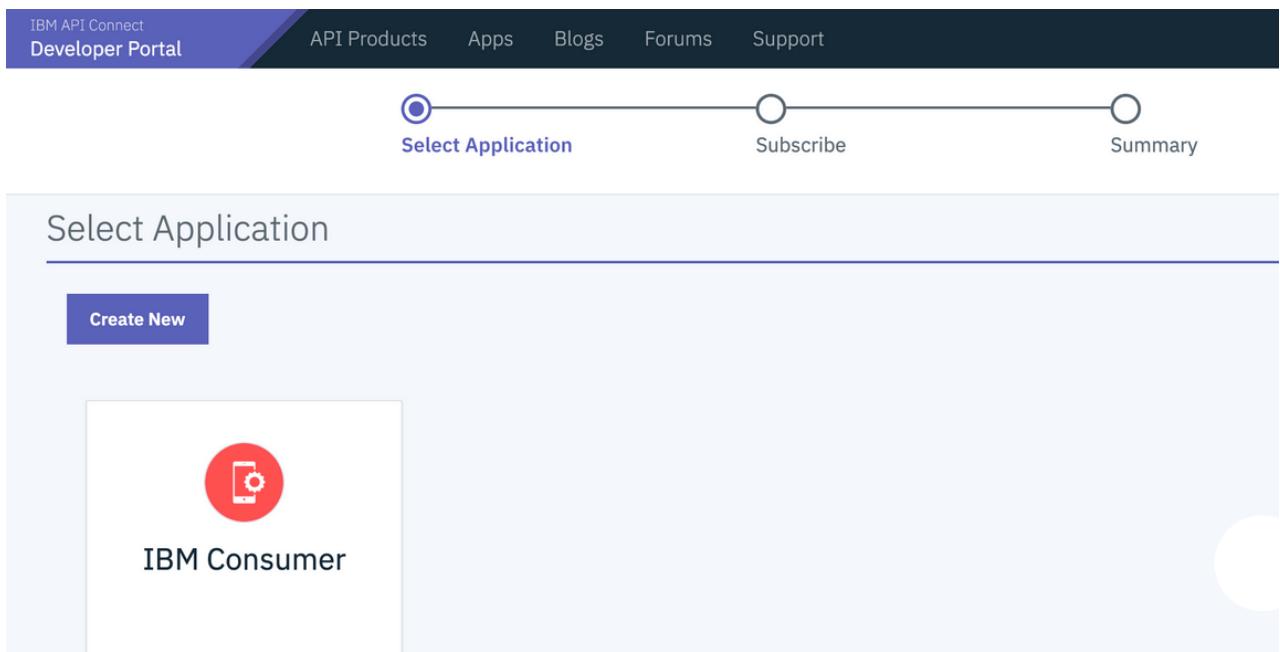
Create

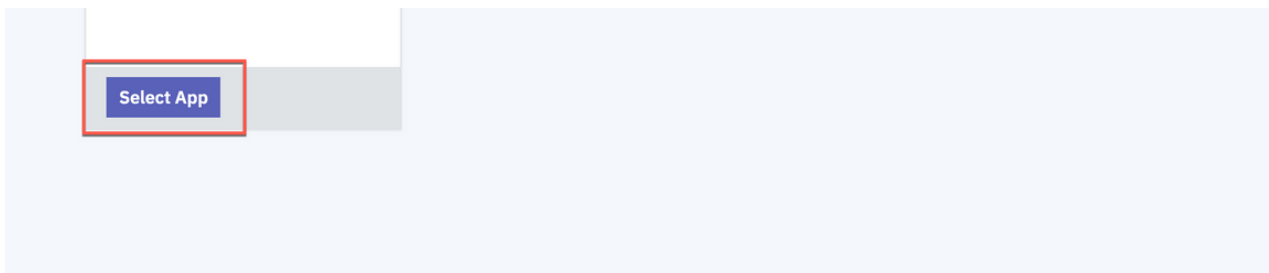
Subscribe to a plan and create your application to make use of our APIs.

- The selected product lists one API (accounts api) and one Plan (Default Plan). Select Subscribe option to subscribe to the plan.

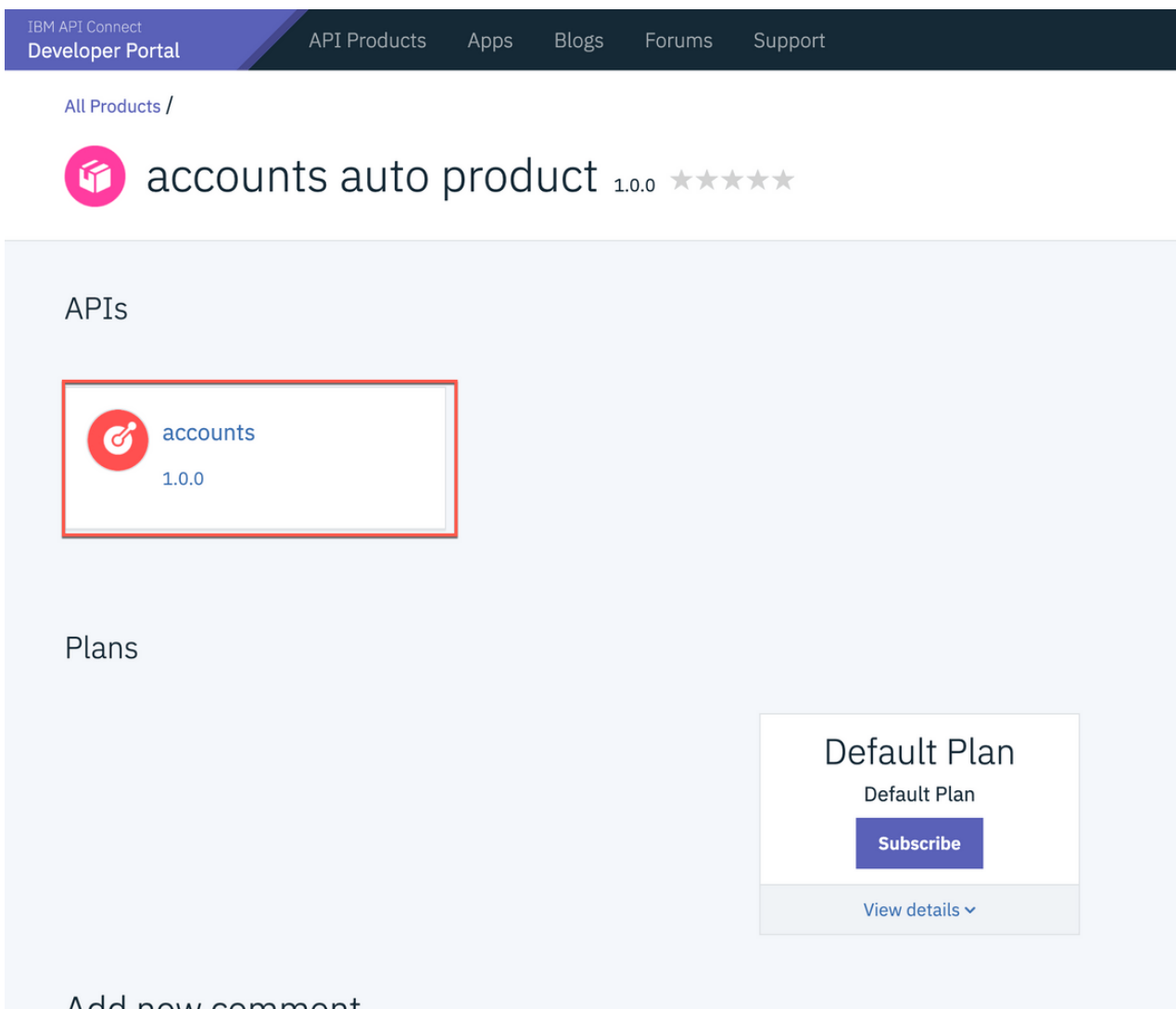


- The Plan subscription requires an Application. You can create a new Application or select the IBM Consumer application that is already created.

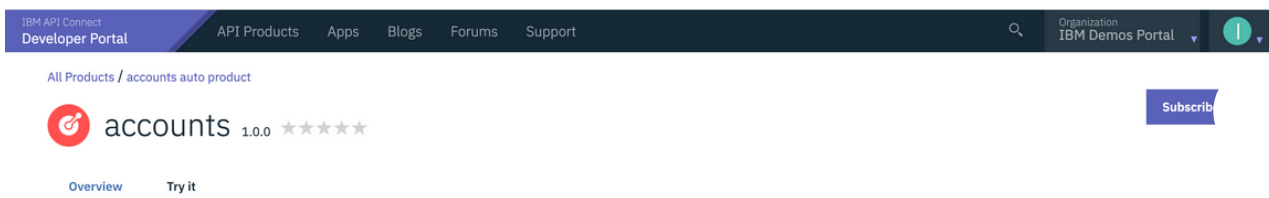




5. Select accounts API to review and test the API. Click accounts to continue



6. Developer Portal displays the type as GraphQL API. Currently Client ID security is enforced. Since there is only on Plan and one Consumer Application, the Consumer ID filed is already populated.



| | | | | | |
|----------------------------|---|-----------------|--|-----------------|--------------------------|
| Download Open API Document | ↓ | | | | |
| Type | GraphQL | | | | |
| Endpoint | Production, Development: https://apic-dev-gw-gateway-apic.cp4inew-138661-3195e5b101a2fc76b9c4875fb79cfa25-0000.us-south.containers.appdomain.cloud/org01/sandbox/accounts | | | | |
| Security | <table border="1"> <tr> <td>clientID</td> <td></td> </tr> <tr> <td>X-IBM-Client-Id</td> <td>apiKey located in header</td> </tr> </table> | clientID | | X-IBM-Client-Id | apiKey located in header |
| clientID | | | | | |
| X-IBM-Client-Id | apiKey located in header | | | | |
| Support | Discuss this API in the forum | | | | |

7. To continue to test the API, use the GraphQL Editor. Submit requests with different field names and watch how quickly the results are returned in the results window. If this was a REST API multiple end points will be needed to achieve the same result.

The screenshot shows the IBM API Connect Developer Portal interface. At the top, there's a navigation bar with 'API Products', 'Apps', 'Blogs', 'Forums', and 'Support'. Below that, the 'accounts' API is displayed with a 1.0.0 version and a 5-star rating. The 'Endpoint' section shows the production and development URLs. The 'Security' section is set to 'Identification' with 'Client ID' set to 'IBM Consumer'. Below this is the GraphQL Editor, which includes a 'Play' button and a 'Docs' link. The editor shows a GraphQL query on the left and the resulting JSON data on the right.

```

1 {
2   accounts(limit: 2) {
3     name {
4       first
5       last
6     }
7   }
8 }
9

```

```

{
  "data": {
    "accounts": [
      {
        "name": {
          "first": "Terrance",
          "last": "Abernathy"
        }
      },
      {
        "name": {
          "first": "Kylar",
          "last": "Kreiger"
        }
      }
    ]
  }
}

```

Summary

Congratulations! You have successfully created a GraphQL Proxy API, published the API as part of a Product, applied controls and rate limits, and tested using Developer Portal.

To learn more about Cloud Pak for Integration and API Life Cycle Management, visit <https://www.ibm.com/cloud/cloud-pak-for-integration>

[Edit this page on GitHub](#)

[IBM](#)

[Privacy](#)

[Terms of Use](#)

Have questions? open
an issue on GitHub.

Built with Gatsby Theme Carbon.
Copyright © 2022 IBM

