

# Anypoint DataGraph hands-on

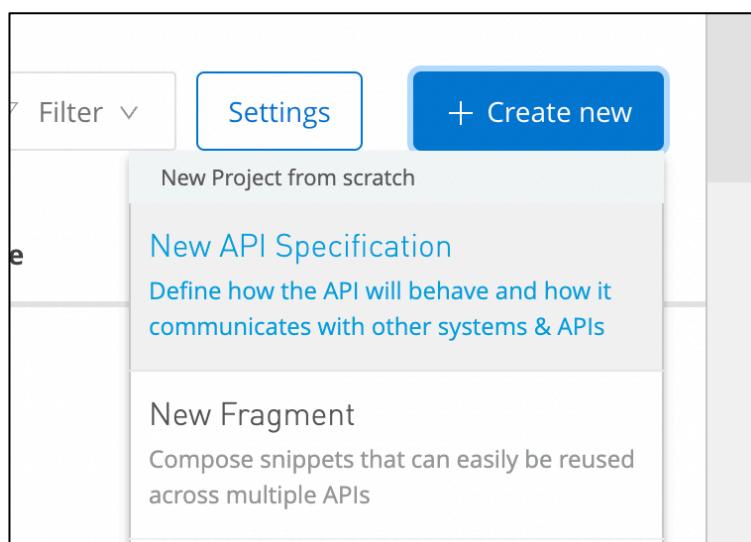
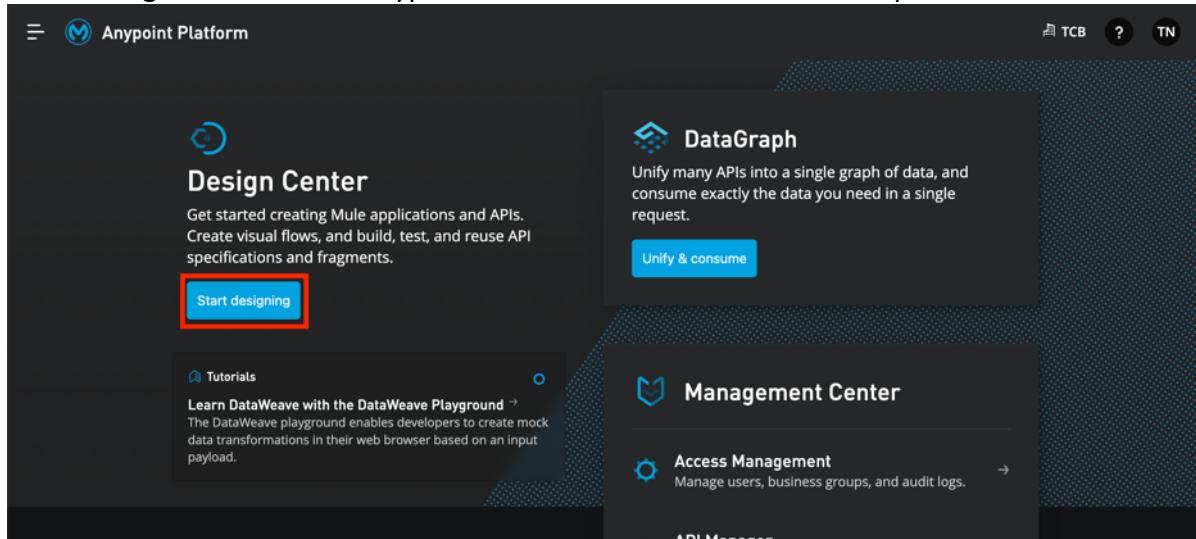


## Prerequisites:

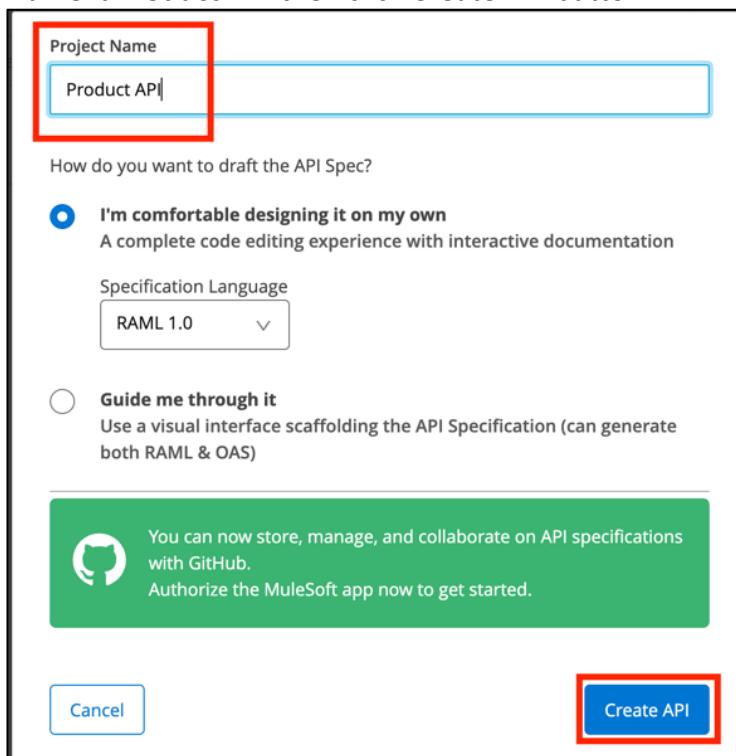
- Install Anypoint Studio to local machine
- Sign up a free trial account in Anypoint Platform
- Clone Mulesoft apps repo from Github to local machine:  
<https://github.com/tuannguyen511/anypoint-datagraph-demo>

## 1. Create Product API and Order API

Go to Design Center in the Anypoint Platform then create New API Specification.



Name it **Product API** then click Create API button.



Copy the content of file **product-api.raml** then paste it to the design area in the Design Center. After that, publish this API to Exchange.

```
#RAML 1.0
title: Product API
version: v1.0

types:
  Product:
    properties:
      productCode:
        type: string
        description: Product Code
      productName?:
        type: string
        description: Product Name
      productLine?:
        type: string
        description: Product Line
      productVendor?:
        type: string
        description: Product Vendor
      productDescription?:
        type: string
        description: Product Description
      quantityInStock?:
        type: number
        description: Quantity in stock
      price?:
        type: number
        description: Price

/products
  get:
```

## Product API

Asset version (required)

API version (required)

Specified in `rootfile`

LifeCycle State ⓘ

Stable  
 Development

Additional help

- [Changing a project's main/root file](#)
- [What is an API version?](#)

Stable  
State of release, ready to consume

Development  
In Process of Design and Development

> More options

Cancel Publish to Exchange

From Anypoint Platform main page, navigate to API Manager.

The screenshot shows the Anypoint Platform main dashboard. At the top, there is a navigation bar with the Anypoint Platform logo, TCB, and TN buttons. Below the navigation bar, there are three main sections: Design Center, DataGraph, and Exchange. The Exchange section is highlighted with a red box around the 'API Manager' link. The 'API Manager' link is described as managing clients, policies, SLAs, traffic, and alerts.

Design Center

Get started creating Mule applications and APIs. Create visual flows, and build, test, and reuse API specifications and fragments.

Start designing

Tutorials

Learn DataWeave with the DataWeave Playground →  
The DataWeave playground enables developers to create mock data transformations in their web browser based on an input payload.

DataGraph

Unify many APIs into a single graph of data, and consume exactly the data you need in a single request.

Unify & consume

Exchange

API Manager

Manage clients, policies, SLAs, traffic, and alerts.

Access Management

Manage users, business groups, and audit logs.

Management Center

Runtime Manager

Select Add new API.

The screenshot shows the 'API Administration (Sandbox)' section of the API Manager. On the left, there's a sidebar with options like 'Sandbox', 'API Administration', 'API Groups', 'Automated Policies', 'Client Applications', 'Custom Policies', and 'Analytics'. In the center, there's a 'Add API' button and a 'Filter by' dropdown. Below these are two main options: 'Add new API' (which is highlighted with a red box) and 'Import API from zip file'. The 'Add new API' option has a sub-description: 'Start a new API configuration from scratch through a guided experience'.

Select as shown below and then click Next.

This screenshot shows the 'Add API' configuration dialog. On the left, there's a sidebar with 'Sandbox' selected and other options like 'Runtime', 'API', 'Endpoint', and 'Review'. The main area is titled 'Runtime' with the sub-instruction: 'Before selecting an API to manage, choose a runtime.' It shows three options: 'Flex Gateway' (marked as 'NEW'), 'Mule Gateway' (selected), and 'Service Mesh'. Below this, there are sections for 'Proxy type' (with 'Connect to existing application (basic endpoint)' selected), 'Mule version' (with 'Mule 4 (recommended)' selected), and buttons for 'Cancel' and 'Next'.

Choose **Product API** then click Next until getting the Review page. Click Save to finish.

API

Select the API you want to manage.

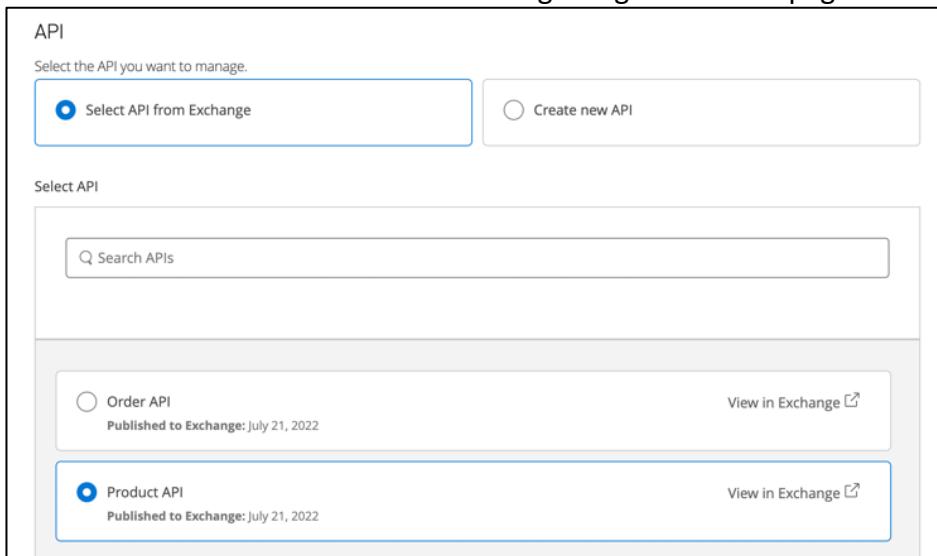
Select API from Exchange       Create new API

Select API

Search APIs

Order API      View in Exchange ↗  
Published to Exchange: July 21, 2022

Product API      View in Exchange ↗  
Published to Exchange: July 21, 2022



Review

Review your selections before saving and deploying your API instance. You can also save the configuration and deploy it later.

Runtime ↗ Edit

Runtime type **Mule Gateway**

Proxy type **Basic Endpoint**

API ↗ Edit

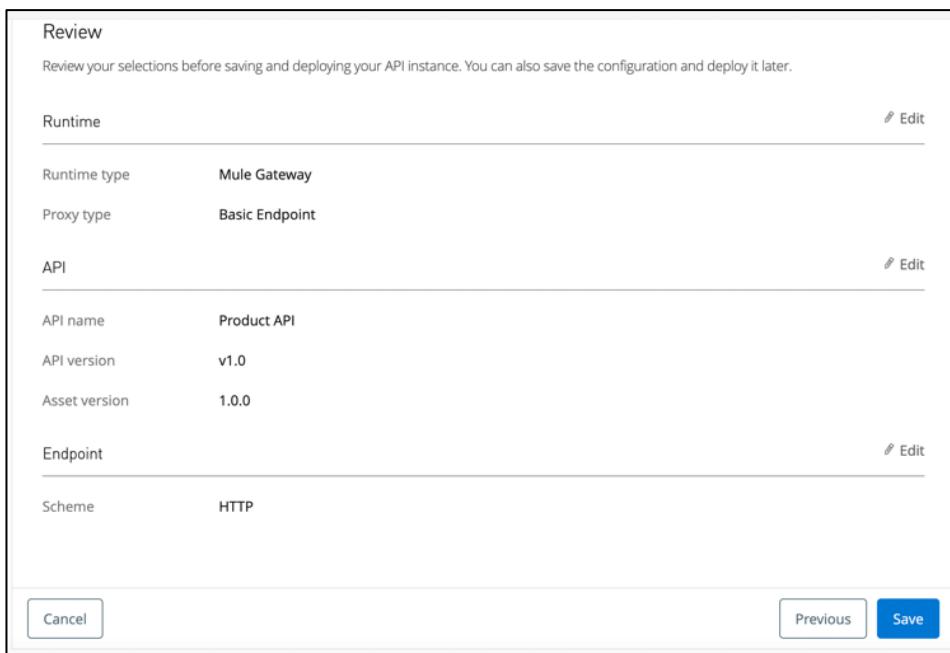
API name **Product API**

API version **v1.0**

Asset version **1.0.0**

Endpoint ↗ Edit

Scheme **HTTP**



After the API is successfully created, click on it and configure Basic Authentication for it.

The screenshots illustrate the steps to add a policy to an API in the API Manager:

**Top Screenshot (API Administration (Sandbox) - Product API (v1.0) - Policies):**

- Left sidebar: SANDBOX, Alerts, Contracts, **Policies**, SLA Tiers, Settings.
- Main area: APIs / Product API / Policies. Sub-section: Automated policies. Message: No automated policies applied.
- Main area: API-level policies. Sub-section: + Add policy (button highlighted with a red box).
- Main area: API-level policies. Sub-section: No policies applied.

**Bottom Screenshot (API Administration (Sandbox) - Product API (v1.0) - Policies):**

- Left sidebar: SANDBOX, Policies.
- Header: TCB, ?, TN.
- Main area: Browse by category. Filter: SECURITY (10) (selected), ALL CATEGORIES (19), QUALITY OF SERVICE (4), COMPLIANCE (2), TROUBLESHOOTING (1), TRANSFORMATION (2).
- Main area: Security section. Policies listed:
  - Basic Authentication - LDAP (radio button)
  - JSON threat protection
  - OAuth 2.0 access token enforcement using Mule OAuth provider
  - JWT Validation
- Bottom right: Cancel, Next.

User Name

User Password

 Show

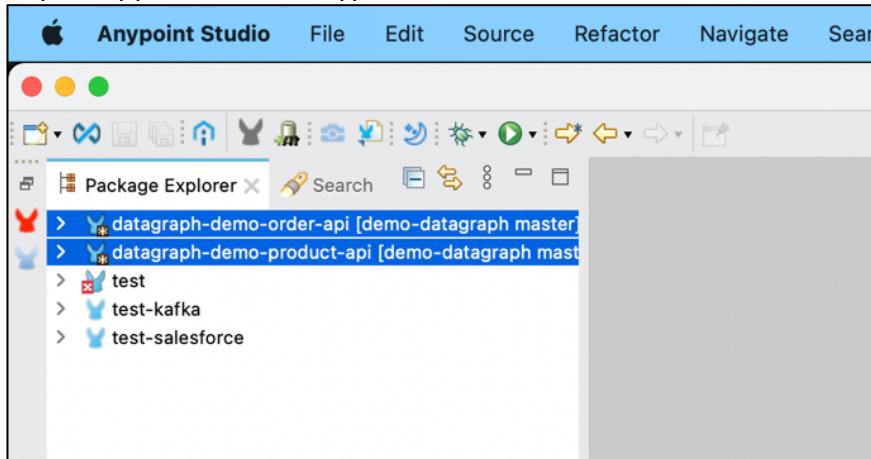
Advanced options >  
Configure policy version, methods and resources

Previous Apply

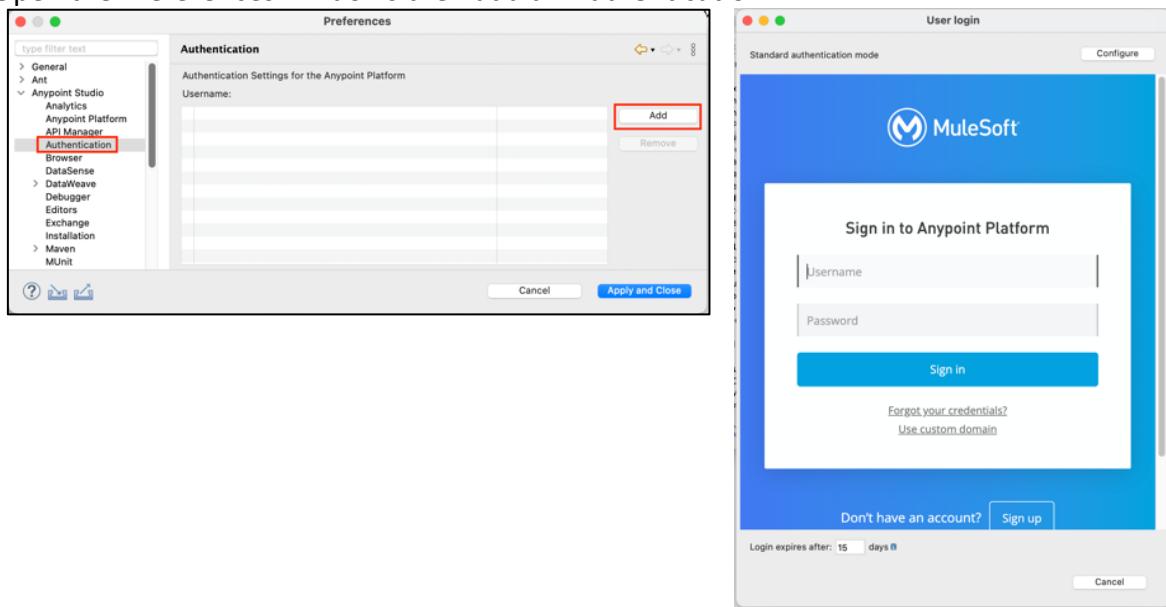
*Perform the same steps as when creating the Product API to create the Order API.*

## 2. Deploy Product and Order application to CloudHub

Import applications to Anypoint Studio.



Open the Preferences windows then add an Authentication.

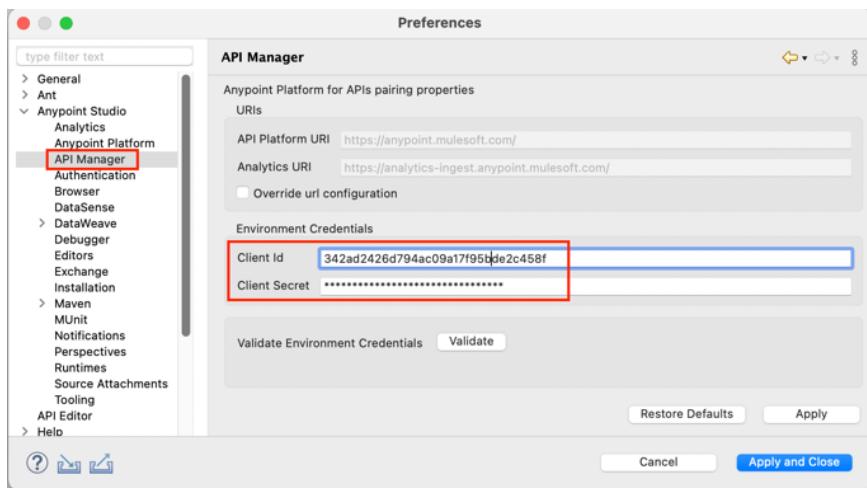


Now update organization Client id and secret in AnyPoint Studio.

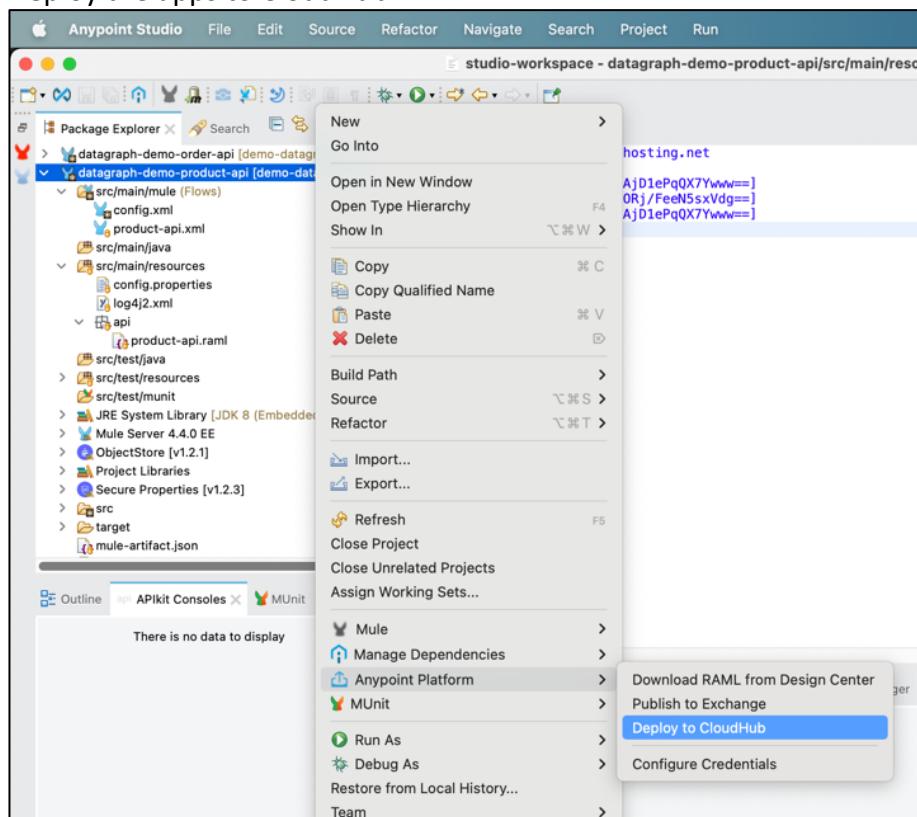
We need to get the credentials from AnyPoint Platform.

The screenshot displays the 'Access Management' interface in the AnyPoint Platform. On the left, the 'Organization' tab is selected under 'ACCESS MANAGEMENT'. The main area shows a table with one organization entry: 'TCB' (Name) with 'Tuan Nguyen' (Owner). On the right, a modal dialog titled 'Organization info' is open for the 'TCB' organization. It contains the following fields:

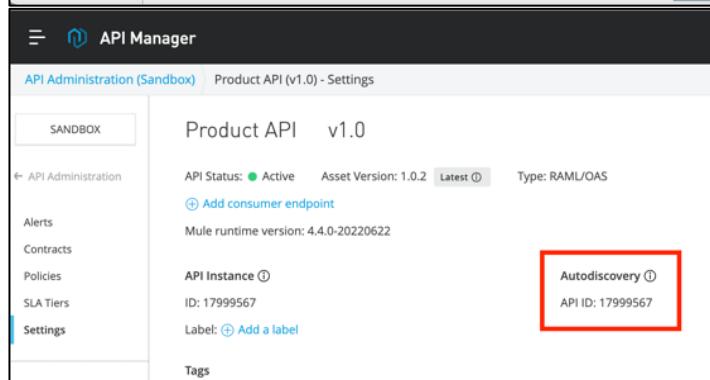
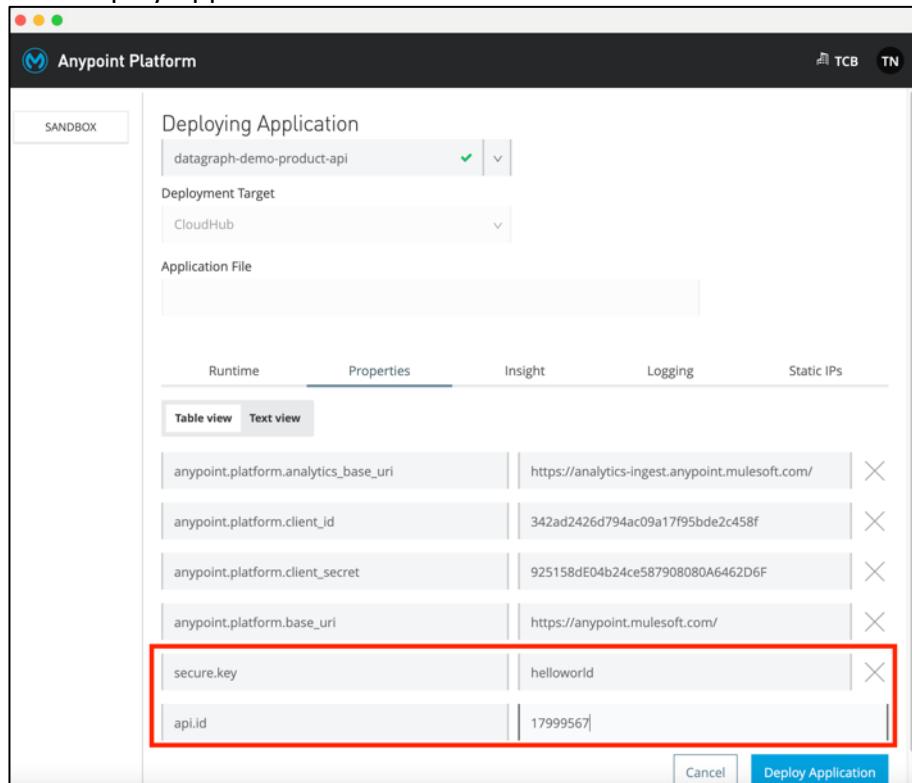
- Organization name \*: TCB
- Organization domain \*: tcb-804
- Owner \*: Tuan Nguyen
- Organization Id: 89924bcb-67e2-4b04-a925-9ac1ddb12cd3
- Client Id: 342ad2426d794ac09a17f95bde2c458f
- Client Secret: (redacted)



## Deploy the apps to CloudHub.



A pop-up shows up. Switch to Properties tab and add two more properties.  
The value for **api.id** can be found in API Manager in Anypoint Platform.  
Click Deploy Application.



Wait for a while until the application is deployed. Click on the Close Window button. Check the status of deployed apps in Runtime Manager in Anypoint Platform.

The screenshot shows two windows side-by-side. The top window is titled 'Runtime Manager' and displays a message: 'Deploying datagraph-demo-product-api to CloudHub'. It includes a note 'You may close this window at any time.' and two buttons: 'Open in Browser' and 'Close Window'. The bottom window is also titled 'Runtime Manager' and shows a list of applications. The sidebar on the left lists 'Sandbox', 'Applications' (which is selected), 'Servers', 'Flex Gateways', 'Alerts', 'VPCs', and 'Load Balancers'. The main area shows 'All Applications (2)'. A table lists the applications:

Name	Server	Status	Runtime Version	Date Modified
datagraph-demo-order-api	CloudHub	Started	4.4.0	2022-07-21 15:23:17
datagraph-demo-product-api	CloudHub	Started	4.4.0	2022-07-21 17:09:22

### 3. Add Product API to DataGraph

Go to DataGraph and click on Add API button.

The screenshot shows the DataGraph interface. At the top, there's a dropdown menu set to 'Unified Data Graph Sandbox'. Below it is a large central area with a 'Start building your unified schema!' heading and a 'Add API' button, which is highlighted with a red box. At the bottom, there are three cards: 'DataGraph Quick Start Guide', 'DataGraph Tutorial', and 'DataGraph Fundamental Concepts'.

Select Product API then click on Next button.

The screenshot shows the DataGraph interface with the title 'DataGraph' at the top. Below it, a navigation bar includes 'Sandbox / Add API' and steps 1 through 5: 'Select API', 'Configure URL', 'Configure Security', 'Preview Schema', and 'Edit Schema'. A search bar labeled 'Search Exchange' is present. Two API entries are listed: 'Order API' and 'Product API'. The 'Product API' entry is highlighted with a red box. At the bottom, there are 'Cancel' and 'Next: Configure URL' buttons, with the 'Next' button also highlighted by a red box.

Select appropriate API version then click Confirm.

The screenshot shows the DataGraph interface with the title 'DataGraph' at the top. Below it, a navigation bar includes 'Sandbox / Add API' and steps 1 through 5: 'Select API', 'Configure URL', 'Configure Security', 'Preview Schema', and 'Edit Schema'. A section titled 'Product API' is shown with the message 'To configure a URL first select and confirm the API version you want to add'. It contains two dropdown menus: 'API version' set to 'v1.0 latest' and 'Asset version' set to '1.0.0 latest'. To the right is an 'About this API' panel with links to 'View API specification' and 'View API documentation'. At the bottom, there are 'Cancel', 'Back', and 'Next: Configure Security' buttons, with the 'Next' button highlighted by a red box. A red box also highlights the 'Confirm selection' button in the center.

Scroll down to add API URL.

The domain can be found in Runtime Manager page.

Sandbox / Add API    ① Select API — ② **Configure URL** — ③ Configure Security — ④ Preview Schema — ⑤ Edit Schema

Add/Edit API URL  
This will be used to make requests to the API

Get an existing URL from Anypoint Platform  
This is for when the API is being managed in API manager or if the API URL has been manually added to Exchange.

Add a new URL  
Use this option if you want to manually add a URL for this API.

No URL found in the API specification  
Use the URL in API specification

Add URL    Edit URL

http://datagraph-demo-product-api.us-e2.cloudhub.io/api

Cancel    Back    **Next: Configure Security**

SANDBOX

Applications

Servers

Flex Gateways

Alerts

VPCs

Load Balancers

Deploy application

Search Applications

All Applications (2)

Name ^

Server

datagraph-demo-order-api

CloudHub

datagraph-demo-product-api

CloudHub

datagraph-demo-product-api

Domain: **datagraph-demo-product-api.us-e2.cloudhub.io** - last Updated 2022-07-21 5:09:22

Mule messages

900  
800  
700  
600  
500  
400  
300  
200  
100  
0

09:30:49 12:00:49 14:30:49 17:00:49 19:30:49

In the next step, configure the Basic authentication when calling the Product API. The username and password are the same as when we configured the policy for the Product API in part 1.

Sandbox / Add API    1 Select API — 2 Configure URL — 3 Configure Security — 4 Preview Schema — 5 Edit Schema

**Product API**

**Authentication**  
Select the authentication policy and provide credentials associated with this API's GET endpoints

Select authentication policy  
Supported authentication policy types are limited to those listed below. [Learn more](#)

Basic Auth

Username Password  
productapi ..... Show

About this API  
[View API specification](#)  
[View API documentation](#)

Cancel Back Next: Preview Schema

Click on Next button.

Sandbox / Add API    1 Select API — 2 Configure URL — 3 Configure Security — 4 Preview Schema — 5 Edit Schema

**Product API**  
v1.0.1.0.0

[API documentation](#) [API specification](#)

API schema preview mode

**API schema**  
Search API Schema...  
All types in this API schema

Operation type  
Query

Level 1 types  
Product

**Query** QueryType  
Type description

Query methods ⓘ  
Search methods... Expand All

Method name ⓘ  
> products (offset:Int!, limit:Int!): [Product]  
> productsByProductCode (productCode:String!): Product

Cancel Back Next: Edit Schema

Select **Product** object type then click on Enable collaboration.

This will enable you to create links from other types to this **Product** type.

The screenshot shows the DataGraph interface for the 'Product API'. On the left, there's a sidebar with 'API schema' and a search bar. The main area shows the 'Product' object type with its details and a 'Collaboration permissions' section. In this section, a button labeled 'Enable collaboration' is highlighted with a red box. At the bottom right of the main area, there are 'Cancel', 'Back', and 'Next: Add to unified schema' buttons.

Select the default query method

`productsByProductCode(productCode:String!): Product`  
then click Next.

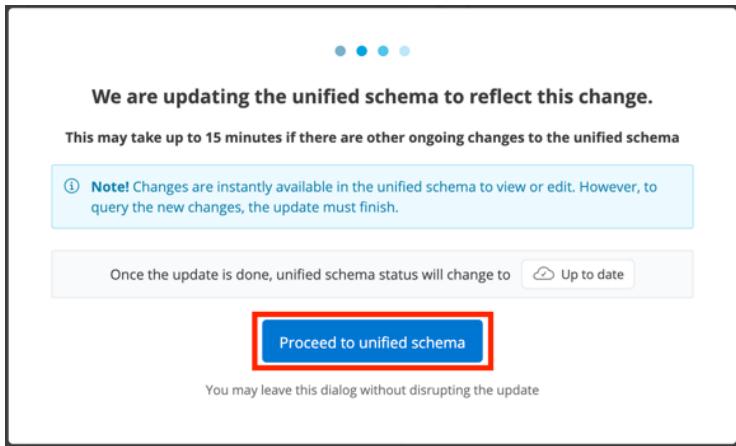
The screenshot shows a modal dialog titled 'Enable collaboration'. It has two steps: 'Select a default query method' (which is active) and 'Select a primary key'. Below this, there's a section for selecting a default query method with a radio button for 'productsByProductCode'. This radio button is highlighted with a red box. At the bottom right of the dialog, there are 'Cancel' and 'Next' buttons, with 'Next' also highlighted with a red box.

Select the primary key `productCode` (`String!`) then click Confirm.

The screenshot shows the 'Enable collaboration' dialog. Step 2, 'Select a primary key', is highlighted. A note says: '⚠ Note! Make sure that the mandatory argument in the default query method and the primary key both identify the same object.' The 'Default query method for this type' is set to `productsByProductCode (productCode:String!): Product`. In the 'Mandatory arguments' section, `productCode:String!` is listed. The 'Select primary keys' dropdown contains `productCode (String!)`, which is highlighted with a red box. At the bottom are 'Cancel', 'Back', and a large blue 'Confirm' button, also highlighted with a red box.

Click Next: Add to unified schema. Then click on Proceed to unified schema.

The screenshot shows the DataGraph API schema configuration for the 'Product' type. Under 'Collaboration permissions', 'Collaboration (Enabled)' is checked. The 'Default query method for this type' is set to `productsByProductCode (productCode:String!): Product`. The 'Primary key for this type' is set to `productCode:String!`. At the bottom, there is a 'Next: Add to unified schema' button, which is highlighted with a red box.



When the status changes to “Up to date”, indicating that the unified schema has been updated with your changes.

Click on Run Operation to try calling the DataGraph API.

Unified Data Graph Sandbox

Unified schema

Up to date

{ } Run Operation

Unified Schema

All types in unified schema

Query

Method name

productsByProductCode (productCode:String!): Product

products (limit:Int!, offset:Int!): [Product]

Select **Create a new application and use it immediately** then click Next.

**Request submitted successfully!**

You may use either a client application or a Client ID + Secret pair to manage your access for running queries, both here and for any external apps you develop (for example, a mobile or web app).

**How do you want to get access to run your queries?**

You can change this later whenever you want.

Use one of my existing applications  
Select an application ▾

Create a new application and use it immediately

I already have credentials (Client ID + Secret) and want to use them immediately

⚠ Failed while requesting tiers

Cancel Next

Complete the fields then click Next.

**Request submitted successfully!**

**Create a new application**

**Application name**  
Use a descriptive name that will help you and others easily identify it later.

Test DataGraph

**Description (Optional)**  
Enter a description

**Application URL (Optional)**  
https://yourcompany.com/applicationURL

**OAuth 2.0 redirect URIs (Optional) ⓘ**  
https://yourcompany.com/callback

**Cancel** **Back** **Next**

Try calling the DataGraph API with simple query.

The screenshot shows the DataGraph API interface. At the top, there's a navigation bar with tabs for 'Unified Data Graph' and 'Sandbox'. Below the navigation, there are buttons for 'Run', 'Prettyify', and 'History'. The main area is titled 'Unified schema / Run Operation'. A code editor window displays a GraphQL query:

```
query testQuerySomeProduct($limit: Int!, $offset: Int!) {
  products(limit: $limit, offset: $offset) {
    productCode
    productName
    productLine
    productVendor
    productDescription
  }
}
```

Below the query, there's a section for 'QUERY VARIABLES' containing:

```
{
  "limit": 10,
  "offset": 0
}
```

The results pane shows the JSON response from the API. It includes a 'data' field with an array of products. The first product is described as a '1952 Harley Davidson Ultimate Chopper' with a detailed description about its features and care requirements. Subsequent products listed include a '1952 Alpine Renault 1300', 'Highway 66 Mini Classics', and 'Official Moto Guzzi logos and insignias, saddle bags located on side of motorcycle, detailed engine, working steering, working suspension, two leather seats, luggage rack, dual exhaust pipes, small saddle bag located on handle bars, two-tone paint with chrome accents, superior die-cast detail, rotating wheels, working kick stand, diecast metal with plastic parts and baked enamel finish.'

## 4. Add Order API to DataGraph

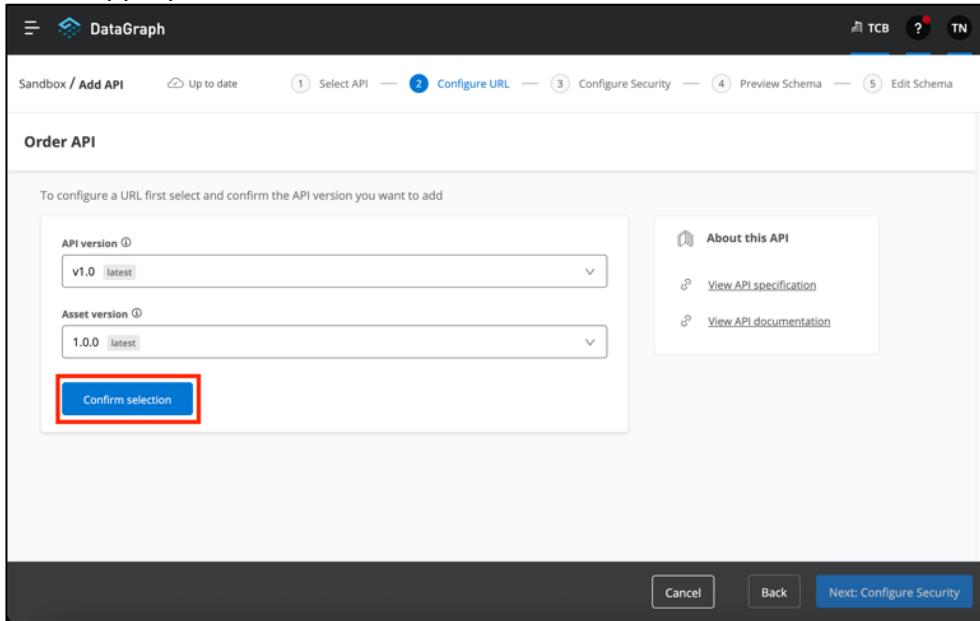
From DataGraph home page → Unified Schema → Add API.

The screenshot shows the DataGraph Unified Schema interface. On the left, there's a sidebar with options like 'Unified Data Graph Sandbox', 'Overview', 'Unified Schema' (which is selected), 'List of APIs added', and 'Response logs'. The main area is titled 'Unified schema' and contains a search bar for 'API Schema...' and a dropdown for 'All types in unified schema'. Below this, under 'Operation type', 'Query' is selected. A 'Query methods' section shows two entries: '> productsByProductCode (productCode:String!): Product' and '> products (limit:Int!, offset:Int!): [Product]'. In the top right corner, there's a blue button labeled '+ Add API' which is highlighted with a red box.

Select Order API. Click Next.

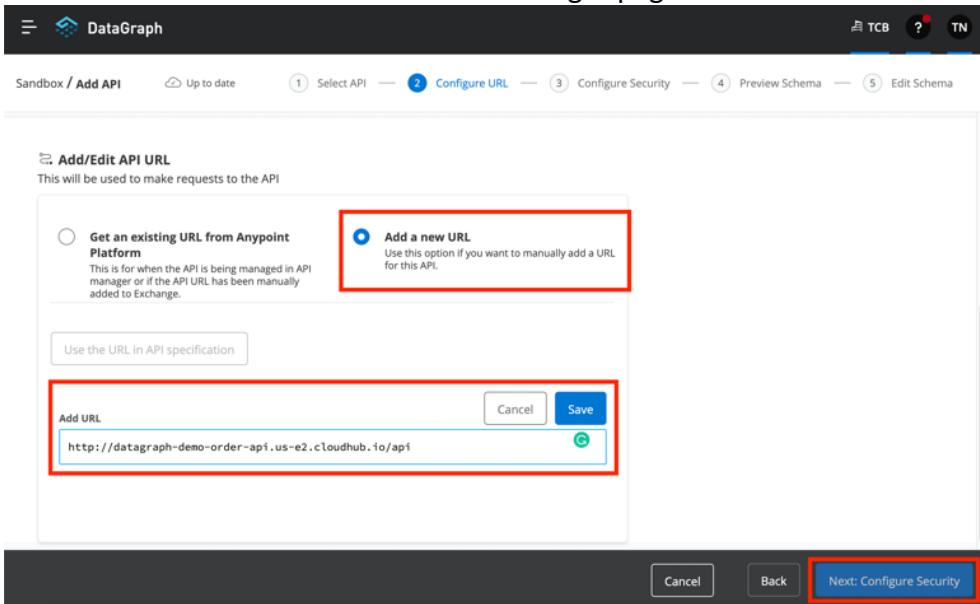
This screenshot shows the 'Add API' wizard at step 1: Select API. The top navigation bar shows 'Sandbox / Add API' and the steps: 1. Select API, 2. Configure URL, 3. Configure Security, 4. Preview Schema, 5. Edit Schema. Below the navigation, there's a search bar 'Search Exchange'. Two API entries are listed: 'Order API' (selected) and 'Product API'. Both entries have a 'Published to Exchange: Jul, 20 2022' note and a 'View in Exchange' link. At the bottom, there are 'Cancel' and 'Next: Configure URL' buttons, with 'Next: Configure URL' highlighted with a red box.

Select appropriate API version then click Confirm.



Scroll down to add API URL.

The domain can be found in Runtime Manager page.



In the next step, configure the Basic authentication when calling the Order API. The username and password are the same as when we configured the policy for the Order API in part 1.

Sandbox / Add API   Up to date   1 Select API — 2 Configure URL — 3 Configure Security — 4 Preview Schema — 5 Edit Schema

**Order API**

**Authentication**  
Select the authentication policy and provide credentials associated with this API's GET endpoints

Select authentication policy  
Supported authentication policy types are limited to those listed below. [Learn more](#)

Basic Auth

Username: orderapi   Password: ..... Show

About this API

[View API specification](#)   [View API documentation](#)

Cancel   Back   **Next: Preview Schema**

Click Next.

Sandbox / Add API   Up to date   1 Select API — 2 Configure URL — 3 Configure Security — 4 Preview Schema — 5 Edit Schema

**Order API**

**API schema preview mode**

**Query**   QueryType  
Type description ▾

Search API Schema...

All types in this API schema

Operation type

**Query**   Mutation

Level 1 types

Order

Query methods

Search methods...

Method name

> orders (offset:Int!, limit:Int!): [Order]  
> ordersByOrderNumber (orderNumber:String!): Order  
> ordersCustomersByCustomerNumber (customerNumber:String!, offset:Int!, limit:Int!): [Order]

Cancel   Back   **Next: Edit Schema**

Select the **OrderProduct** object type then switch its Desired state value to **Visible**.

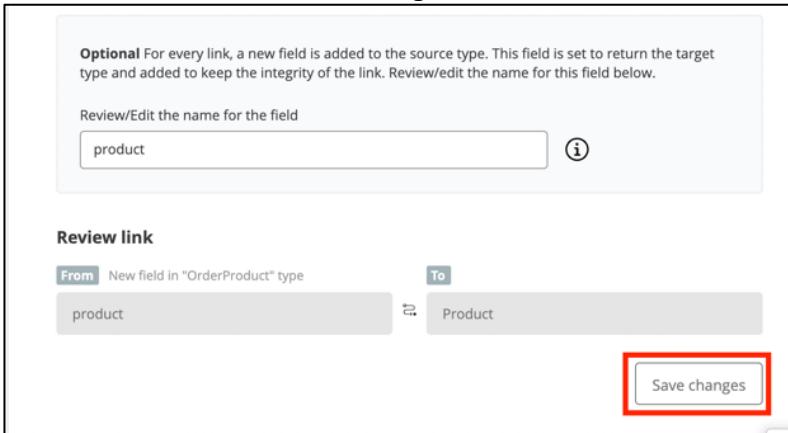
The screenshot shows the DataGraph interface for editing an API schema. On the left, there's a sidebar with sections like 'API schema', 'Mutation', 'Level 1 types', 'Order', 'Nested types' (which contains 'OrderProduct'), 'Input types', 'OrderInput', and 'OrderProductInput'. The 'OrderProduct' item in the 'Nested types' section is highlighted with a red box. In the main panel, the 'OrderProduct' type is selected. At the top right, there's a 'Desired state' toggle switch, which is set to 'Visible' and also highlighted with a red box. Below the type name, there's a 'Type name' field with 'OrderProduct' and a 'Rename Type' button. Under the 'Fields' section, there's a table with one row: 'Field name: price', 'Returned dataType: Float', and 'Desired state: Visible'. At the bottom right of the main panel, there are 'Cancel', 'Back', and 'Next: Add to unified schema' buttons.

Scroll to the **Link to another type** pane, and in **Select the type you want to link to (Target)**, select **Product**.

For the foreign key field, select `productCode (String)`

The screenshot shows the 'Link to another type' configuration pane. On the left, it lists 'Source (Current type)' as 'OrderProduct' and 'Select the field that will be the foreign key in the source' as 'productCode (String)'. To the right, there's a dropdown menu titled 'Select the type you want to link to (Target)' with 'Product' selected. Below this, it says 'Primary key for the selected type above productCode (String!)'. At the bottom, there are 'Cancel', 'Back', and 'Next: Add to unified schema' buttons.

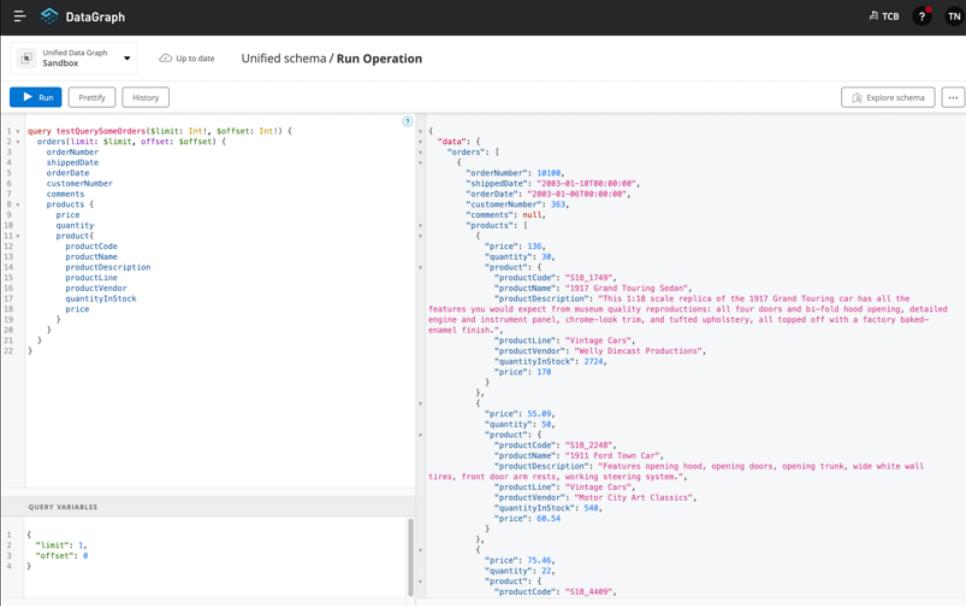
Scroll down and click Save changes.



Click Next: Add to unified schema.

The screenshot shows the 'Next: Add to unified schema' step in the DataGraph interface. The 'OrderProduct' type is selected in the sidebar. A red box highlights the 'Next: Add to unified schema' button at the bottom right of the screen.

Wait for a while for the update to complete, then click Run Operation to try calling the DataGraph API.



The screenshot shows the DataGraph interface with the following details:

- Top Bar:** Shows "DataGraph", "Sandbox", "Up to date", "Unified schema / Run Operation", and "Run" (button).
- Left Panel (Query Editor):** Displays a GraphQL query named "testQuerySomeOrders". The query includes variables for limit (1) and offset (0). It selects orders with specific fields: orderNumber, shippedDate, orderDate, customerNumber, comments, products (with price, quantity, productCode, productName, productDescription, productLine, productVendor, quantityInStock, price), and shippingAddress (with addressLine1, city, state, zip, country).
- Right Panel (Results):** Displays the JSON response from the query. The response includes a "data" object with an "orders" array containing two items. Item 1 is for order number 18108, shipped on 2003-01-01T00:00:00Z, ordered on 2003-01-06T00:00:00Z, and shipped to customer number 363. Item 2 is for order number 18109, shipped on 2003-01-01T00:00:00Z, ordered on 2003-01-06T00:00:00Z, and shipped to customer number 363. Both orders have a comment of "null". Each order has a "products" array containing one item. The first product is a vintage car with code 518\_3749, vendor Welly Diecast Productions, and price \$18.99. The second product is a vintage town car with code 518\_2248, vendor City Art Classics, and price \$55.89. Both products have a quantity of 50 and a quantityInStock of 549.
- Bottom Panel (Variables):** Shows the query variables: {"limit": 1, "offset": 0}.

## 5. Play with DataGraph

### Scenario 1:

Query list of Products

### Scenario 2:

Query Product by Product Code

### Scenario 3:

Query list of Orders

### Scenario 4:

Query Order by Order Number

### Scenario 5:

Query Order by Customer Number

### Scenario 6:

Create an Order, then check the Order just created

### Scenario 7:

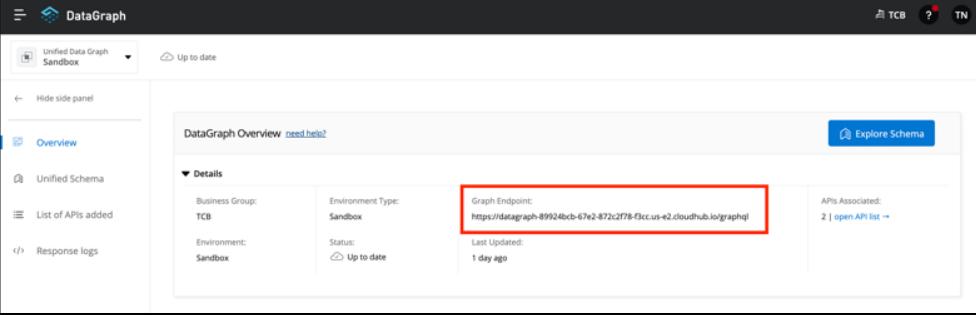
Query Customer information and related Orders by Customer Number

*Note: All Scenario's solutions can be found in **hands-on-solution.txt** file*

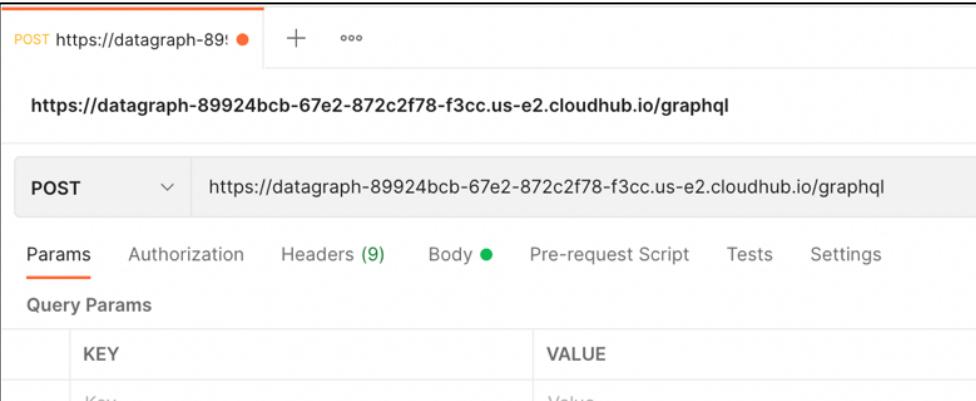
## 6. Call DataGraph API from client side

### a. Call DataGraph API from Postman

Copy the Graph endpoint from DataGraph → Overview then paste to Postman.



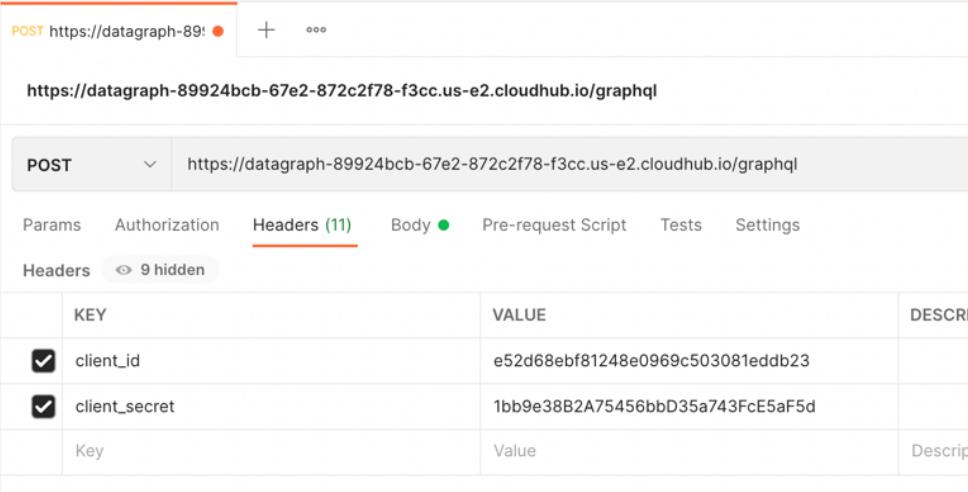
The screenshot shows the DataGraph interface with the 'Overview' tab selected. In the 'Details' section, the 'Graph Endpoint' field is highlighted with a red box, containing the URL: `https://datagraph-89924bcb-67e2-872c2f78-f3cc.us-e2.cloudhub.io/graphql`.



The Postman interface shows a POST request to the copied URL: `https://datagraph-89924bcb-67e2-872c2f78-f3cc.us-e2.cloudhub.io/graphql`. The 'Params' tab is selected, showing a single entry 'Key' under 'Query Params'.

Switch to Headers tab and add 2 headers: `client_id` and `client_secret`.

The value of `client_id` and `client_secret` can be found in Exchange → My application → Name of your application.



The Postman interface shows the 'Headers' tab selected, containing two entries: `client_id` with value `e52d68ebf81248e0969c503081eddb23` and `client_secret` with value `1bb9e38B2A75456bbD35a743FcE5aF5d`.

Client ID: e52d68ebf81248e0969c503081eddb23  
 Client Secret: ..... Show

Application URL: -  
 Redirect URIs: -

Sandbox

- DataGraph
- TCB

In Body tab, select GraphQL. Make sure schema is fetched.

Write some query and variables then click Send

POST https://datagraph-89:443/

https://datagraph-89924bcb-67e2-872c2f78-f3cc.us-e2.cloudhub.io/graphql

POST https://datagraph-89924bcb-67e2-872c2f78-f3cc.us-e2.cloudhub.io/graphql

Params Authorization Headers (11) Body \* Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Auto-fetch Schema Fetched

QUERY

```

1 query queryProductsByCode($code: String!) {
2   product: productsByProductCode(productCode: $code) {
3     productCode
4     productName
5     productLine
6     productVendor
7     quantityInStock
8     suggestedPrice: price
9     productDescription
10   }
11 }
```

GRAPHQL VARIABLES

```

1 $code: "S700_2610"
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 997 ms Size: 719 B Save Response

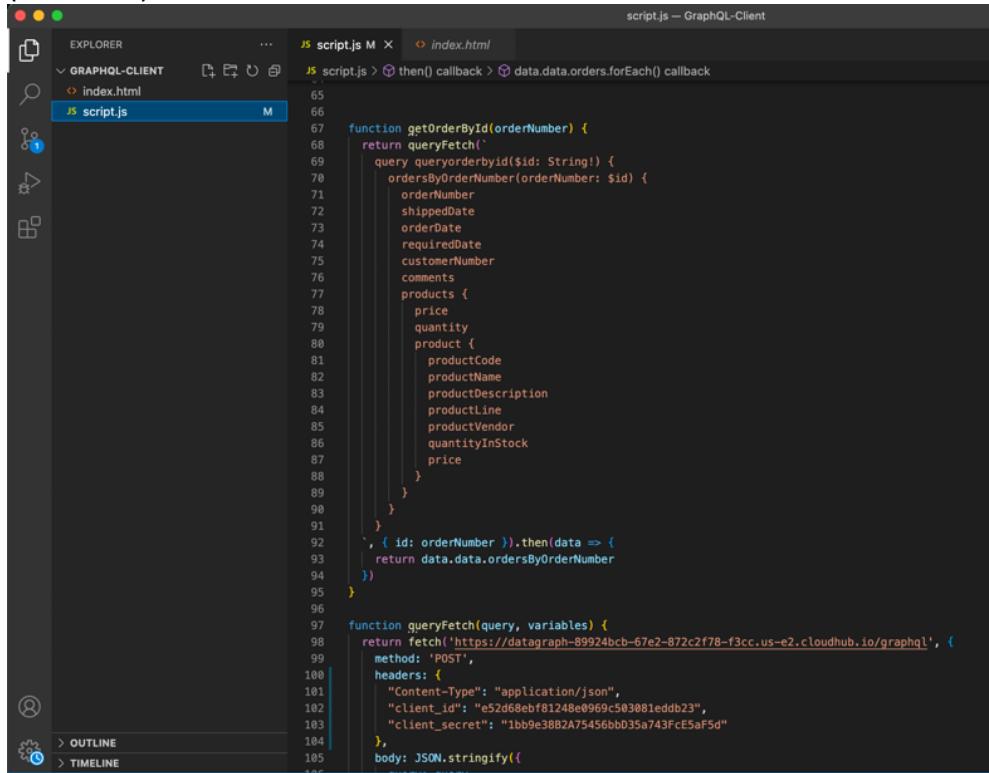
```

1 {
2   "data": {
3     "product": {
4       "productCode": "S700_2610",
5       "productName": "The USS Constitution Ship",
6       "productLine": "Ships",
7       "productVendor": "Red Start Diecast",
8       "quantityInStock": 7083.0,
9       "suggestedPrice": 72.28,
10      "productDescription": "All wood with canvas sails. Measures 31 1/2\" Length x 22 3/8\" High x 8 1/4\" Width.  
Extras include 4 boats on deck, sea sprite on bow, anchors, copper railing, pilot houses, etc."
11    }
12 }
```

## b. Call DataGraph API from JavaScript

*Note: the HTML and JavaScript code can be found in **GraphQL-Client** folder.*

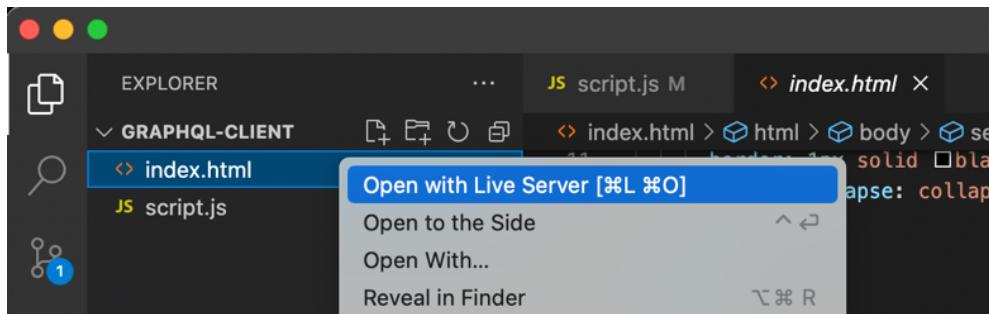
Open folder in some IDE, change the DataGraph API endpoint, client\_id and client\_secret (if needed). Then start the server



The screenshot shows the VS Code interface with the 'script.js' file open in the editor. The code is a JavaScript script that uses the fetch API to query a GraphQL endpoint. It defines two functions: 'getOrderById' which takes an order number and returns the order details, and 'queryFetch' which takes a query and variables, then sends a POST request to the DataGraph API endpoint. The code includes placeholder values for 'client\_id' and 'client\_secret'.

```
function getOrderById(orderNumber) {
  return queryFetch(`query queryorderbyid($id: String!) {
    ordersByOrderNumber(orderNumber: $id) {
      orderNumber
      shippedDate
      orderDate
      requiredDate
      customerNumber
      comments
      products {
        price
        quantity
        product {
          productCode
          productName
          productDescription
          productLine
          productVendor
          quantityInStock
          price
        }
      }
    }
  }, { id: orderNumber }).then(data => {
  return data.data.ordersByOrderNumber
})
}

function queryFetch(query, variables) {
  return fetch('https://datagraph-89924bcb-67e2-872c2f78-f3cc.us-e2.cloudhub.io/graphql', {
    method: 'POST',
    headers: {
      "Content-Type": "application/json",
      "client_id": "e52d68ebf81248e0969c503081eddb23",
      "client_secret": "1bb9e3882a75456bb035a743FcE5aF5d"
    },
    body: JSON.stringify({
      query: query,
      variables: variables
    })
  })
}
```



Select an Order Number and check the result.

<input type="button" value="Select an Order ▾"/>
<b>Order Number</b>
<b>Shipped Date</b>
<b>Order Date</b>
<b>Required Date</b>
<b>Customer Number</b>
<b>Comments</b>
<b>List of products</b>

<input type="text" value="10107"/> <input type="button" value="▼"/>						
<b>Order Number</b>	10107					
<b>Shipped Date</b>	2003-02-26T00:00:00					
<b>Order Date</b>	2003-02-24T00:00:00					
<b>Required Date</b>	2003-03-03T00:00:00					
<b>Customer Number</b>	131					
<b>Comments</b>	Difficult to negotiate with customer. We need more marketing materials					
<b>List of products</b>						
Product Code	Product Name	Product Description	Product Line	Product Vendor	Quantity	Price
S10_1678	1969 Harley Davidson Ultimate Chopper	This replica features working kickstand, front suspension, gear-shift lever, footbrake lever, drive chain, wheels and steering. All parts are particularly delicate due to their precise scale and require special care and attention.	Motorcycles	Min Lin Diecast	30	81.35
S10_2016	1996 Moto Guzzi 1100i	Official Moto Guzzi logos and insignias, saddle bags located on side of motorcycle, detailed engine, working steering, working suspension, two leather seats, luggage rack, dual exhaust pipes, small saddle bag located on handle bars, two-tone paint with chrome accents, superior die-cast detail , rotating wheels , working kick stand, diecast metal with plastic parts and baked enamel finish.	Motorcycles	Highway 66 Mini Classics	39	105.86
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Model features, official Harley Davidson logos and insignias, detachable rear wheelie bar, heavy diecast metal with resin parts, authentic multi-color tampon-printed graphics, separate engine drive belts, free-turning front fork, rotating tires and rear racing slick, certificate of authenticity, detailed engine, display stand , precision diecast replica, baked enamel finish, 1:10 scale model, removable fender, seat and tank cover piece for displaying the superior detail of the v-twin engine	Motorcycles	Red Start Diecast	27	172.36
		Official logos and insignias, saddle bags located on				