TGH

Making Integrations Simpler

boomi Partner

Get Details of Deployed API's Using GraphQL

Author

Sharukh Khan Darvesh

Contents:

Introduction:

- GraphQL is a query language for API's and a runtime for executing those queries against our requested data.
- GraphQL provides a more efficient, powerful, and flexible alternative to the traditional REST API.

Advantages:

- With GraphQL, we can request only the data that we need, and nothing more. This helps in reducing over-fetching or under-fetching of data, which is a common issue with REST API's.
- GraphQL queries are hierarchical and closely match the structure of the data they retrieve. This makes it easy to understand and predict the shape of the response.
- GraphQL typically uses a single endpoint for all queries and mutations.
- GraphQL API's are defined by a schema that specifies the types of data that can be queried.

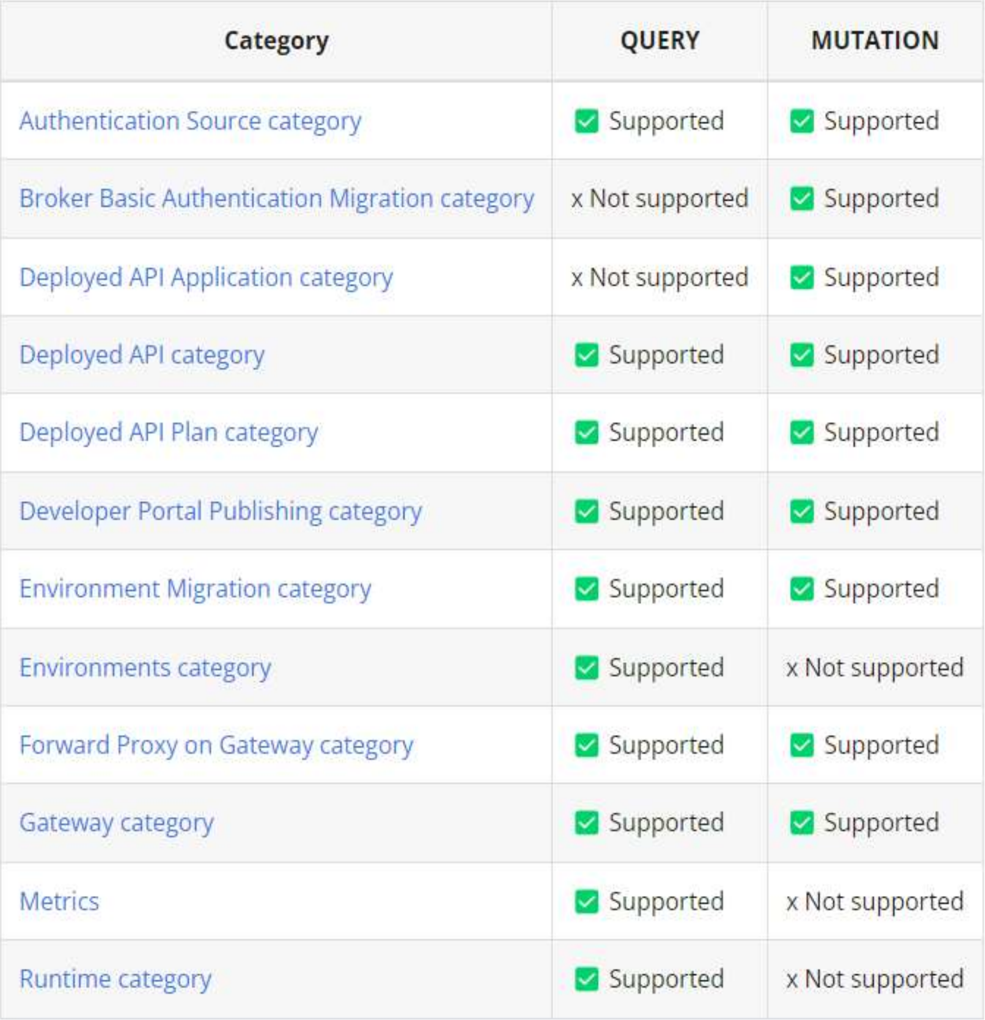
Functions:

- Functions have arguments that are passed in when a function is executed.
- Functions are implemented via HTTP Post of a text payload containing the function call to a server.
- GraphQL has two primary Functions:

i. **Query** - It is a function for reading the data from the server and displaying it. Like any database query, this will connect to the GraphQL API endpoint and retrieve the necessary data.

ii. **Mutation** - It is an update to server-side data, means that you can change values with new values. A Mutation can be used to insert, update, or delete data.

GraphQL Categories:



Category	QUERY	MUTATION
Authentication Source category	☑ Supported	☑ Supported
Broker Basic Authentication Migration category	x Not supported	☑ Supported
Deployed API Application category	x Not supported	☑ Supported
Deployed API category	☑ Supported	☑ Supported
Deployed API Plan category	☑ Supported	☑ Supported
Developer Portal Publishing category	☑ Supported	☑ Supported
Environment Migration category	☑ Supported	☑ Supported
Environments category	☑ Supported	x Not supported
Forward Proxy on Gateway category	☑ Supported	☑ Supported
Gateway category	☑ Supported	☑ Supported
Metrics	☑ Supported	x Not supported
Runtime category	☑ Supported	x Not supported

Metrics:

- API Management Metrics services collect the metadata of API calls through the API Gateway or the Web Services Server and allows you to retrieve the data using a GraphQL client.
- Metrics services return information such as the count of calls per API, successful API requests, popular authentication types for APIs, and the quota status of a subscribed API.
- Boomi retains metrics data for 540 days.

Pre-Requisites To Enable Metrics Data Collection:

- By default, metrics services are disabled. You can enable the services in two ways:

1. Enable in Settings > Account > Features > API Metrics Access.
2. Enable on the API Gateway, Molecules, and Clouds in Properties > Settings & Configuration > Properties > Advanced > Capture API Metrics.

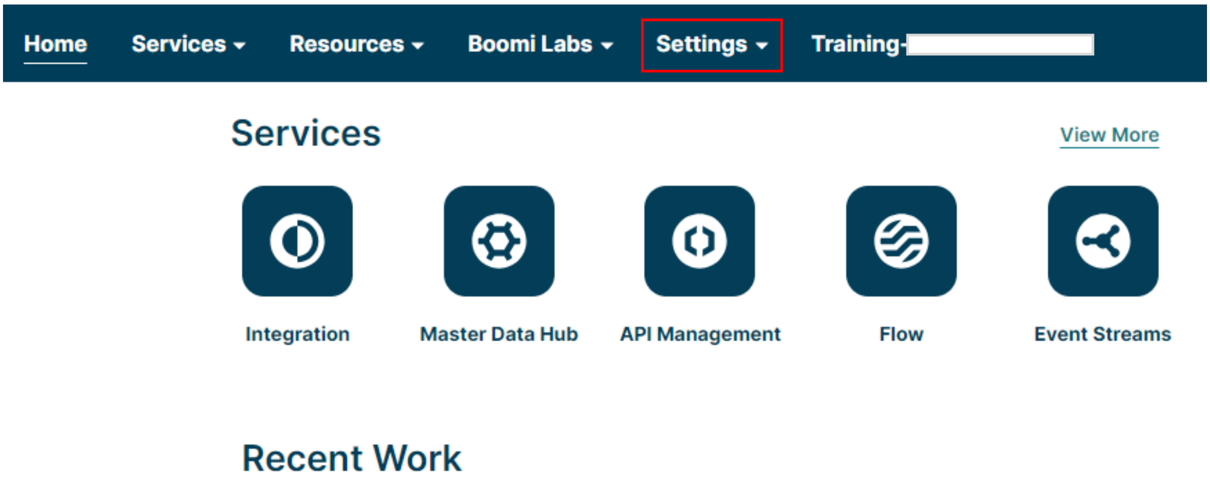
- You must restart each Gateway after enabling or disabling API Metrics.

Practical Implementation in Boomi:

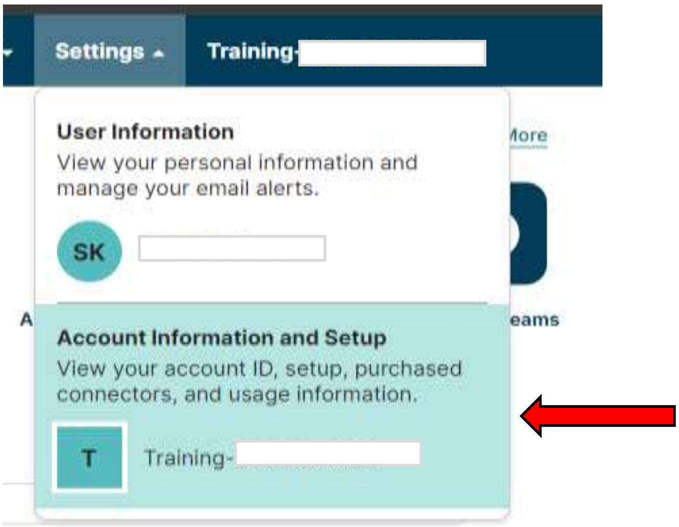
Let us see how to get details of deployed API's using GraphQL.

Step 01: First we need to enable API Metrics. Click on Settings.

Enable in Settings > Account Information & Setup > Features > API Metrics Access.



Step 02: click on Account Information & Setup, a new tab will open.



User Information

View your personal information and manage your email alerts.

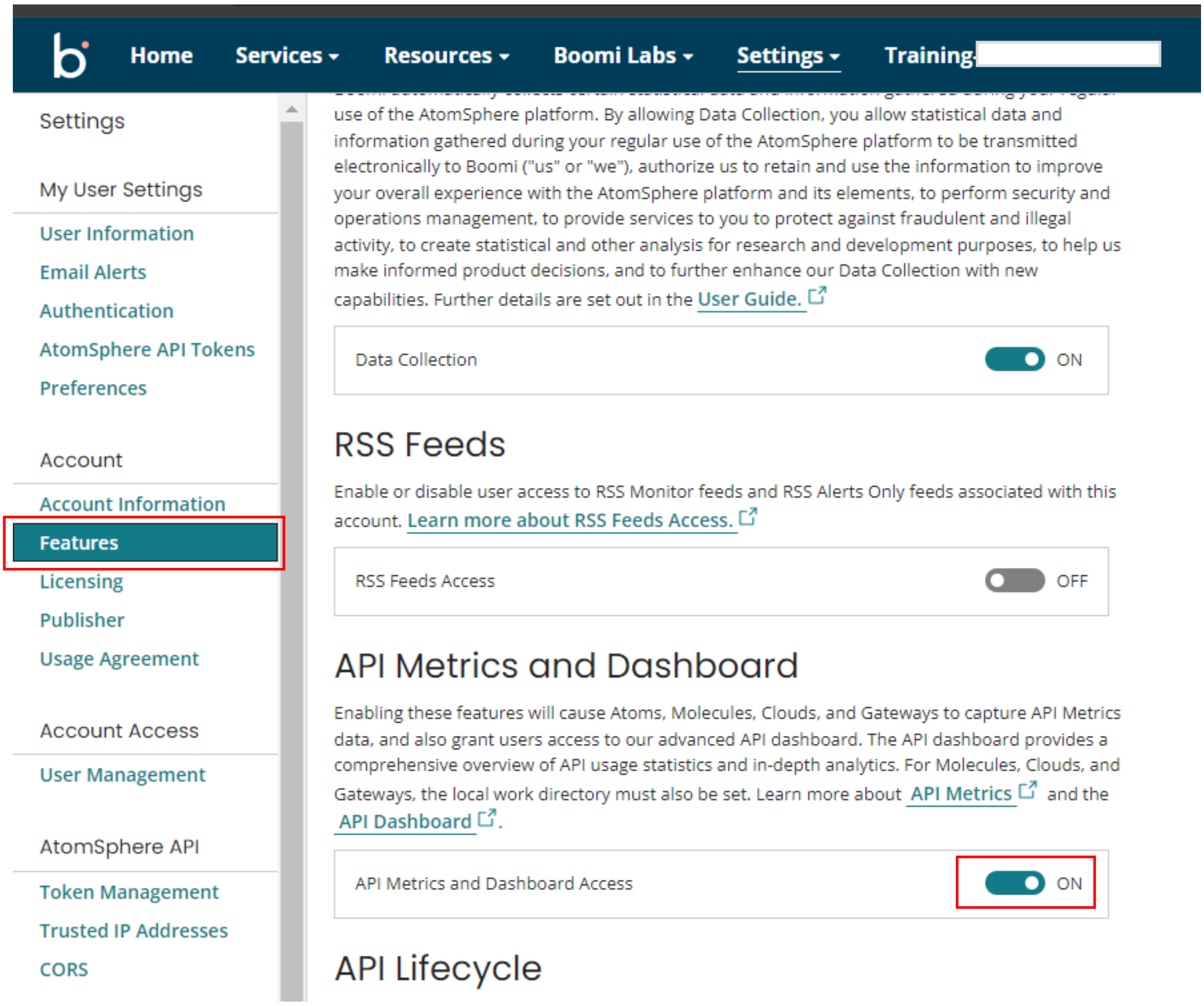
SK [redacted]

Account Information and Setup

View your account ID, setup, purchased connectors, and usage information.

T Training-[redacted]

Step 03: Click on features and scroll down. You will get to see a toggle API Metrics and Dashboard, turn ON the toggle so that the metrics will be captured.



Settings

My User Settings

- User Information
- Email Alerts
- Authentication
- AtomSphere API Tokens
- Preferences

Account

- Account Information
- Features
- Licensing
- Publisher
- Usage Agreement

Account Access

- User Management

AtomSphere API

- Token Management
- Trusted IP Addresses
- CORS

use of the AtomSphere platform. By allowing Data Collection, you allow statistical data and information gathered during your regular use of the AtomSphere platform to be transmitted electronically to Boomi ("us" or "we"), authorize us to retain and use the information to improve your overall experience with the AtomSphere platform and its elements, to perform security and operations management, to provide services to you to protect against fraudulent and illegal activity, to create statistical and other analysis for research and development purposes, to help us make informed product decisions, and to further enhance our Data Collection with new capabilities. Further details are set out in the User Guide.

Data Collection ON

RSS Feeds

Enable or disable user access to RSS Monitor feeds and RSS Alerts Only feeds associated with this account. Learn more about RSS Feeds Access.

RSS Feeds Access OFF

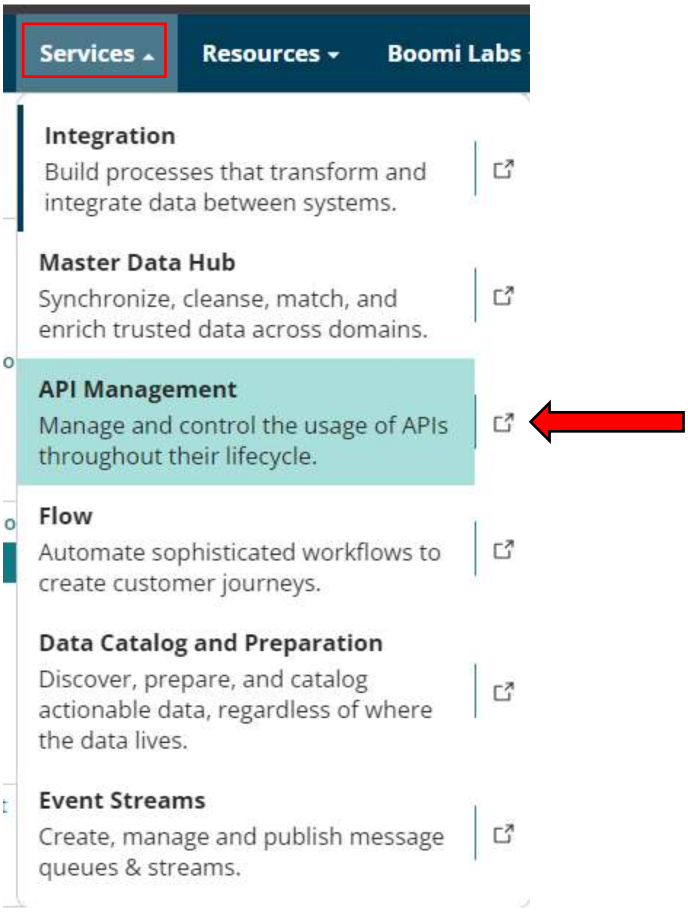
API Metrics and Dashboard

Enabling these features will cause Atoms, Molecules, Clouds, and Gateways to capture API Metrics data, and also grant users access to our advanced API dashboard. The API dashboard provides a comprehensive overview of API usage statistics and in-depth analytics. For Molecules, Clouds, and Gateways, the local work directory must also be set. Learn more about API Metrics and the API Dashboard.

API Metrics and Dashboard Access ON

API Lifecycle

Step 04: Go to services and select API Management.



Integration
Build processes that transform and integrate data between systems.

Master Data Hub
Synchronize, cleanse, match, and enrich trusted data across domains.

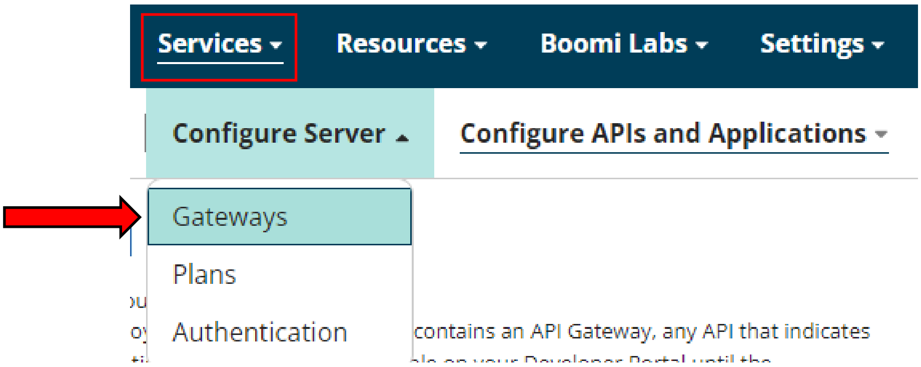
API Management
Manage and control the usage of APIs throughout their lifecycle.

Flow
Automate sophisticated workflows to create customer journeys.

Data Catalog and Preparation
Discover, prepare, and catalog actionable data, regardless of where the data lives.

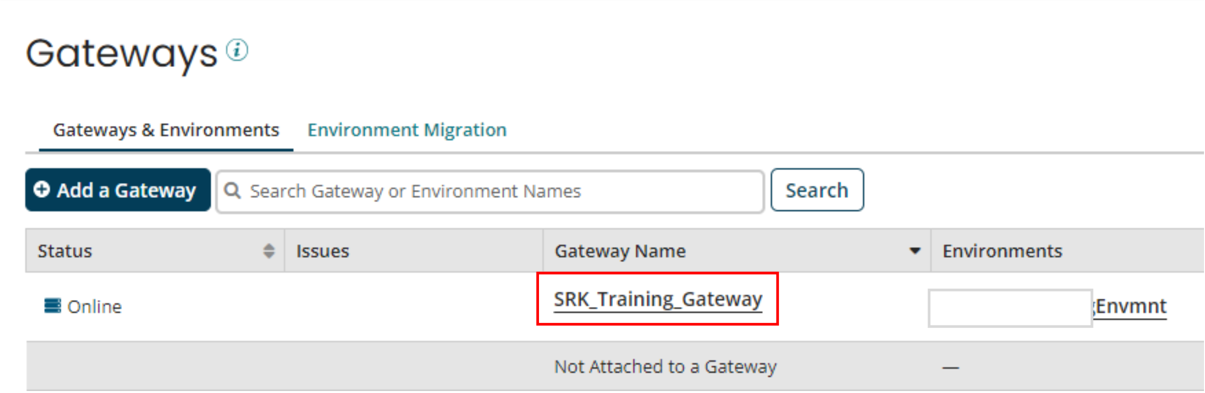
Event Streams
Create, manage and publish message queues & streams.

Step 05: A new tab will open, click on Configure Server and select Gateways.



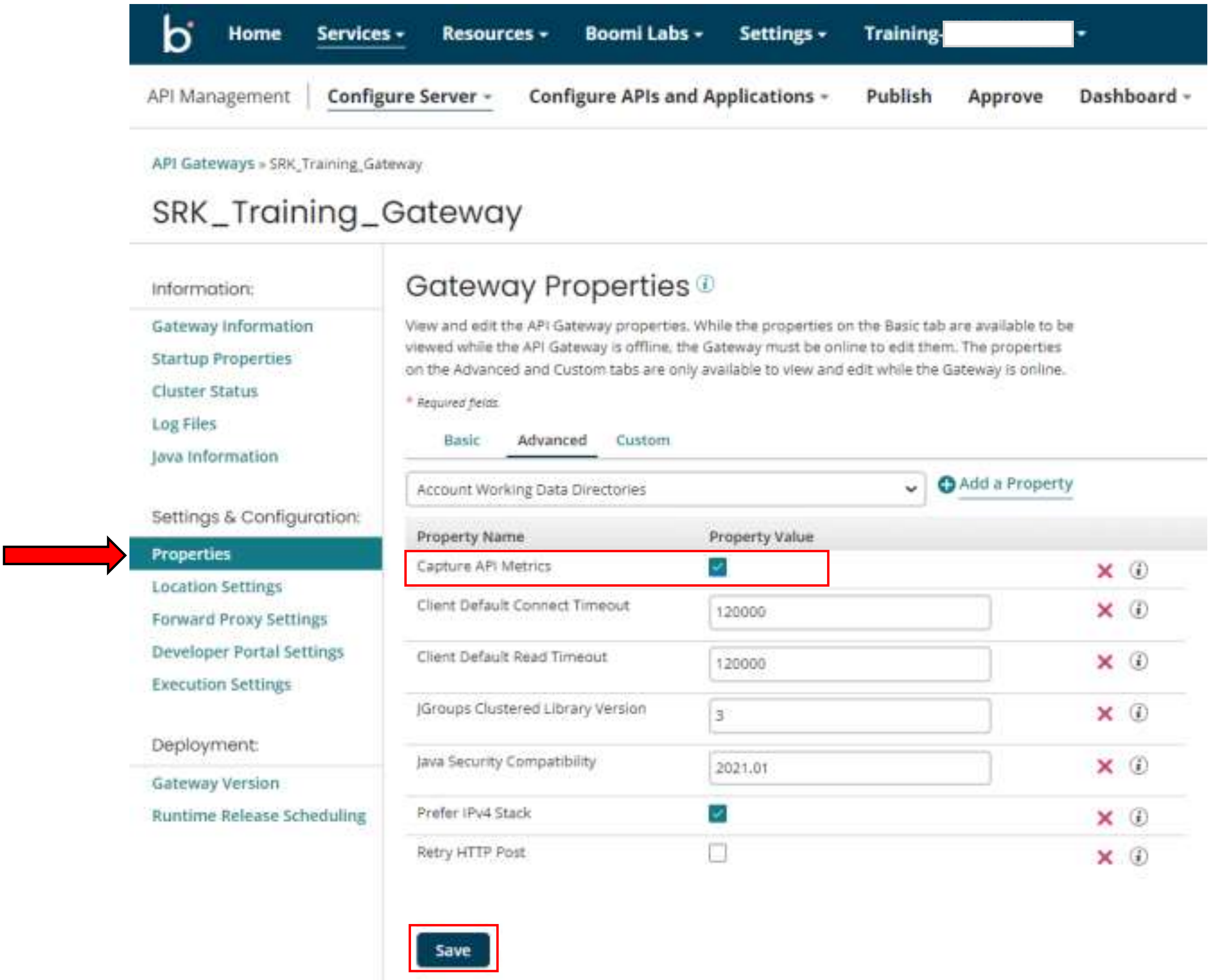
contains an API Gateway, any API that indicates

Step 06: Click on your gateway.



Status	Issues	Gateway Name	Environments
Online		SRK_Training_Gateway	Envmnt
		Not Attached to a Gateway	—

Step 07: A new tab will appear there we need to select properties and inside it advanced options check the checkbox of capture API metrics and save it. Then you need to restart your runtime.

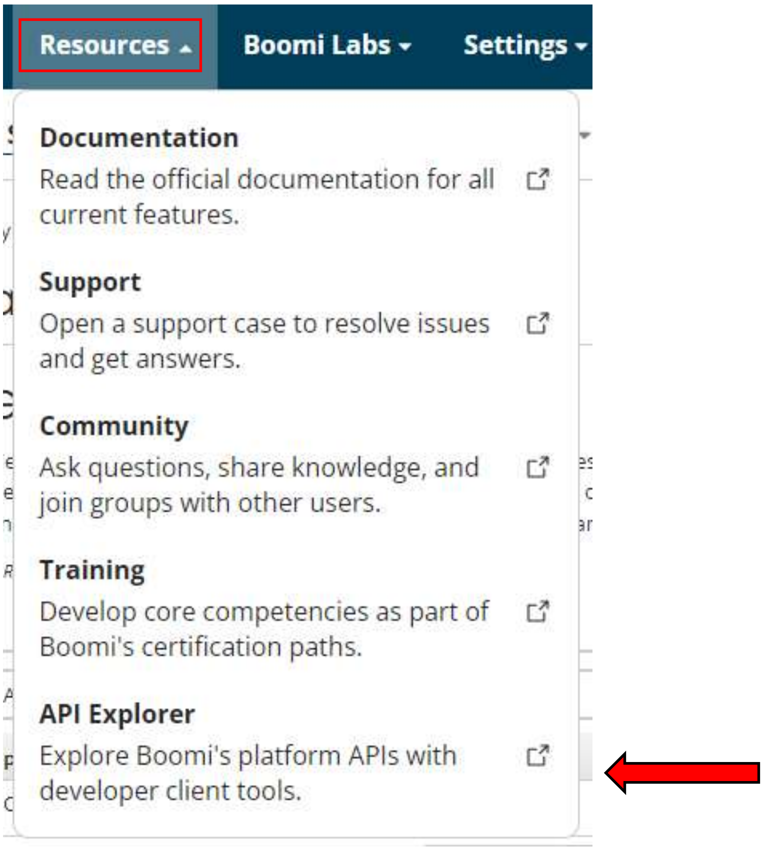


View and edit the API Gateway properties. While the properties on the Basic tab are available to be viewed while the API Gateway is offline, the Gateway must be online to edit them. The properties on the Advanced and Custom tabs are only available to view and edit while the Gateway is online.

* Required fields

Property Name	Property Value
Capture API Metrics	☑
Client Default Connect Timeout	120000
Client Default Read Timeout	120000
JGroups Clustered Library Version	3
Java Security Compatibility	2021.01
Prefer IPv4 Stack	☑
Retry HTTP Post	☐

Step 08: Once we restart the runtime engine we can see API Explorer in the Resources.



Documentation
Read the official documentation for all current features.

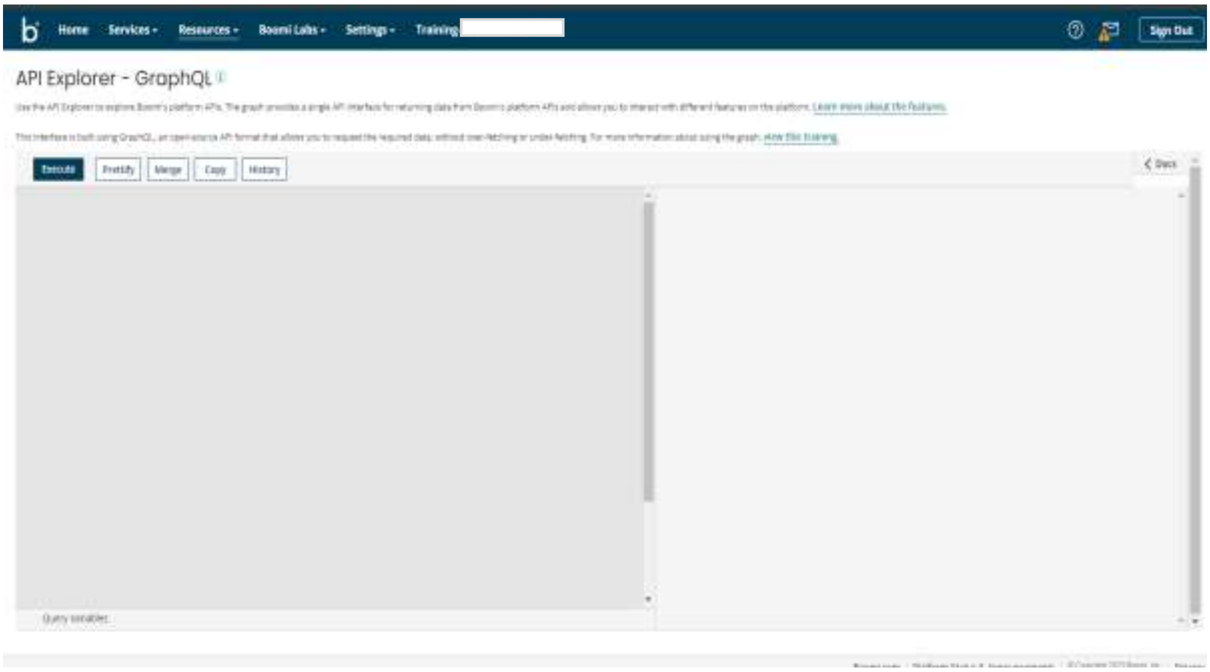
Support
Open a support case to resolve issues and get answers.

Community
Ask questions, share knowledge, and join groups with other users.

Training
Develop core competencies as part of Boomi's certification paths.

API Explorer
Explore Boomi's platform APIs with developer client tools.

Step 09: Click on it, a new tab will appear as shown below.



API Explorer - GraphQL

Use the API Explorer to explore Boomi's platform APIs. The graph provides a single API interface for returning data from Boomi's platform APIs and allows you to interact with different features on the platform. Learn more about the features.

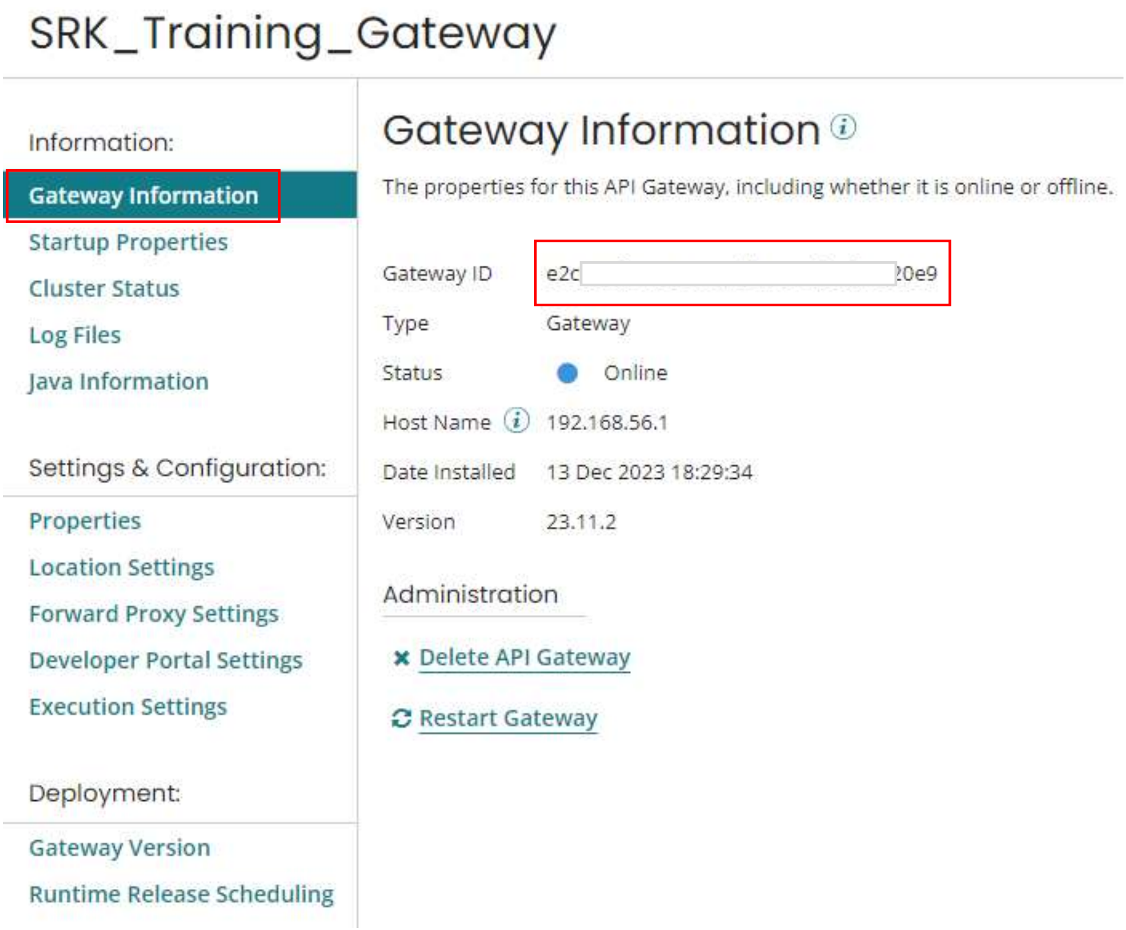
This interface is built using GraphQL, an open-source API format that allows you to request the required data, without over-fetching or under-fetching. For more information about using the graph, view this training.

Step 10: We will going to query the details of deployed API using “deployedApiCallDetail()” method where we pass the Start Date, End Date and Gateway ID.

```
query{
 deployedApiCallDetail(input:
 {
  requestTsStart: "2023-12-16"
  requestTsEnd: "2023-12-18"
  filter: "gateway.id = 'e2[redacted]e9'"
 }
 )
{
  data {
    requestTs
    atom { id }
    gateway { id }
    account { id }
    deployedApi { id }
    authType
    status
    method
    requestUrl
    client { remoteAddress }
    client { userAgent }
    user { username }
    transactionId
    quotaCount
    rateReset
    bytesReceived
    responseDuration
    backendDuration
    bytesSent
    rateCount
  }
  nextRequestTs
  nextTransactionId
}
}
```

Step 11: Paste the above mentioned query on the left side of API Explorer.

Step 12: To get the Gateway ID, repeat the step 04, 05 & 06. Later click on Gateway Information, there we will get Gateway ID.

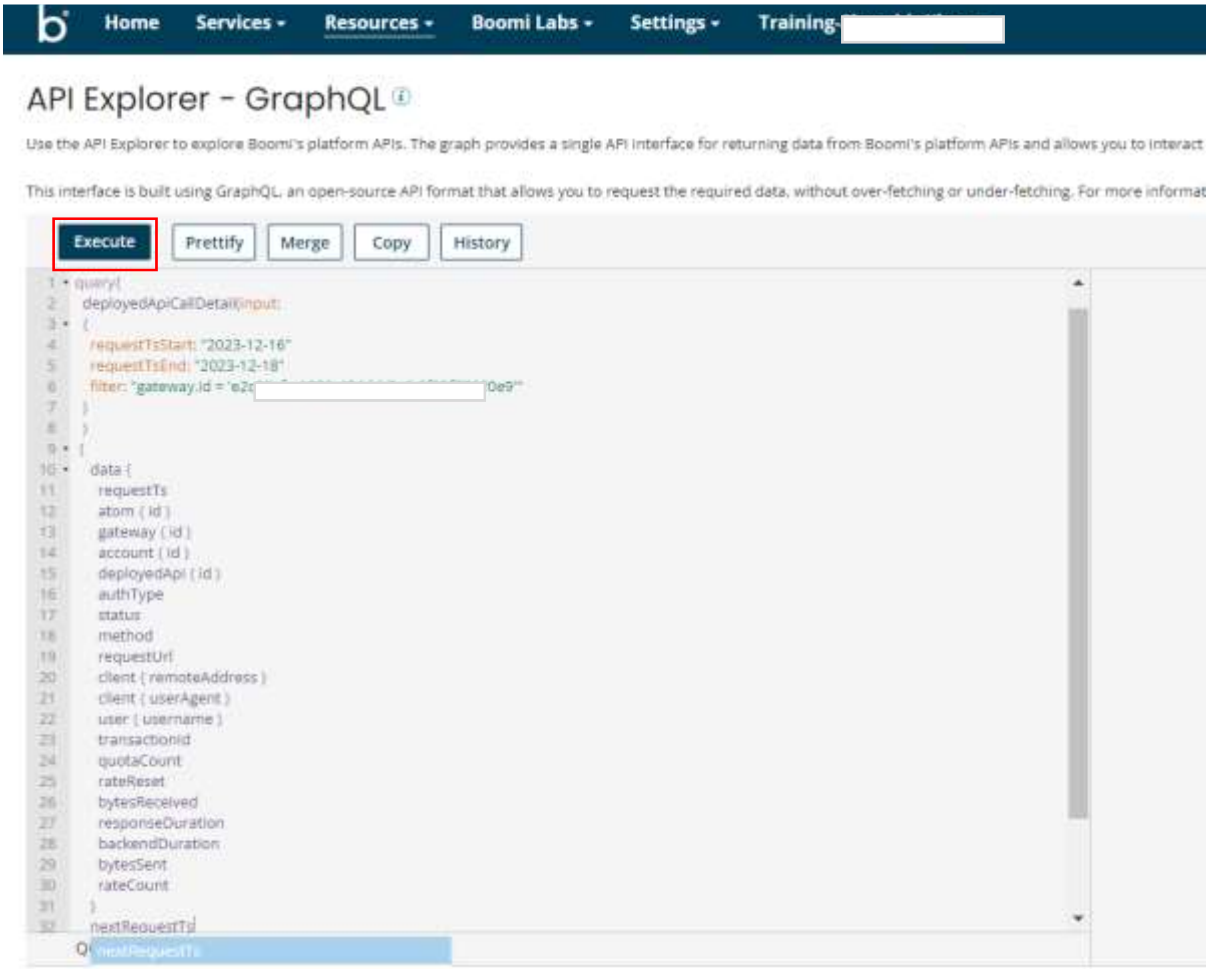


Gateway Information

The properties for this API Gateway, including whether it is online or offline.

Gateway ID	e2c[redacted]0e9
Type	Gateway
Status	● Online
Host Name	192.168.56.1
Date Installed	13 Dec 2023 18:29:34
Version	23.11.2

Step 13: After pasting the Gateway Id it looks as shown below. And click on Execute button to execute the specified query.



API Explorer - GraphQL

Use the API Explorer to explore Boomi's platform APIs. The graph provides a single API interface for returning data from Boomi's platform APIs and allows you to interact

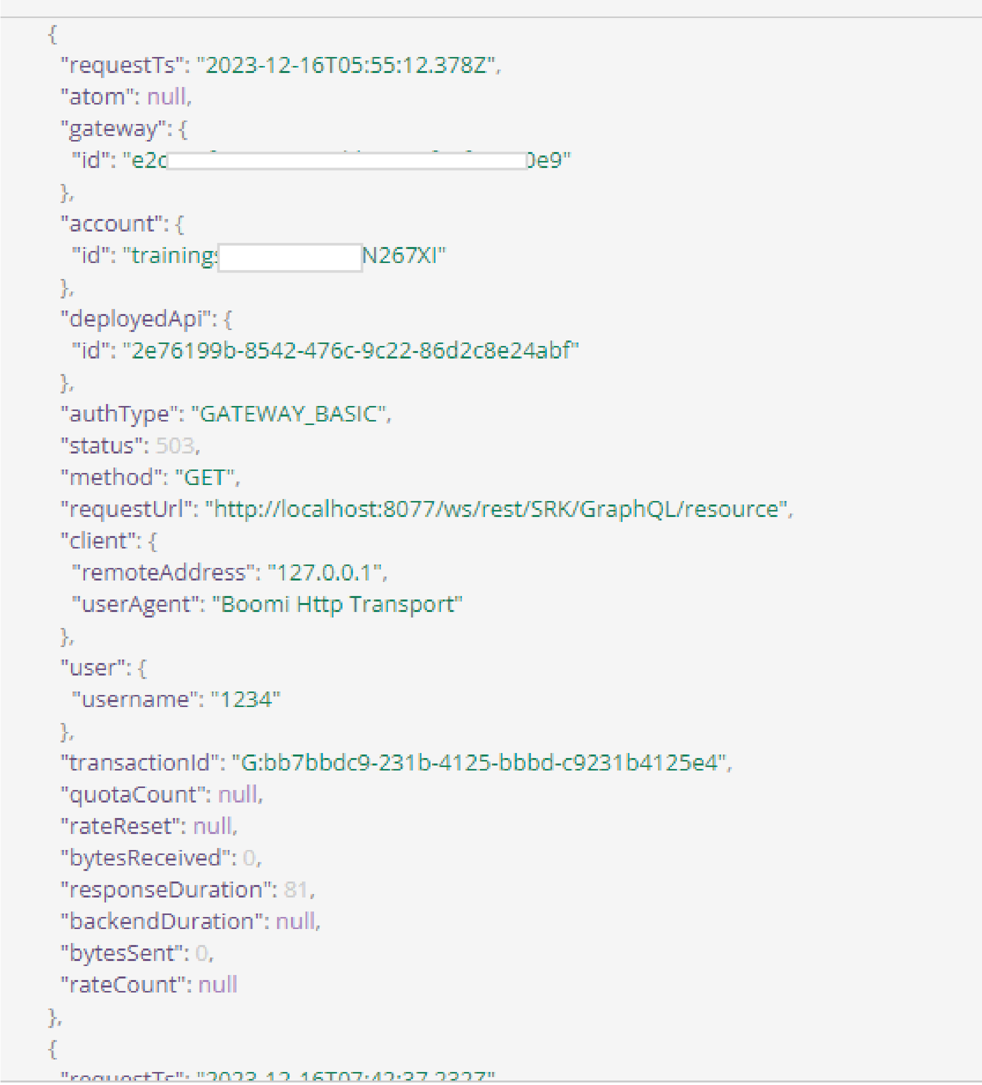
This interface is built using GraphQL, an open-source API format that allows you to request the required data, without over-fetching or under-fetching. For more informat

```
query{
 deployedApiCallDetail(input:
 {
  requestTsStart: "2023-12-16"
  requestTsEnd: "2023-12-18"
  filter: "gateway.id = 'e2c[redacted]0e9'"
  }
 )
{
  data {
   requestTs
   atom { id }
   gateway { id }
   account { id }
   deployedApi { id }
   authType
   status
   method
   requestUrl
   client { remoteAddress }
   client { userAgent }
   user { username }
   transactionId
   quotaCount
   rateReset
   bytesReceived
   responseDuration
   backendDuration
   bytesSent
   rateCount
  }
  nextRequestTs
```

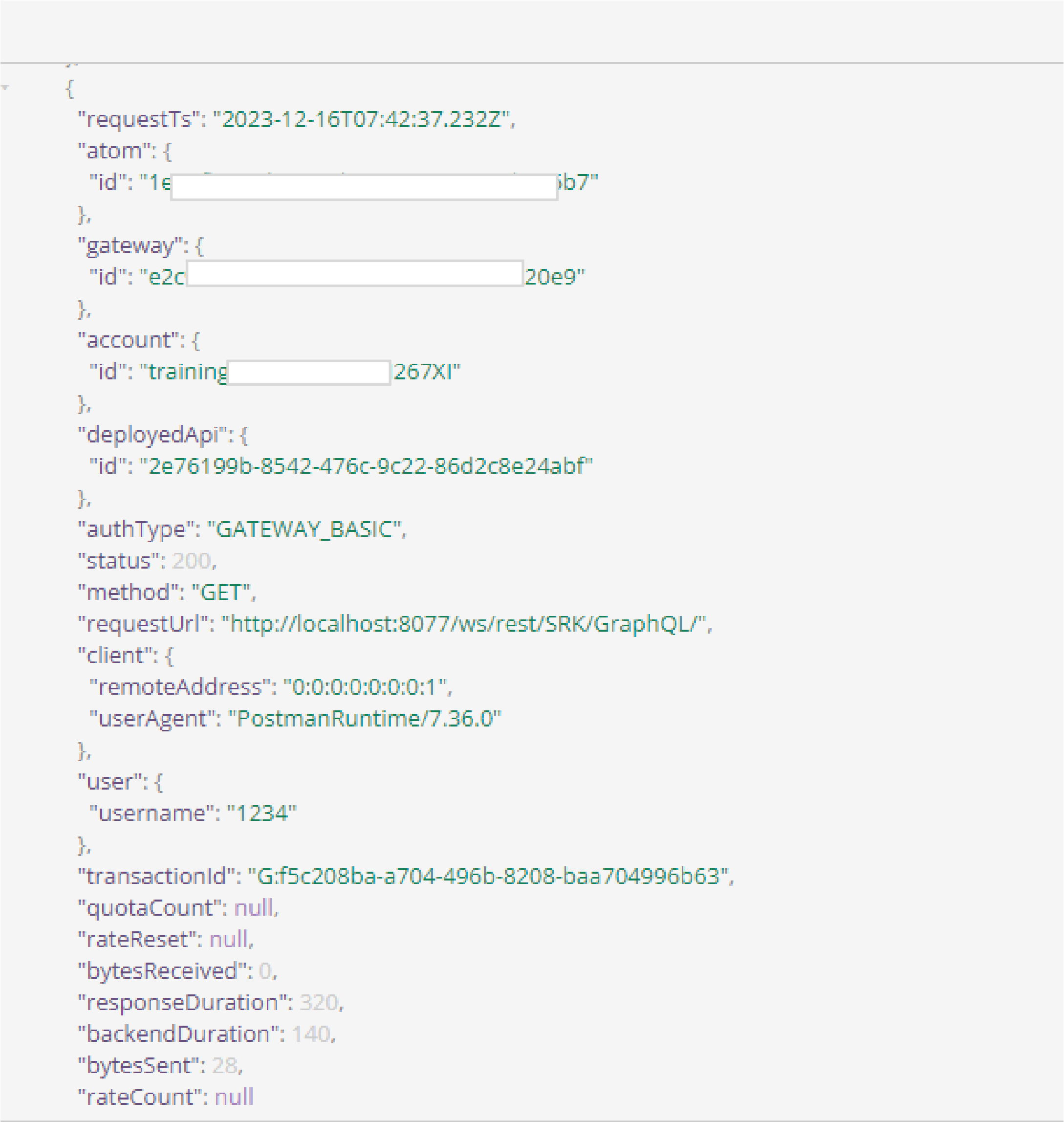
Step 14: You will get the response in right side of API Explorer as shown below.

In the response we can see GatewayID, AccountID, DeployedAPI Id, Auth Type, Status, method, URL, we can also see the username for the Auth type, we can also see the rate limit, message size and Quota Limit, Response time, etc,.

Here in the first picture we can see an 503 error and in the second picture we can see the 200 response.



```
{
  "requestTs": "2023-12-16T05:55:12.378Z",
  "atom": null,
  "gateway": {
    "id": "e2c[redacted]0e9"
  },
  "account": {
    "id": "training[redacted]N267XI"
  },
  "deployedApi": {
    "id": "2e76199b-8542-476c-9c22-86d2c8e24abf"
  },
  "authType": "GATEWAY_BASIC",
  "status": 503,
  "method": "GET",
  "requestUrl": "http://localhost:8077/ws/rest/SRK/GraphQL/resource",
  "client": {
    "remoteAddress": "127.0.0.1",
    "userAgent": "Boomi Http Transport"
  },
  "user": {
    "username": "1234"
  },
  "transactionId": "G:bb7bbdc9-231b-4125-bbbd-c9231b4125e4",
  "quotaCount": null,
  "rateReset": null,
  "bytesReceived": 0,
  "responseDuration": 81,
  "backendDuration": null,
  "bytesSent": 0,
  "rateCount": null
},
{
  "requestTs": "2023-12-16T07:42:37.232Z",
```



```
  },
  {
    "requestTs": "2023-12-16T07:42:37.232Z",
    "atom": {
      "id": "1e[redacted]b7"
    },
    "gateway": {
      "id": "e2c[redacted]20e9"
    },
    "account": {
      "id": "training[redacted]267XI"
    },
    "deployedApi": {
      "id": "2e76199b-8542-476c-9c22-86d2c8e24abf"
    },
    "authType": "GATEWAY_BASIC",
    "status": 200,
    "method": "GET",
    "requestUrl": "http://localhost:8077/ws/rest/SRK/GraphQL/",
    "client": {
      "remoteAddress": "0:0:0:0:0:0:0:1",
      "userAgent": "PostmanRuntime/7.36.0"
    },
    "user": {
      "username": "1234"
    },
    "transactionId": "G:f5c208ba-a704-496b-8208-baa704996b63",
    "quotaCount": null,
    "rateReset": null,
    "bytesReceived": 0,
    "responseDuration": 320,
    "backendDuration": 140,
    "bytesSent": 28,
    "rateCount": null
```

This is how we can see the details of deployed API’s based on the Gateway ID.

References:

- https://help.boomi.com/docs/Atomsphere/API%20Management/Topics/api-GraphQL_API_Management_APIs_18f1a55a-b3d7-4b9e-ab0a-162fc4a67686
- https://help.boomi.com/docs/Atomsphere/API%20Management/Topics/api-metrics_0e0f3adb-2fcb-4af5-bbd1-aee58d2e713f

TGH

Making Integrations Simpler

boomi Partner

TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.

Email address

connect@techygeekhub.com

Phone number

+ 011-40071137
+ 91-8810610395

Our offices

Noida Office
iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office
Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081