

Kong Gateway利用手順書 (1.2.0版)

2024年03月01日
一般社団法人データ社会推進協議会

改版履歴

バージョン	改版内容	公開日
1.0.0	初版 (Kong Gateway 2.8.0 を対象に作成)	2022/07/01
1.1.0	Kong Gateway 3.1.0 向けに手順書を改版	2023/03/01
1.2.0	Kong Gateway 3.4.0 向けに手順書を改版	2024/03/01

目次

1.	はじめに.....	4
1-1.	概要.....	4
1-2.	事前準備.....	4
1-3.	表記方法.....	4
1-3-1.	コマンドの表記方法.....	4
1-3-2.	コマンド入力結果の表記方法.....	4
2.	Kong Gateway の概要.....	5
2-1.	Kong Gateway とは.....	5
2-2.	公式ドキュメント.....	7
3.	利用手順.....	8
3-1.	サービスの登録.....	9
3-2.	ルートの登録.....	10
3-3.	API の利用.....	12
3-4.	サービス、ルートの更新.....	19
3-5.	サービス、ルートの削除.....	23

1. はじめに

1-1. 概要

本書では、Kong Gateway の概要と、その代表的な API (Application Programming Interface) の利用手順を説明する。

1-2. 事前準備

本書の説明においては、Kong Gateway の実行環境が構築済みであり、Kong Gateway が稼働していることを前提とする。Kong Gateway の実行環境の構築方法に関しては、別紙の「Kong Gateway ビルド手順書」及び「Kong Gateway 構築手順書」を参照。

1-3. 表記方法

1-3-1. コマンドの表記方法

(例)

```
# source ~/ENV.sh
```

コマンド入力を表す箇所については、上記のように実線で囲んでいる。
行頭の # はプロンプトであり、入力するのはそれ以降の青い背景色の部分である。

1-3-2. コマンド入力結果の表記方法

(例)

```
HTTP/1.1 201 Created  
Date: Wed, 27 Dec 2023 07:45:51 GMT  
Content-Type: application/json; charset=utf-8  
:
```

コマンド入力結果を表す箇所については、上記のように破線で囲み橙色の背景色で表記している。

2. Kong Gateway の概要

本章では、Kong Gateway の概要及び公式ドキュメントについて説明する。

2-1. Kong Gateway とは

Kong Gateway は、Kong Inc. が提供するソフトウェアで、マイクロサービスおよび分散アーキテクチャ向けに最適化された API ゲートウェイである。オープンソース版と商用版が提供されており、いずれのソフトウェアも API ゲートウェイ機能を有している。

受け付けた Web アクセスをバックエンドの API 実行サーバに転送する API 用リバースプロキシとして動作し、マイクロサービス型の複数の API 実行サーバを統合管理することができる。

図 2-1 に示す通り、Kong Gateway は、API ゲートウェイとして、認証、アクセス制限、レート制限、監視などの機能を提供する。また、プラグイン (Kong Plugin Hub : <https://docs.konghq.com/hub/>) を使用することにより、機能を拡張することができる。

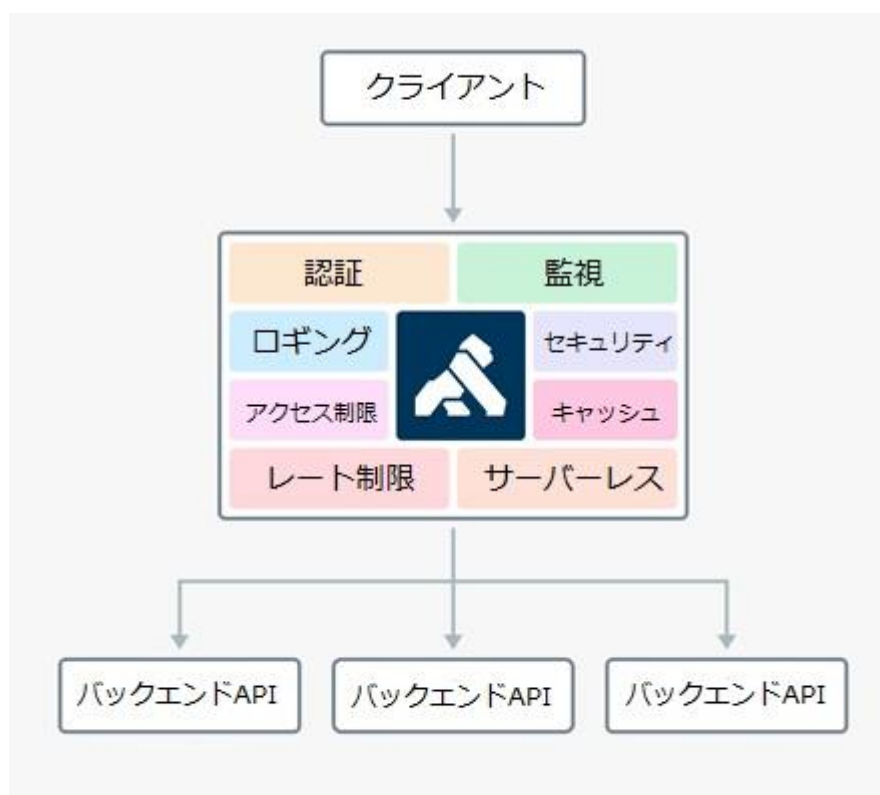


図 2-1 Kong Gateway の概要

参考までに、エリア・データ連携基盤の推奨モジュール「API ゲートウェイ」の機能要件と Kong Gateway の機能との対応関係を表 2-1 に示す。

表 2-1 API ゲートウェイの推奨モジュール要件と Kong Gateway の機能との対応関係

API ゲートウェイの要件				Kong Gateway の機能 (本書内の記載箇所)	
分類	機能	要件	必須	推奨	
基本機能	認証	API キー等により API を実行したユーザーの正当性を確認できること	●		キー認証による保護 (3-3(2))
	ルーティング	受け付けた要求をバックエンドのマイクロサービスへ転送できること	●		ルートの登録 (3-2)
	ライフサイクル管理	API ゲートウェイで管理する API を登録/参照/変更/削除できること	●		サービスの登録 (3-1) ルートの登録 (3-2) サービス、ルートの更新 (3-4) サービス、ルートの削除 (3-5)
	バージョン管理	バージョンの異なるマイクロサービスの API を管理できること	●		管理 API のサービスオブジェクト (表 2-2 の No. 3)
セキュリティ	アクセス制限	アクセス可能な API やデータを制限できること		●	アクセス制御のプラグイン (表 2-2 の No. 7)
	レート制限	単位時間あたりの API 実行回数を制限できること		●	レート制限による保護 (3-3(3))
利用可能 API	REST API	REST API を管理・公開できること	●		API の利用 (3-3)
分析・ モニタリング	アクセス分析	API アクセス数を分析し、統計情報を可視化できること		●	moesif を利用したプラグイン (表 2-2 の No. 9)
	アラート監視	API で発生したアラートを検出できること		●	moesif を利用したプラグイン (表 2-2 の No. 9)

2-2. 公式ドキュメント

本書では、Kong Gateway（動作確認バージョン：Kong v3.4.0）を対象に、代表的な機能の利用手順の概要を説明する。詳細については、以下の公式ドキュメントを確認すること。

表 2-2 公式ドキュメント一覧

No.	公式ドキュメントの URL	概要
1	https://docs.konghq.com/gateway/3.4.x/	Kong Gateway の公式ドキュメント
2	https://docs.konghq.com/gateway/3.4.x/admin-api/	管理用の REST API のリファレンス
3	https://docs.konghq.com/gateway/3.4.x/admin-api/#service-object	管理用の REST API のサービスオブジェクト
4	https://docs.konghq.com/gateway/3.4.x/how-kong-works/routing-traffic/	リクエスト転送に関するリファレンス
5	https://docs.konghq.com/hub/	Kong Gateway のプラグインカタログ
6	https://docs.konghq.com/hub/kong-inc/rate-limiting/	レート制限のプラグイン
7	https://docs.konghq.com/hub/kong-inc/acl/	アクセス制御のプラグイン
8	https://docs.konghq.com/hub/kong-inc/prometheus/	Prometheus を利用したアクセス分析プラグイン
9	https://docs.konghq.com/hub/moesif/kong-plugin-moesif/	moesif を利用したプラグイン

3. 利用手順

Kong Gateway では、クライアントのリクエストをバックエンドの API 実行サーバに転送する際に、サービスとルートという設定情報を使用して管理する。

サービスは、バックエンド API への接続設定を定義する。例えば、接続先 URL やタイムアウト時間などである。

ルートは、クライアントのリクエストをサービスに紐づける設定を定義する。例えば、特定のパスへのアクセスを、指定したサービスに紐づける。

サービスとルートを組み合わせることで、クライアントからのリクエストをバックエンドの API 実行サーバに転送する。(図 3-1)

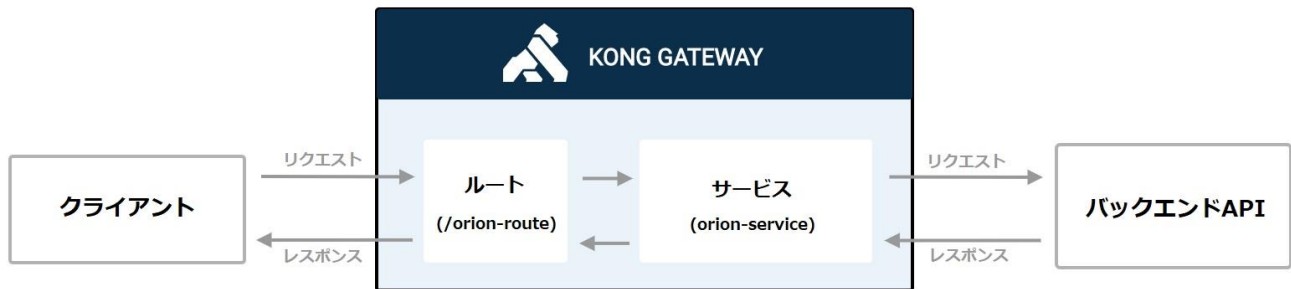


図 3-1 Kong Gateway のルーティングフロー

以降では、サービス、ルートを登録し、Kong Gateway を介してバックエンド API を実行する手順を説明する。

3-1. サービスの登録

以下のコマンドを実行し、サービスを登録する。Kong Gateway では、サービスやルートなどの設定を行うための管理用の REST API（以降、管理 API）を 8001 番ポートで受け付けている。

```
# curl -i -X POST http://example.kong-gateway.com:8001/services ¥
--data name=orion-service ¥
--data url='http://example.orion.com:1026'
```

- `example.kong-gateway.com` : Kong Gateway のホスト名
- `orion-service` : サービスの名称
- `http://example.orion.com:1026` : バックエンド API 実行サーバの URL

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#service-object>

登録に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 201 Created
Date: Wed, 27 Dec 2023 06:28:07 GMT
Content-Type: application/json; charset=utf-8
:
{
  "path": null,
  "port": 1026,
  "client_certificate": null,
  "enabled": true,
  "created_at": 1703658487,
  "name": "orion-service",
  "updated_at": 1703658487,
  "tags": null,
  "write_timeout": 60000,
  "read_timeout": 60000,
  "protocol": "http",
  "host": "10.0.1.253",
  "tls_verify_depth": null,
  "ca_certificates": null,
  "retries": 5,
  "tls_verify": null,
  "connect_timeout": 60000,
  "id": "2d0adc14-974b-44ee-aa7c-e01381d2c75a"
}
```

3-2. ルートの登録

以下のコマンドを実行し、ルートを登録する。

```
# curl -i -X POST http://example.kong-gateway.com:8001/services/orion-service/routes \
--data 'paths[]=/orion' \
--data name=orion-route
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-service** : このルートに紐づけるサービスの名称
- **/orion** : このルートに一致するパスのリスト
- **orion-route** : ルートの名称

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#route-object>

登録に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
Connection: keep-alive
:
{
  "paths": [
    "/orion"
  ],
  "regex_priority": 0,
  "preserve_host": false,
  "created_at": 1703658757,
  "name": "orion-route",
  "headers": null,
  "updated_at": 1703658757,
  "tags": null,
  "service": {
    "id": "2d0adc14-974b-44ee-aa7c-e01381d2c75a"
  },
  "strip_path": true,
  "request_buffering": true,
  "response_buffering": true,
  "path_handling": "v0",
  "https_redirect_status_code": 426,
  "hosts": null,
  "snis": null,
  "protocols": [
    "http",
    "https"
  ],
  "methods": null,
```

```
"sources": null,  
"destinations": null,  
"id": "d35f77c2-ffde-4185-a1a2-2b82581b1949"  
}
```

以上でサービス、ルートが登録された。

なお、「services」API または 「routes」API をオプション指定せずに GET メソッドで実行することで、登録されているサービスまたはルートの一覧を確認することができる。詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#list-services>

<https://docs.konghq.com/gateway/3.4.x/admin-api/#list-routes>

3-3. API の利用

(1) 転送の確認

前章までで登録したサービス、ルートの設定が適切に機能し、リクエストをバックエンドの API 実行サーバに転送できているか確認する。

以下のコマンドを実行する。Kong Gateway は、クライアントからのリクエストを 8000 番ポートで受け付けている。

```
# curl -i -X GET http://example.kong-gateway.com:8000/orion/version
```

上記のリクエストは、Kong Gateway を介して、以下の URL に転送される。

```
http://example.orion.com:1026/version
```

成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 744
:
{
  "orion" : {
    "version" : "3.10.1",
    "uptime" : "0 d, 3 h, 51 m, 3 s",
    "git_hash" : "9a80e06abe7f690901cf1586377acec02d40e303",
    "compile_time" : "Mon Jun 12 16:55:20 UTC 2023",
    "compiled_by" : "root",
    "compiled_in" : "buildkitsandbox",
    "release_date" : "Mon Jun 12 16:55:20 UTC 2023",
    "machine" : "x86_64",
    "doc" : "https://fiware-orion.rtdf.io/en/3.10.1/",
    "libversions": {
      "boost": "1_74",
      "libcurl": "libcurl/7.74.0 OpenSSL/1.1.1n zlib/1.2.12 brotli/1.0.9 libidn2/2.3.0
libpsl/0.21.0 (+libidn2/2.3.0) libssh2/1.9.0 nghttp2/1.43.0 librtmp/2.3",
      "libmosquitto": "2.0.15",
      "libmicrohttpd": "0.9.76",
      "openssl": "1.1",
      "rapidjson": "1.1.0",
      "mongoc": "1.23.1",
      "bson": "1.23.1"
    }
  }
}
```

(2) キー認証による保護

現在の設定では、全てのリクエストをバックエンドの API 実行サーバに転送する。以下の手順では、Kong Gateway の key-auth プラグインを使用して、正しいキーを持つリクエストのみを転送するように設定する。これにより、不正なリクエストは全て拒否され、バックエンド API を不正使用から保護する。

以下のコマンドを実行し、サービスに key-auth プラグインを設定する。プラグインは、指定のサービスのみではなく、指定のルートや全体に設定可能。

```
# curl -i -X POST ¥
--url http://example.kong-gateway.com:8001/services/orion-service/plugins/ ¥
--data 'name=key-auth'
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-service** : このプラグインを紐づけるサービスの名称
- **key-auth** : プラグインの名称

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#plugin-object>

成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
Connection: keep-alive
:
{
  "route": null,
  "protocols": [
    "grpc",
    "grpc",
    "http",
    "https"
  ],
  "config": {
    "run_on_preflight": true,
    "anonymous": null,
    "key_names": [
      "apikey"
    ],
    "hide_credentials": false,
    "key_in_header": true,
    "key_in_query": true,
    "key_in_body": false
  },
  "consumer": null,
  "enabled": true,
```

```
"service": {
  "id": "2d0adc14-974b-44ee-aa7c-e01381d2c75a"
},
"name": "key-auth",
"instance_name": null,
"updated_at": 1703659281,
"tags": null,
"created_at": 1703659281,
"id": "70fbc389-40b2-43c2-83b8-7271ac80113f"
}
```

以下のコマンドを実行し、key-auth プラグインが設定されていることを確認する。

```
# curl -i -X GET http://example.kong-gateway.com:8000/orion/version
```

正しく設定されていると、以下のレスポンスが返却される。正しいキーが指定されていないため、認証エラーとなる。

```
HTTP/1.1 401 Unauthorized
Date: Wed, 27 Dec 2023 06:46:09 GMT
Content-Type: application/json; charset=utf-8
:
{
  "message": "No API key found in request"
}
```

キー認証で保護された API を利用するためには、ユーザを登録し、有効な認証キーを発行する必要がある。以下のコマンドを実行し、ユーザを登録する。

```
# curl -i -X POST ¥
  --url http://example.kong-gateway.com:8001/consumers/ ¥
  --data 'username=TestUser'
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **TestUser** : ユーザ名

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#consumer-object>

登録に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 201 Created
Date: Wed, 27 Dec 2023 06:47:38 GMT
Content-Type: application/json; charset=utf-8
:
{
  "created_at": 1703659658,
  "username": "TestUser",
  "custom_id": null,
  "updated_at": 1703659658,
  "tags": null,
  "id": "de81cbb8-03fd-4f05-80d6-5a7e77c7de12"
}
```

続いて、以下のコマンドを実行し、登録したユーザに対して、有効な認証キーを発行する。

```
# curl -i -X POST ¥
    --url http://example.kong-gateway.com:8001/consumers/TestUser/key-auth
```

- `example.kong-gateway.com` : Kong Gateway のホスト名
- `TestUser` : ユーザ名

発行に成功すると、以下のレスポンスが返却される。「XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX」が発行された認証キーである。

```
HTTP/1.1 201 Created
Date: Wed, 27 Dec 2023 06:50:37 GMT
Content-Type: application/json; charset=utf-8
:
{
  "created_at": 1703659837,
  "ttl": null,
  "key": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "consumer": {
    "id": "de81cbb8-03fd-4f05-80d6-5a7e77c7de12"
  },
  "tags": null,
  "id": "96d61c7f-bd83-4abf-969a-15eaf6522b39"
}
```

以下のコマンドを実行し、発行された認証キーを指定して API にアクセスできることを確認する。

```
# curl -i -X GET http://example.kong-gateway.com:8000/orion/version?apikey=<認証キー>
```

成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 744
:
{
  "orion" : {
    "version" : "3.10.1",
    :
  }
}
```


(3) レート制限による保護

レート制限は、API を偶発的または悪意のある乱用から保護する。レート制限なしの場合、クライアントは好きなだけリクエストすることができ、他のクライアントからのリクエストを受け付けられないようなリクエストの急増につながる可能性がある。以下の手順では Kong Gateway の rate-limiting プラグインを使用して、単位時間あたりのリクエスト数を制限する。

以下のコマンドを実行し、サービスに rate-limiting プラグインを設定する。プラグインは、指定のサービスのみではなく、指定のルートや全体に設定可能。

```
# curl -i -X POST ¥
      --url http://example.kong-gateway.com:8001/services/orion-service/plugins/ ¥
      --data 'name=rate-limiting' ¥
      --data config.minute=5
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-service** : このプラグインを紐づけるサービスの名称
- **rate-limiting** : プラグインの名称
- **5** : 1分あたりに実行できるリクエスト数

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/hub/kong-inc/rate-limiting/>

成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 201 Created
Date: Wed, 27 Dec 2023 07:25:54 GMT
Content-Type: application/json; charset=utf-8
:
{
  :
  "name":"rate-limiting",
  :
  "config":{
    :
    "minute":5,
    :
  }
}
```

以下のコマンドを **6 回連続**で実行し、rate-limiting プラグインが設定されていることを確認する。

```
# curl -i -X GET http://example.kong-gateway.com:8000/orion/version?apikey=<認証キー>
```

6回目のリクエストでは、以下のレスポンスが返却される。

```
HTTP/1.1 429 Too Many Requests
Date: Wed, 27 Dec 2023 07:29:18 GMT
Content-Type: application/json; charset=utf-8
:
{
  "message": "API rate limit exceeded"
}
```

3-4. サービス、ルートの更新

管理 API を使用して、登録済みのサービス、ルートの設定内容を更新する。以下の例では、既存のサービスにグループ化、フィルタリング用のタグを設定する。

```
# curl -i -X PUT http://example.kong-gateway.com:8001/services/orion-service/ ¥  
--data url='http://example.orion.com:1026' ¥  
--data 'tags[]=orion-service-tag'
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-service** : 更新するサービスの名称
- **http://example.orion.com:1026** : バックエンド API 実行サーバの URL
- **orion-service-tag** : 設定するタグ文字列

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#service-object>

更新に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 200 OK  
Date: Wed, 27 Dec 2023 07:33:48 GMT  
Content-Type: application/json; charset=utf-8  
:  
{  
  :  
  "tags":[  
    "orion-service-tag"  
  ],  
  :
```

なお、以下のコマンドを実行し、タグ名を指定してサービス情報を取得することができる。

```
# curl -i -X GET http://example.kong-gateway.com:8001/services?tags=orion-service-tag
```

成功すると、以下のレスポンスが返却され、指定したタグに関連付けられたすべてのサービス情報が確認できる。

```
HTTP/1.1 200 OK
Date: Wed, 27 Dec 2023 07:36:59 GMT
Content-Type: application/json; charset=utf-8
:
{
  "data": [
    {
      :
      "tags": [
        "orion-service-tag"
      ],
      :
      "name": "orion-service"
    }
  ],
  "next": null
}
```

同様に、既存のルートにグループ化、フィルタリング用のタグを設定する。

```
# curl -i -X PUT http://example.kong-gateway.com:8001/routes/orion-route/ ¥  
--data 'paths[]=/orion' ¥  
--data 'tags[]=orion-route-tag'
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-route** : 更新するルートの名称
- **/orion** : このルートに一致するパスのリスト
- **orion-route-tag** : 設定するタグ文字列

※管理 API で指定できるパラメータの詳細については、下記の公式ドキュメントを参照。

<https://docs.konghq.com/gateway/3.4.x/admin-api/#route-object>

更新に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 200 OK  
Date: Wed, 27 Dec 2023 07:39:36 GMT  
Content-Type: application/json; charset=utf-8  
  
:  
{  
  :  
  "tags":[  
    "orion-route-tag"  
  ],  
  :
```

なお、以下のコマンドを実行し、タグ名を指定してルート情報を取得することができる。

```
# curl -i -X GET http://example.kong-gateway.com:8001/routes?tags=orion-route-tag
```

成功すると、以下のレスポンスが返却され、指定したタグに関連付けられたすべてのルート情報が確認できる。

```
HTTP/1.1 200 OK
Date: Wed, 27 Dec 2023 07:42:41 GMT
Content-Type: application/json; charset=utf-8
:
{
  "data": [
    {
      :
      "tags": [
        "orion-route-tag"
      ],
      :
      "name": "orion-route"
    }
  ],
  "next": null
}
```

3-5. サービス、ルートの削除

管理 API を使用して、登録済みのサービス、ルートを削除する。以下のコマンドを実行して、既存のサービスを削除する。

```
# curl -i -X DELETE http://example.kong-gateway.com:8001/services/orion-service/
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-service** : 削除するサービスの名称

削除に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 204 No Content
Date: Wed, 27 Dec 2023 07:45:51 GMT
:
```

同様に、既存のルートを削除する。

```
# curl -i -X DELETE http://example.kong-gateway.com:8001/routes/orion-route/
```

- **example.kong-gateway.com** : Kong Gateway のホスト名
- **orion-route** : 削除するルートの名称

削除に成功すると、以下のレスポンスが返却される。

```
HTTP/1.1 204 No Content
Date: Wed, 27 Dec 2023 07:47:15 GMT
:
```