

Kong Gateway利用手順書 (1.0.0版)

2022年07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	2022/07/01

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.....	18
3-5.	サービス、ルートの削除.....	20

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) の利用手順を説明します。

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します。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」を参照ください。

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ます。
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す。

1-3-2. コマンド入力結果の表記方法

(例)

```
HTTP/1.1 201 Created  
Date: Wed, 09 Mar 2022 13:20:02 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ます。

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します。

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 ゲートウェイです。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ます。

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ます。

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します。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ます。

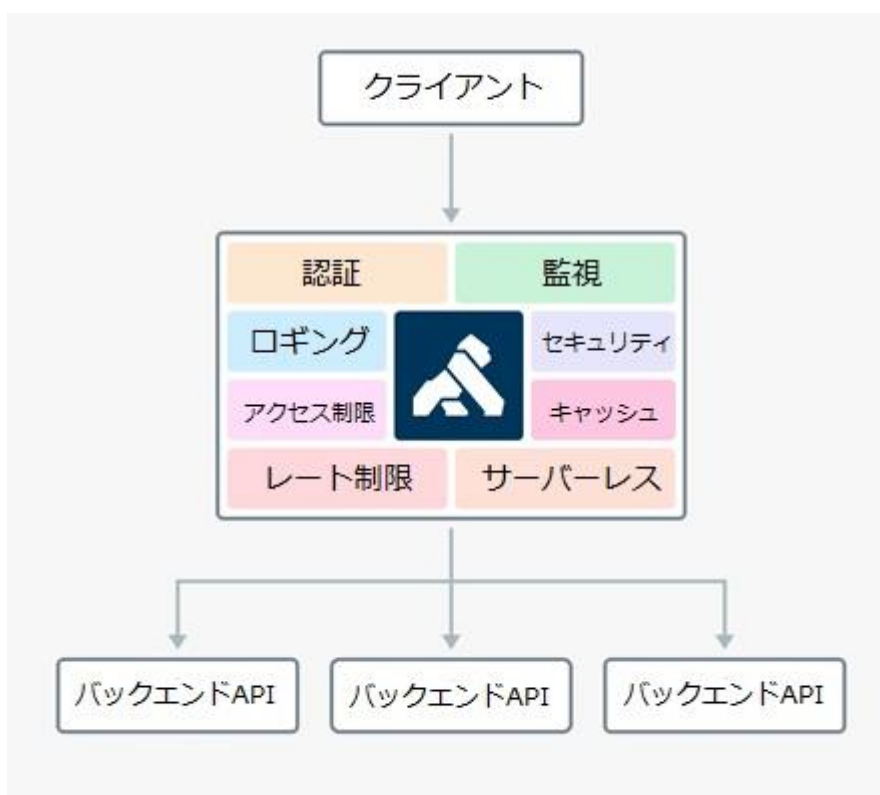


図 2-1 Kong Gateway の概要

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 に示します。

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

分類	API ゲートウェイの要件				Kong Gateway の機能 (本書内の記載箇所)
	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 v2.8.0）を対象に、代表的な機能の利用手順の概要を説明します。詳細については、以下の公式ドキュメントを確認してください。

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/	Kong Gateway の公式ドキュメント
2	https://docs.konghq.com/gateway/2.8.x/admin-api/	管理用の REST API のリファレンス
3	https://docs.konghq.com/gateway/2.8.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/2.8.x/reference/proxy/	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します。

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

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します。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ます。

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 実行サーバに転送します。(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します。

3-1. サービスの登録

以下のコマンドを実行し、サービスを登録します。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ます。

```
# 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ください。

<https://docs.konghq.com/gateway/2.8.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ます。

```
HTTP/1.1 201 Created  
Content-Type: application/json  
Connection: keep-alive  
  
{  
  "host": "example.orion.com",  
  "created_at": 1519130509,  
  "connect_timeout": 60000,  
  "id": "5fc74a83-ff26-4cb6-846b-e0f3f986123e",  
  "protocol": "http",  
  "name": "orion-service",  
  "read_timeout": 60000,  
  "port": 1026,  
  "path": null,  
  "updated_at": 1519130509,  
  "retries": 5,  
  "write_timeout": 60000  
}
```

3-2. ルートの登録

以下のコマンドを実行し、ルートを登録します。

```
# 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ください。

<https://docs.konghq.com/gateway/2.8.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ます。

```
HTTP/1.1 201 Created
Content-Type: application/json
Connection: keep-alive

{
  "created_at":1519131139,
  "strip_path":true,
  "hosts":null,
  "preserve_host":false,
  "regex_priority":0,
  "updated_at":1519131139,
  "paths":[
    "/orion"
  ],
  "service":{
    "id":"5fc74a83-ff26-4cb6-846b-e0f3f986123e"
  },
  "methods":null,
  "protocols":[
    "http",
    "https"
  ],
  "id":"f9ce2ed7-c06e-4e16-bd5d-3a82daef3f9d"
}
```

以上でサービス、ルートが登録されました。

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ます。詳細については、下記の

公式ドキュメントを参照ください。

<https://docs.konghq.com/gateway/2.8.x/admin-api/#list-services>

<https://docs.konghq.com/gateway/2.8.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します。

以下のコマンドを実行します。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ます。

```
# 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 に転送されます。

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

成功すると、以下のレスポンスが返却されます。

```
HTTP/1.1 200 OK
Content-Type: application/json
Connection: keep-alive

{
  "orion" : {
    "version" : "3.6.0",
    "uptime" : "0 d, 0 h, 1 m, 28 s",
    "git_hash" : "973850279e63d58cb93dff751648af5ec6e05777",
    "compile_time" : "Wed Mar 2 10:34:48 UTC 2022",
    "compiled_by" : "root",
    "compiled_in" : "5e6b6f1167f7",
    "release_date" : "Wed Mar 2 10:34:48 UTC 2022",
    "machine" : "x86_64",
    "doc" : "https://fiware-orion.rtdf.io/en/3.6.0/",
    "libversions": {
      "boost": "1_66",
      "libcurl": "libcurl/7.61.1 OpenSSL/1.1.1k zlib/1.2.11 nghttp2/1.33.0",
      "libmosquitto": "2.0.12",
      "libmicrohttpd": "0.9.70",
      "openssl": "1.1",
      "rapidjson": "1.1.0",
      "mongoc": "1.17.4",
      "bson": "1.17.4"
    }
  }
}
```

(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 実行サーバに転送してます。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します。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します。

以下のコマンドを実行し、サービスに `key-auth` プラグインを設定します。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です。

```
# 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ください。

<https://docs.konghq.com/gateway/2.8.x/admin-api/#plugin-object>

成功すると、以下のレスポンスが返却されます。

```
HTTP/1.1 201 Created
Content-Type: application/json
Connection: keep-alive

{
  "protocols": [
    "grpc", "grpcs", "http", "https"
  ],
  "enabled": true,
  "consumer": null,
  "tags": null,
  "name": "key-auth",
  "id": "fe5a32c3-dac3-46e4-9051-67d80a4372ad",
  "config": {
    "key_names": [
      "apikey"
    ],
    "hide_credentials": false,
    "run_on_preflight": true,
    "key_in_header": true,
    "key_in_query": true,
    "key_in_body": false,
    "anonymous": null
  },
  "service": {
    "id": "5fc74a83-ff26-4cb6-846b-e0f3f986123e"
  },
  "created_at": 1647347260,
  "route": null
}
```

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します。

```
# 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ます。正しいキーが指定されていないため、認証エラーとなります。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json
Connection: keep-alive

{
  "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ります。以下のコマンドを実行し、ユーザを登録します。

```
# 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ください。

<https://docs.konghq.com/gateway/2.8.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ます。

```
HTTP/1.1 201 Created
Content-Type: application/json
Connection: keep-alive

{
  "custom_id": null,
  "username_lower": "testuser",
  "tags": null,
  "created_at": 1647507635,
  "id": "f2d1f9bc-dd99-4212-ae39-62b7d8b8fbfa",
  "type": 0,
  "username": "TestUser"
}
```

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します。

```
# 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ます。「XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX」が発行された認証キーです。

```
HTTP/1.1 201 Created
Content-Type: application/json
Connection: keep-alive

{
  "consumer":{
    "id":"f2d1f9bc-dd99-4212-ae39-62b7d8b8fbfa"
  },
  "tags":null,
  "created_at":1647508111,
  "key":"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "ttl":null,
  "id":"f243323c-5554-42a9-96bb-20bd92fa998a"
}
```

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します。

```
# 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ます。

```
HTTP/1.1 200 OK
Content-Type: application/json
Connection: keep-alive

{
  "orion" : {
    "version" : "3.6.0",
    :
  }
}
```


(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します。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ります。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します。

以下のコマンドを実行し、サービスに rate-limiting プラグインを設定します。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です。

```
# 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ください。

<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ます。

```
HTTP/1.1 201 Created
Content-Type: application/json
Connection: keep-alive

{
  "name": "rate-limiting",
  :
```

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します。

```
# 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ます。

```
HTTP/1.1 429 Too Many Requests
Content-Type: application/json
Connection: keep-alive

{
  "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します。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します。

```
# 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ください。

<https://docs.konghq.com/gateway/2.8.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ます。

```
HTTP/1.1 200 OK
Content-Type: application/json
Connection: keep-alive
```

```
{
  "tags":[
    "orion-service-tag"
  ],
  :
```

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します。

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

<https://docs.konghq.com/gateway/2.8.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ます。

```
HTTP/1.1 200 OK
Content-Type: application/json
Connection: keep-alive

{
  "tags":[
    "orion-route-tag"
  ],
  :
```

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、ルートを削除します。以下のコマンドを実行して、既存のサービスを削除します。

```
# 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ます。

```
HTTP/1.1 204 No Content  
Connection: keep-alive
```

同様に、既存のルートを削除します。

```
# 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ます。

```
HTTP/1.1 204 No Content  
Connection: keep-alive
```