API Gateways Comparison

express gateway

vs

Kong

Comparison Guide by the
API experts over at:

LunchBadger

API Gateways: Express Gateway Vs. KONG

ABSTRACT

We've created this comparison page to make it easy to understand the major differences (and similarities) between two popular projects for the API Gateway use case. In this review we'll be comparing Express Gateway and Kong across multiple dimensions and "at-a-glance."'

What is a microservices API Gateway?

A microservices API Gateway sits in front of microservices and performs the external functions of an API Gateway to clients including other microservices. Gateways in general make it much simpler to develop, secure, manage, and scale endpoints by moving most of the infrastructure level logic from both the backend microservices and client, into the gateway.

Further reading: microservices.io

Elevator Pitches

Express Gateway

Express Gateway is an API Gateway that can sit at the heart of any microservices architecture, regardless of what language or platform you're using. Express Gateway secures your microservices and exposes them through APIs using Node.js, ExpressJS and Express middleware. Developing microservices, orchestrating and managing them can now be done insanely fast, on one platform, seamlessly, and without having to introduce additional infrastructure. Many policies for enterprise authentication schemes and other enterprise features for other open source based gateways are available only in their paid versionsThe maintainers of Express Gateway intend to make enterprise policies and many enterprise features for Express Gateway - free.An API Gateway is a critical infrastructure component in the enterprise that makes available backend services to mobile, web and other external clients via a set of protocols and commonly through a set of RESTful application programming interfaces (APIs). An API Gateway makes it much simpler to develop, secure, manage, and scale endpoints by moving most of the required logic from the client, into the gateway.

The project can do so because it leverages the 3,000+ ExpressJS middleware components to quickly build functionality that is rapidly becoming commoditized anyway.Express Gateway's single core strength is its existing ecosystem to leverage and limitless customization and extensibility not just through plugins but because its core is completely written in JavaScript, a language that is ubiquitous among developers.

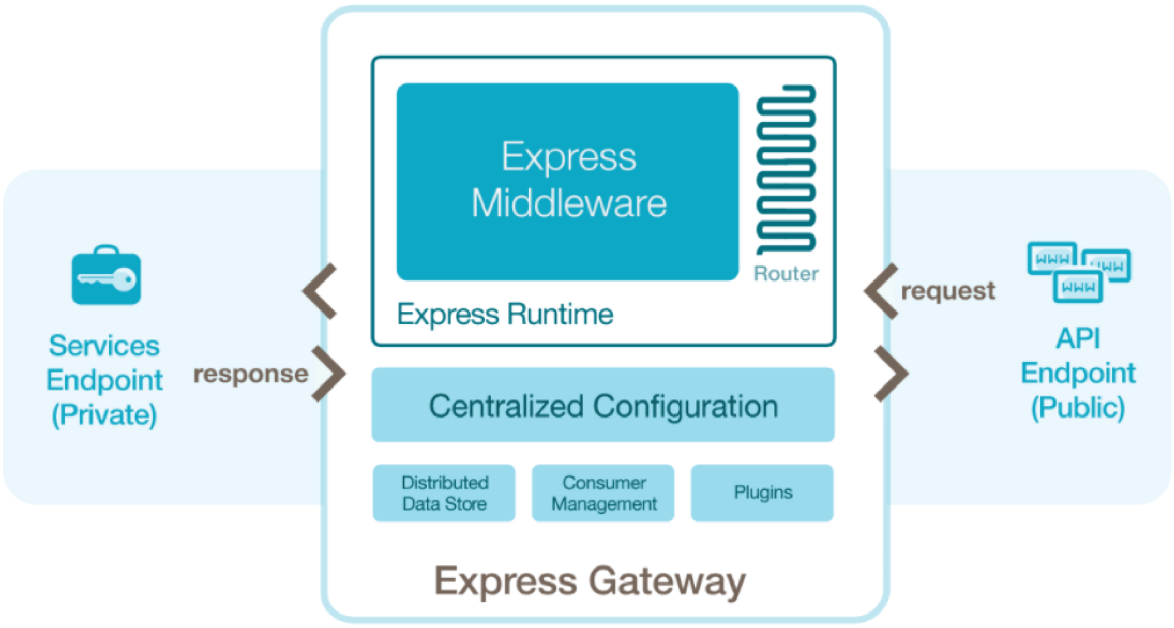
Learn more about Express Gateway.

Kong

Kong is a scalable, open source API Layer (also known as an API Gateway, or API Middleware). Kong runs in front of any API and is extended through plugins, which provide extra functionality and services beyond the core platform.

Learn more about Kong.

Express Gateway Features & Architecture



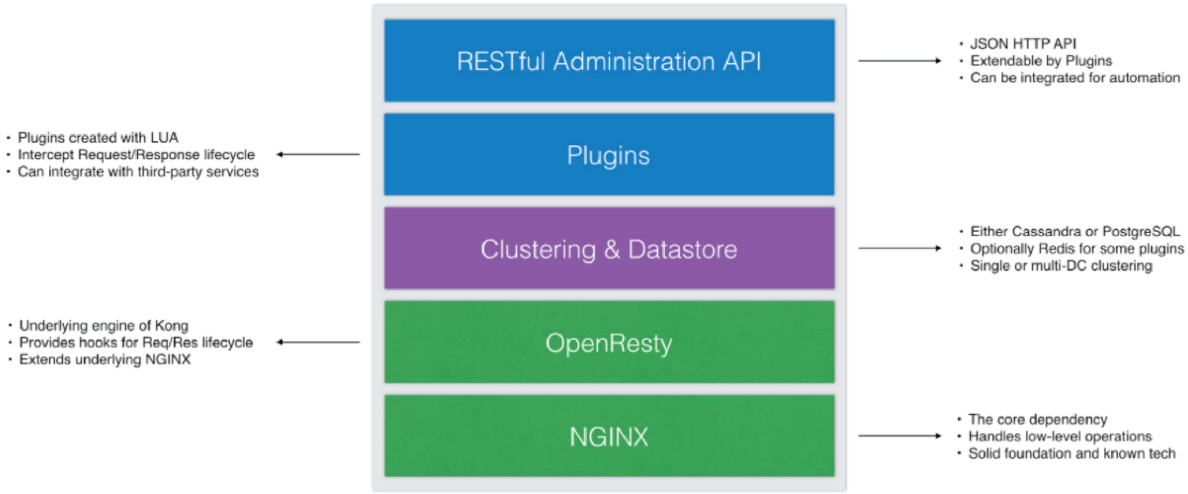
Express Gateway is an open source API Gateway written in Node.js and built on top of Express.js and Express Middleware. Express.js is the most popular framework for Node.js applications. Express Gateway is Apache 2.0 licensed with commercial support available from LunchBadger.

Features include:

- Simple configuration via a YAML file
- Plugin architecture
- Extensible with over 3,000 modules and growing
- Runs anywhere (Docker, private or public cloud)
- Auto-detects and hot-reload of configuration changes
- Define multiple policy sets (pipeline) per API endpoint
- Supports any language
- Supports any framework
- Supports many microservice use cases, patterns and designs
- Works with any orchestrator
- Works with any service mesh
- Plugs directly into existing DevOps tooling and pipelines

Further reading: Express Gateway GitHub repository

KONG Features & Architecture



Kong is an open source API Gateway written in Lua and built on top of NGNIX and Apache Cassandra or PostgreSQL. Kong is Apache 2.0 licensed with commercial support available via Kong Enterprise.

Features include:

- Configuration and administration tasks split between REST API and CLI
- Extensible via 36 available plugins (30 open source, 6 commercial)
- Enterprise plugins offered by Kong are paid
- Runs anywhere (Kubernetes, Docker, private or public cloud)
- Scales by simply adding more machines
- Dynamically balances load across backend services
- Spawns new processes vs to reload configuration changes
- Support for one set of policies for all API endpoints which can be modified with conditional flow
- Supports any language or framework
- Supports all microservice use cases, patterns and designs
- Works with any orchestrator and service mesh
- Plugs directly into existing DevOps tooling and pipelines

Further reading: Kong GitHub repository

Express Gateway Getting Started

Getting started with Express Gateway is very easy because all you need is Node.js with an optional backend data store. Getting up and running with Express Gateway is a simple, four step process executed at the command-line (detailed here.)

Installation:

- Installing Express Gateway via npm
- Creating an Express Gateway
- Following the command-line prompts to configure the gateway from the available templates
- Running Express Gateway

If you choose to get started using Docker, all you need is a Node.js Docker image. You can even compile your gateway into a standalone executable using pkg. Finally, if you need a data store because you intend to provide your Express Gateway users with API keys, you have numerous options concerning Redis. These include the Redis Docker image, Redis Cloud, or Compose's hosted Redis offering.

KONG Getting Started

Getting started with Kong depends on the environment you are targeting. Installation packages and instructions are available here.

Supported Environments:

- Docker
- Kubernetes
- Mesosphere DC/OS
- AWS
- CentOS
- Red hat
- Debian
- Ubuntu
- OSX
- Google Cloud
- Vagrant

The easiest way to deploy Kong is using Docker or Kubernetes. Unless you already have expertise in setting up and running Cassandra or PostgreSQL in production, you will likely want to use Kong's Docker or Kubernetes installation instructions so you can easily start one of the required databases. You can also use Amazon RDS to run a hosted PostgreSQL instance for Kong. To deploy a Kong instance with the same configuration, your new Kong instance needs to be able to connect to the same database as your old Kong instance, or you need to copy the database state from one database instance to another.

Database Support

Express Gateway

By default, Express Gateway uses a built-in, in-memory data store. Express Gateway can run with or without a backend. If a persistent backed is desired, Express Gateway supports Redis. Express Gateway stores most of its configuration in a YAML configuration file. Express Gateway only stores transactional data, like user information and access tokens, in its data store. This means if you are just using Express Gateway for rate limiting and header transformation, you don't need a data store at all.

KONG

Kong must store its configuration in either PostgreSQL or Cassandra. This includes which plugins you're using, which APIs are exposed and every other tunable parameter of your Kong deployment.

Administration and Maintenance

Express Gateway

Kong and Express Gateway have very different models for managing their gateway configuration. Express Gateway's high-level configuration is defined in a YAML file that you can track in GitHub. There is also a CLI and admin API for managing users and credentials. At the moment there is no officially supported GUI for the Admin API. However, unless you need to create users and credentials, you can configure Express Gateway entirely using a YAML file. The YAML file is one such feature that makes Express Gateway cloud native and ready to be run containers and orchestrators like Kubernetes to take advantage of cloud native features like ConfigMaps.

LunchBadger offers Express Gateway Enterprise which adds a GUI called the Canvas that allows a developer or devops engineer to configure and manage multiple Express Gateway instances and automatically deploy instances in Kubernetes for scale and cloud native management.

KONG

As mentioned, Kong stores your configuration in a backend data store, either PostgreSQL or Cassandra. You configure Kong using a API. If you're a user of the community version, you need to either interact with the RESTful API directly via HTTP requests using a tool like curl, or via a third-party GUI like Konga. The officially supported Kong admin GUI is for Kong Enterprise subscribers only.

Express Gateway Built-in Plugin Support

Express Gateway is currently 100% free and there are no paid plugins. Express Gateway has enterprise policies such as OAuth2 introspection and JWT generation and verification natively for free. Other gateways like Kong only offer such features as paid plugins or not at all.

Licensing costs for Express Gateway Enterprise are sized by production deployment and is directly correlated with your success of using the product and receiving its value. Express Serverless Platform extends Express Gateway Enterprise by adding microservice integration and composition capabilities by writing simple functions to provide one seamless experience to build microservices and expose them as APIs under one Kubernetes runtime.

KONG Built-in Plugin Support

Kong is built on an freemium model. Plugins like rate limiting and features like the admin GUI require a Kong Enterprise subscription. Kong does not provide information on how much Kong Enterprise costs on their website, so you'll need to inquire to get pricing.

Check out the side-by-side comparison chart on the next page.

Built-in Plugin Support

Below is a comparison of built-in plugins for Express Gateway and Kong. The key differences include Kong only having rudimentary rate limiting, OAuth 2, and request transformation plugins for community users.

A subscription to Kong Enterprise is required to get access to the full functionality of these plugins.

With Express Gateway, you get the fully featured plugins for free.

Features Comparison

Features	Express Gateway	Kong
Public or Private Cloud	✓	✓
On Premises	✓	✓
Con guration & Administration	**YAML, CLI, API**	**CLI, GUI, API**
Database (Persistent Storage)	**Redis**	**Cassandra, Postgre SQL**
HTTPS	✓	✓
CORS	✓	✓
Basic Authentication	✓	✓
Key Authentication	✓	✓
OAuth 2	✓	**(1)
JWT	✓	**(2)
Finegrain Access Control	✓	**(3)
Rate Limiting	✓	**(4)
Request Transformation	✓	**(5)
Response Transformation	✓	**(5)
Pipeline Driven Conditional Actions	✓	✗
Pipeline Driven Expressions	✓	✗
Consumer Management	✓	✓

1. OAuth 2.0 works with Kong Community Edition, but you need to pay for the Enterprise edition if you want to validate third party tokens. A 3rd party introspection plugin available.
2. Express Gateway is able to issue JWT as well as a result of an OAuth2 flow, while Kong releases just an opaque one. A 3rd party introspection plugin available.
3. In Kong you can define which consumers get access to the endpoint, but this might be a stretch to consider fine grained versus Express Gateway which can set access control down to the scope-level with verb(s) pairing for HTTP.
4. Rate Limiting has an extended version, where the website claims to have higher flexibility and better performances. However, the list of additional features hasn't been completely disclosed.
5. Request Transformer has an extended version that allows for adding or replacing content in the upstream request based on variable data found in the client request, such as request headers, query string parameters, or URI parameters as defined by a URI capture group

Custom Plugin Support

Express Gateway Custom Plugin Support

When built-in plugins are not enough, both Kong and Express Gateway have a mechanism for writing custom plugins. Express Gateway plugins are written in JavaScript using the Express.js framework.

Express Gateway plugins are analogous to Express middleware, which makes it easy to reuse the prolific JavaScript ecosystem in your API gateway. JavaScript is the number one programming language in terms of repos on GitHub and the number four language in terms of job listings on LinkedIn in 2017.

KONG Custom Plugin Support

Kong is built on top of NGINX, which means Kong plugins must be written in Lua using the lua-nginx-module. Unfortunately, Kong does not have official support for writing plugins in nginScript, NGINX's unique JavaScript runtime. It should also be noted that the Lua language doesn't have sufficient adoption to make either the Github or Linkedin lists mentioned previously.

Enterprise Versions

Express Gateway Enterprise and Express Serverless Platform

LunchBadger has built a commercial product around Express Gateway with two editions:

Express Gateway Enterprise adds a visual interface called the Canvas to manage, configure and orchestrate multiple instances of Express Gateway. Express Gateway Enterprise also provides a Kubernetes based runtime and a set of microservices that automate all actions performed within the Canvas into Kubernetes in real time in development and immutable deployment for testing and production.

Express Gateway Enterprise utilizes Kubernetes for multicloud management and scale. Utilizing Kubernetes natively also makes it easier for enterprises to realize their cloud strategy and its benefits across application and infrastructure under the same leading container orchestrator.

Express Serverless Platform further extends Express Gateway Enterprise into a full blown microservices platform. It adds microservices integration and composition capabilities under the same unified experience provided by the Canvas, Kubernetes, Kubernetes automation – as ONE seamless platform to develop microservices as functions and expose them as APIs.

Kong

KONG Enterprise

Kong offers Kong Enterprise. Kong Enterprise brings a breadth as an API solutions with not only a gateway but also a developer portal, dashboard for management, and API analytics. Kong Enterprise focuses on API management as its strength in its offering but has no native capabilities to develop the actual microservices and APIs themselves. Enterprise features for Kong are only offered by Kong in its Enterprise Edition like OAuth2 introspection and others.

Quick Reference Links

Express Gateway

- Github Repository
- Documentation
- Installation and Getting Started
- Plugins
- Commercial Support

KONG

- Github Repository
- Documentation
- Installation and Getting Started
- Plugins
- Commercial Support

Summary

Express Gateway

Both Kong and Express Gateway have their strengths and weaknesses. Express Gateway is relatively new, but is built on the well-established Express.js framework and is written entirely in JavaScript. Express Gateway is already running in production at both small and large companies handling significant production workloads. Although it currently only supports Redis as a backend data store, for many use cases a data store is not required because Express Gateway is configured via a YAML file rather than a GUI. This has the additional benefit of letting you track your configurations in Git. Express Gateway is arguably easier to setup and manage, unless you're already an expert with NGINX, Lua, PostgreSQL or Cassandra.

KONG

Kong has a compelling story around performance and reliability because it is built on top of NGINX. That being said, it also exhibits some potentially cumbersome dependencies. Kong's extensibility can be viewed as limited because the custom plugins must be written in Lua. Furthermore, using Kong also requires you to adopt either PostgreSQL or Cassandra, which is troubling if you have no experience with either database. Finally, Kong is configured entirely through an admin GUI rather than a configuration file, so tracking configuration in a source version control system like Git is difficult.