

API Gateways Comparison



express gateway

VS



Comparison Guide by the
API experts over at:



LunchBadger

API Gateways: Express Gateway Vs. Tyk

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 Tyk across multiple dimensions, "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 versions. The 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](#).

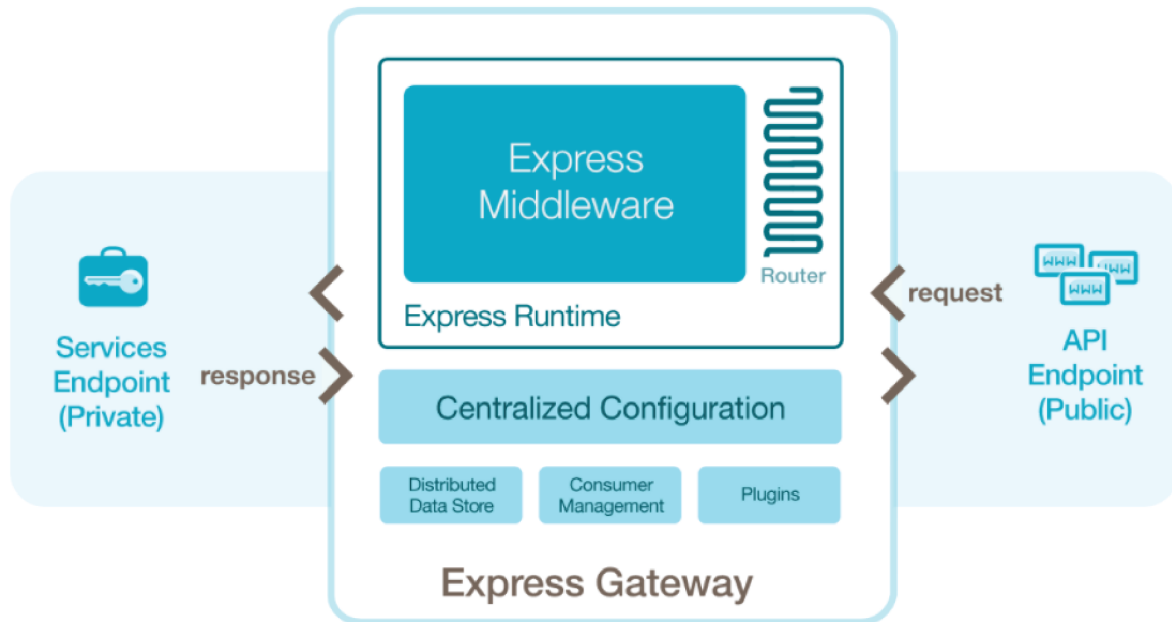


Tyk

Tyk is an open source API Gateway that is fast, scalable and modern. Tyk is built using Golang. Out of the box, Tyk offers an API Management Platform with an API Gateway, API Analytics, Developer Portal and API Management Dashboard with some components open source and other paid for only.

Learn more about [Tyk](#)

Express Gateway Features & Architecture



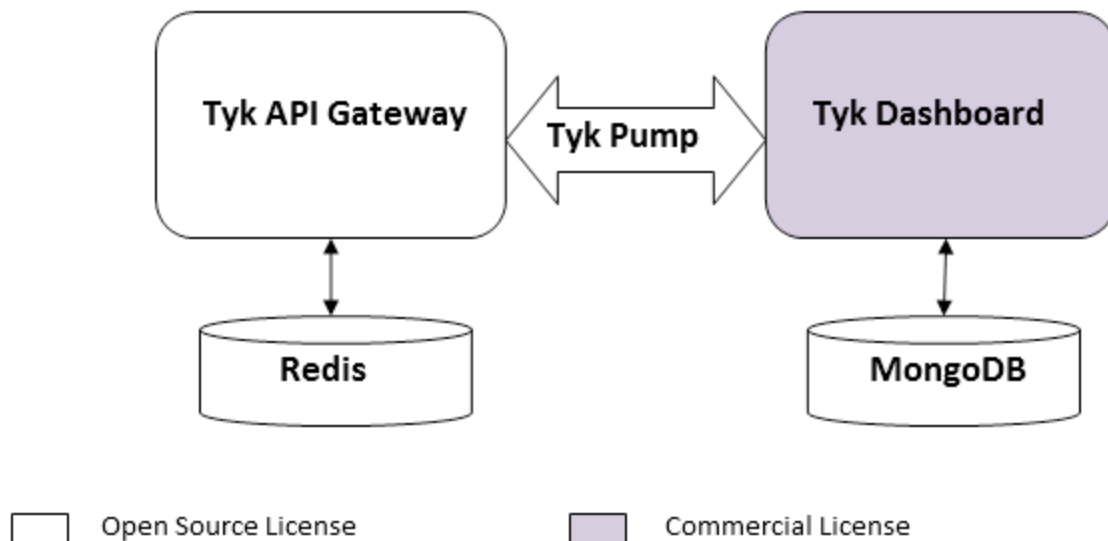
Express Gateway is an open source API Gateway written in Node.js and built on top of Express.js and Express Middleware. Express.js is the most popular framework for Node.js applications. Express Gateway is Apache 2.0 licensed with commercial support available from LunchBadger.

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](#)

Tyk Features & Architecture



While the core API Gateway is free and open source, the Dashboard requires a separate commercial license, and may have to be paid for, depending on its usage. In other words, a full-featured on-premise deployment of the Tyk platform comes with a set of pricing tiers:

- A free edition limited to 1 Gateway instance
- A Premium Edition limited to 2 Gateway instances at £6,000 per annum subscription
- A Premium Edition with unlimited number of Gateway instances at £10,000 per annum subscription

Tyk also offers hosted deployment called "Tyk Cloud" which is free up to 50,000 API calls per day, and then charged £200 per month (for up to 1 million API calls per day), £1,000 per month (for up to 5 million API calls per day), £2,000, per month (for up to 10 million API calls per day) and so on.

More info on Tyk's pricing model [here](#).

Tyk's features include:

- RESTful API
- Multiple access protocols
- Rate limiting
- Quotas
- Granular access control
- Key expiry
- API versioning
- Black and white lists and ignored endpoint access
- Analytics logging
- Webhooks
- IP whitelisting
- Zero downtime restarts

Further reading: [Tyk 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](#).

Tyk Getting Started

Tyk offers installation packages for common OS platforms and the choice of a cloud, hybrid or on-premise deployment. The Tyk installation instructions and dependencies vary depending on where you want to deploy. The Tyk installation is a four-step process, involving two different databases (Redis and MongoDB), plus two Tyk-specific components. Depending on the how you scale your deployment, these characteristics could lead to higher administration costs over the long run.

- [Getting Started with Tyk Cloud](#)
- [Getting Started with Tyk Hybrid](#)
- [Getting Started with Tyk On-Premise](#)



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.

Tyk

The core API Gateway only needs Redis as a dependency, but the full-featured product, including the Dashboard, requires MongoDB and another component called Tyk Pump which serves as a connector between the API Gateway and the Dashboard.



Administration and Maintenance

Express Gateway

Tyk 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.

It is interesting to note that the recommended approach to scale and load-balance both Express Gateway and Tyk are based on Kubernetes. This [Express Gateway demo](#), with [source code](#), explains the set-up using Kubernetes.

Tyk

Tyk users may choose either a web-based GUI or REST APIs for Administration. Tyk's configuration settings are stored in a Redis database. For details on how to scale Tyk, you may follow the relevant documentation [here](#).

Features Comparison

For an apples-to-apples comparison, let's consider Express Gateway and the core API Gateway offered by Tyk (the free and open source component of the Tyk platform.)

Both Express Gateway and Tyk support plugins to extend the core platform, and enable custom protocol handlers, request / response transformations, authentication modules, etc. Tyk actually supports two types of [plugins](#): JavaScript (or JSVM) plugins for code injection into the request path, and 'rich plugins', that replace existing functionality with custom-built implementations. Richer plugins have to be written in Lua or Python, or using a gRPC protocol (which arguably permits you use any language or framework.) These plugins run on a separate process from the core server - which may translate to a bit more coding and debugging effort.

The plugin architecture of Express Gateway is much simpler, mainly because, these plugins are in essence, Express middleware, and run in-process with the core server. Express framework being overwhelmingly popular among developers, it should be easier to code and / or find plugins with needed functionality, compared to plugins coded in Lua or Python. Moreover, the Express Gateway plugin framework specifies differentiated extension points like policies, conditions, routes, etc. that should lead to more elegant design, when extending the platform.

For some projects, fine-grained access control may be needed to restrict access to certain URL paths within an API endpoint, to a specific group of users. In Express Gateway, one may use '[scopes](#)' to implement this functionality, whereas, Tyk offers [security policies](#) (including 'partitioned policies'). The approaches are different, but the end goals achieved are similar.

Express Gateway also allows you to configure the request flow for an API endpoint to a greater extent. [Conditions](#) can be set on actions to be executed on a request, based on matching URL patterns, hosts, request methods, etc. Javascript code blocks can be inserted on to a request pipeline by means of '[expressions](#)', for purposes like setting request headers, enforcing redirects, and performing request transformations. Meanwhile, Tyk allows read-only access to '[context variables](#)' within the request pipeline, and inserting request headers. Expressions in Express Gateway allow for a more general-purpose and lightweight method of code injection. To achieve equivalent capability in Tyk a user would have to code and install a full-fledged JSVM plugin.

Finally, Express Gateway supports both user-based and **IP-address-based rate limits**, while Tyk does not support the latter capability as of now. On the other hand, Tyk may be the preferred choice, if you need an out-of-the-box dashboard displaying real-time stats of API usage. Tyk also comes with a web-based Administration Console (as part of the open-source offering itself) in addition to REST APIs for administration that can be invoked using curl. For Express Gateway users, a CLI and Admin APIs are the available options. Express Gateway utilizes Prometheus to instrument infrastructure and application metrics.

Features Comparison

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

1. MongoDB is require separately for the dashboard as well as installing Tyk Pump

2. With Tyk rate limiting is rather constrained. For example you cannot rate limit by IP.



Custom Plugin Support

Express Gateway

When built-in plugins are not enough, both Tyk 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.

Tyk

Tyk plugins can be written in a variety of languages. For a complete list of supported languages, check out the documentation [here](#). Note that the scope in which you can extend the core of Tyk is limited, whereas with Express gateway there are no such limitations.



Enterprise Versions

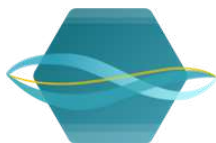
Express Gateway Enterprise and Express Serverless Platform

In addition to [Express Gateway Support and Expert Services](#), 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.



Tyk API Gateway

Tyk offers Tyk Enterprise. Tyk Enterprise brings enterprise support, high availability by default, and multiple data center support for federated deployment. Tyk's API management solution is highly scalable and comprehensive but is limited to API management and API design. The developer must still have implementation practices for their own microservices to be wired into Tyk. Tyk has "virtual endpoints" to be able to route API requests to other processes.



Quick Reference Links

Express Gateway

- [Github Repository](#)
- [Documentation](#)
- [Installation and Getting Started](#)
- [Plugins](#)
- [Commercial Support](#)

Tyk API Gateway

- [Github Repository](#)
- [Documentation](#)
- [Installation & Getting Started](#)
- [Commercial Support](#)

Summary

Express Gateway

Among the open source API Gateways available today, Express Gateway and Tyk stand out as two of the more compelling offerings with capabilities such as proxying, rate-limiting, and support for OAuth2. Express Gateway is relatively new, but is built on the well-established Express.js framework and is written entirely in JavaScript. 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.

Tyk

Tyk is a feature-rich API platform with hosted and on-premise deployment options. It is not just an API Gateway, but a platform supporting rich administrative user interfaces, and real-time analytics for API usage patterns. The added capabilities come at a price - licensing, set-up and maintenance of the platform may just be a bit more complex for certain scenarios.