FAQ

Table of Contents

Here are the frequently asked questions:

General questions

What is the license?

All of the Funktion source code is licensed under the [Apache License 2.0](https://www.apache.org/licenses/LICENSE-2.0)

How do I get started?

Please Install Funktion then follow the Getting Started Guide

How do I install funktion?

See the Install Guide

What is serverless?

The term `serverless` just means that with lambda style programming the developer just focuses on writing `functions` only - there is no need for developers to think about managing servers or even containers.

Its not that there are no servers - of course there are - its just that developers don't need to think about them at all, they are managed for you by the platform.

How does Funktion compare to other serverless frameworks?

There are many frameworks out there for serverless. From 30,000 feet they are all quite similar.

Here are the main differences of Funktion:

- Funktion focusses on being Kubernetes and OpenShift native rather than some generic serverless frameworks (like Open Whisk). So Funktion is designed to reuse Kubernetes abstractions like Deployments, Service, Ingress/Route, auto scaling along with mounting ConfigMaps into pods to quickly update pods when the source of functions change to provide rapid developer feedback.
- Event triggering from over 200 different connectors spanning most middleware technologies, databases, messaging systems, APIs, social media networks and cloud services via deep integration with Apache Camel.
- Funktion can trigger any functions and HTTP endpoints from other frameworks and technologies via different trigger technologies than HTTP. Quite a few serverless frameworks, particularly kubernetes specific ones, focus on just exposing a function in some language over

HTTP. We see that as important; but only part of the picture.

- Funktion supports a flow language to orchestrate flows between many functions and endpoints to make composite flows. Similar to AWS Step Functions in concept though it delegates to the endpoint technology for persistence, retries and transactions etc.
- We are working on deep integration into the openshift.io open source developer platform for kubernetes/openshift to support rich development, debugging and CI / CD capabilities along with integration into Eclipse Che. Also we're working on a rich web UI for visually triggering functions from a variety of connectors and building flows across functions such as in this video.