

A lightweight open-source microservice framework.

Develop microservices with Java EE / Jakarta EE technologies and extend them with Node.js, Go and other languages. Migrate existing Java EE applications to microservices and cloud-native architecture.

[Get started](#)[POM generator](#)[PRODUCT](#) [ANNOUNCEMENT](#)[DEVELOPERS](#)

Support for MicroProfile 2.2 and Java 12 is here

[View on blog >](#)[PRODUCT](#) [DEVELOPERS](#)

Announcing KumuluzEE Fault Tolerance 2.0.0

[View on blog >](#)[ANNOUNCEMENT](#) [COMMUNITY](#) [NEWS](#)[PRODUCT](#)

KumuluzEE is celebrating 5 years!
What we've done and the roadmap ahead

[View on blog >](#)

KumuluzEE is Java Duke's Choice Award Winner

KumuluzEE has won the prestigious 2015 Java Duke's Choice Award. The Duke's Choice Award is the Java community equivalent of winning an Oscar. The award celebrates extreme innovation using Java technology.



KumuluzEE is part of MicroProfile

The [MicroProfile](#) is a baseline platform definition that optimizes Enterprise Java for a microservices architecture and delivers application portability across multiple MicroProfile runtimes.



Supports Java, Java EE / Jakarta EE, Node.js, Go and other languages

KumuluzEE supports Java, Node.js, Go and other programming languages. It also provides support for Blockchain, gRPC, GraphQL and many other advanced technologies.



What is KumuluzEE

KumuluzEE is a lightweight framework for developing microservices using standard Java, Java EE / Jakarta EE technologies and migrating existing Java applications to microservices. KumuluzEE packages microservices as standalone JARs. KumuluzEE microservices are lightweight and optimized for size and start-up time. They fit perfectly with Docker containers.

KumuluzEE also provides extensions for developing common patterns in cloud native architectures, including configuration, logging, discovery, circuit-breakers, metrics, security, event streaming and more.

[View all projects >](#)[View MicroProfile specification >](#)

Microservices with Java EE

A standard approach for deploying Java EE applications is packing all components into a single EAR/WAR archive and deploying the archive on the application server. This leads to monolithic architectures which make applications difficult to maintain and scale in cloud (PaaS) environments. The microservice architecture addresses these shortcomings by decomposing an application into a set of stateless microservices. Each microservice has a well-defined functionality and an interface for communication with other services.

[Explore the tutorials >](#)[Quickstart development with online pom generator >](#)

Main features

KumuluzEE allows you to decompose Java applications into microservices and prepare them for the cloud-native architecture. It allows you to create standalone lightweight Java and Java EE / Jakarta EE microservices that can be efficiently deployed, executed and scaled in the cloud on PaaS and Docker-like environments or as standalone applications. It also supports other languages, such as Node.js, Go, and others.

KumuluzEE microservices have minimal footprint and require minimal or no configuration. KumuluzEE

HIGHLIGHTS

Full support for Java SE and EE (Jakarta EE) with support for other languages, including Node.js and Go.



Fastest start-up



Smallest JAR



Fully cloud-native



Optimized for Docker and Kubernetes

Pure JavaEE microservices

Develop pure Java/JavaEE/JakartaEE based microservices using standard Java EE APIs and technologies and pack and execute them as stand-alone (Fat/Uber) JARs.

Standard EE APIs and tech

Develop microservices using standard Java EE APIs and technologies.

Leverage your existing knowledge

Leverage your existing Java knowledge and extend it to gradually migrate existing Java and Java EE applications to microservices and cloud-native architecture with common cloud-native patterns.

Easy migration

Gradually migrate existing Java EE applications to microservices and cloud-native architecture.

Fastest start-up & small footprint

KumuluzEE generates pure, efficient Java microservices, which start considerably faster and have lower footprint than the majority of other Java microservice frameworks. Microservices are lightweight with small footprint, rapid start and container-ready.

Classical Java EE application server

Part of your application can run as microservices and the rest of the application can continue to operate on your classical Java EE application server. KumuluzEE provides full interoperability for container authentication, CDI, remote EJB calls, remote JMS, data sources, etc.

Configurable environments

KumuluzEE is not a traditional Java EE app server, neither does it embed an app-server inside microservices. It provides configurable Java EE environment inside stand-alone JARs, which only require JRE (Java Runtime Environment) to execute.

Open source

KumuluzEE is fully open, standard-based and extensible.

Zero lock-in

KumuluzEE is not for Java only. In addition to Java it provides support for Node.js and Go, with other languages coming soon.

Extended microservices

KumuluzEE allows you to extend your microservices with common cloud-native patterns.

Full cloud-native architecture and more

KumuluzEE provides full support for configuration, service discovery, health, logging, security, fault tolerance, circuit-breakers, event streaming, metrics, REST patterns and more. KumuluzEE microservices provide first-class support for API gateways. KumuluzEE also supports gRPC, GraphQL, Blockchain, event streaming and many other advanced technologies.

Support for gradual migration

KumuluzEE provides full support for gradual migration of existing Java EE applications to the microservice architecture. Step-by-step migration is supported so that you can extract one or more services from existing application and migrate them to microservices.

Read more about it on [our blog!](#)



ervices for Java EE, you need to take a look at yet simple approach to automating microservices for you.

Harshad Oak

Author of Rightrix Solutions and IndicThreads



KumuluzEE is a great community effort to power Java EE API based microservices, especially on the cloud.

Reza Rahman

Senior Java Technologist, Author, Speaker



KumuluzEE truly brings microservice architecture ready, Fat JAR, modularized dependency, pluggable web container makes it ready to environment. If you are building apps for microservice journey with us, you are in the right place.

Arun Gupta

Vice President, Developer Relations



EXAMPLE CODE

OrderResource.java pom.xml pom.xml (MicroProfile)

```
1  @Path("/orders")
2  @Produces(MediaType.APPLICATION_JSON)
3  @Consumes(MediaType.APPLICATION_JSON)
4  @RequestScoped
5  public class OrdersResource {
6
7      @PersistenceContext(unitName = "books")
8      private EntityManager em;
9
10     @GET
11     @Path("/{id}")
12     public Response getOrder(@PathParam("id") Integer id) {
13
14         BookOrder o = em.find(BookOrder.class, id);
15
16         if (o == null) {
17             return Response.status(Response.Status.NOT_FOUND).build();
18         }
19
20         return Response.ok(o).build();
21     }
22 }
```

Read more about in [tutorials](#).

THEY TRUST US



Österreichische
Sozialversicherung



ABANKA



AdriaticSlovenica d.o.o.
NOSTRUM INVESTMENT BANK



Energetika Ljubljana
Oskrbljujemo s pozitivno energijo!



REPUBLIC OF SLOVENIA
MINISTRY OF PUBLIC
ADMINISTRATION



PBS.
POŠTNA BANKA SLOVENIJE, d.d.
Banka skupina Neve Kredite banke Maribor d.d.



SOLVERA
LYNX



INFORMATIKA



Want to be listed here? Please [contact us](#)!

KumuluzEE

A lightweight open-source micro service framework.

Get started with KumuluzEE



Member of KumuluzDigital, the Kumuluz family of products. Copyright © 2013 - 2019. All rights reserved.

[Kumuluz](#) and KumuluzEE are registered trademarks.

This website uses cookies to improve user experience. By using our website you consent to all cookies.

[I agree](#)