

# alpakka

## Alpakka

(Alpakka) (Reactive) . . . :  
: .  
, Apache Kafka, Amazon S3, Amazon Kinesis, RabbitMQ Sink .  
:  
Akka Streams , Akka Actor .  
: , Sink .  
: . . . .  
:  
.  
: . , .  
: . .  
by chat gpt

Reactive Stream .

Alpakka Documentation  
Version 5.0.0

Java

Overview

Data Transformations

AMQP

Apache Camel

Apache Cassandra

Apache Geode

Apache Kafka

Apache Kudu

Apache Solr

Avro Parquet

AWS EventBridge

AWS DynamoDB

AWS Kinesis and Firehose

AWS Lambda

AWS S3

AWS SNS

AWS SQS

Azure Storage Queue

Couchbase

Elasticsearch

Eventuate

Alpakka Documentation

The [Alpakka project](#) is an open source initiative to implement stream-aware and reactive integrations with [Akka Streams](#), and has been designed from the ground up to understand streaming natively and provide a declarative programming model, with built-in support for backpressure. Akka Streams is a [Reactive Streams](#) and JDK 8 implementation and therefore [fully interoperable](#) with other implementations.

Overview

Versions

Contributing

External Components

Self-contained examples

Other documentation

Integration Patterns

Release Notes

Data Transformations

Parsing Lines

JSON

Compressing/decompressing

Comma-Separated Values - CSV

RecordIO Framing

Text and charsets

Extensible Markup Language - XML

Apache Camel

External library

Apache Cassandra

Artifacts

Sessions

Reading from Cassandra

Writing to Cassandra

Custom Session creation

- <https://doc.akka.io/docs/akka/current/stream/stream-flows-and-basics.html> - Akka Stream API .
-

```

plugins {
    id 'java'
    id 'org.springframework.boot' version '2.7.8-SNAPSHOT'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
}

group = 'com.webnori'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
    maven { url 'https://repo.spring.io/milestone' }
    maven { url 'https://repo.spring.io/snapshot' }
}

dependencies {

    def scalaVersion = '2.13'
    def akkaVersion = '2.7.0'

    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-aop:2.7.0'
    implementation 'org.springframework.boot:spring-boot-starter-validation:2.7.5'

    compileOnly 'org.projectlombok:lombok'
    annotationProcessor 'org.projectlombok:lombok'

    //DB
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'com.mysql:mysql-connector-j'

    //AKKA
    implementation platform("com.typesafe.akka:akka-bom_${scalaVersion}:${akkaVersion}")
    implementation "com.typesafe.akka:akka-actor_${scalaVersion}:${akkaVersion}"
    implementation "com.typesafe.akka:akka-stream_${scalaVersion}"
    implementation "com.typesafe.akka:akka-stream-kafka_${scalaVersion}:4.0.0"

    //Logging
    implementation 'ch.qos.logback:logback-classic:1.2.3'
    implementation "com.typesafe.akka:akka-slf4j_${scalaVersion}:${akkaVersion}"

    //Swagger
    implementation 'org.springdoc:springdoc-openapi-ui:1.6.6'

    //Test
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    testImplementation "org.scalatestplus:junit-4-12_${scalaVersion}:3.3.0.0-SNAP2"
    testImplementation "com.typesafe.akka:akka-testkit_${scalaVersion}:${akkaVersion}"
    testImplementation "com.typesafe.akka:akka-stream-testkit_${scalaVersion}"
    testImplementation "com.typesafe.akka:akka-stream-kafka-testkit_${scalaVersion}:4.0.0"
}

tasks.named('test') {
    useJUnitPlatform()
}

```

- [Alpakka - AWS LocalStack - S3](#)
- [Alpakka Kafka Test -JAVA](#)
- [KafkaToS3WithPaquet](#)

: [ReactiveStreams](#)