What’s new with Apache Camel 3?

Claus Ibsen
Red Hat

Andrea Cosentino
Red Hat
About

Claus Ibsen

- Senior Principal Software Engineer at Red Hat
- ASF Member & Java Champion
- Apache Camel tech lead
- Author of Camel in Action books
- Based in Denmark
- Blog: http://www.davsclaus.com
- Twitter: @davsclaus
- Github: davsclaus
About Andrea Cosentino

- Senior Software Engineer at Red Hat
- ASF Member
- Apache Camel PMC chair
- Apache Camel tech lead
- Based in Italy
- Twitter: @oscerd2 Github: oscerd
Agenda

- What’s Camel
- Camel 3 Projects
- Camel Releases
- Camel K
- Camel Quarkus
- Camel Kafka Connector
- More Material
- Q & A
What’s Camel
What is Apache Camel

The *swiss knife* of integration
What is Apache Camel

The giant swiss knife 😅

Packed with a lot of functionality

The *giant swiss* knife 😊
What is Apache Camel

- Java based integration framework (library)
- Based on Integration Patterns (EIPs)
- Comes with 300+ components (connectors)
- DSL (Java or XML) to describe integration flow (routes)
- Can integrate everything ... almost everything
Runs on popular Java Runtimes

- WildFly
- QUARKUS
- VERT.X
- MICROPROFILE
- Apache Kafka
- Karaf
Camel Community

- The biggest and most active community for open source integration software
- 600+ unique contributors
- 3800+ pull requests (mostly from external contributors)
- 13+ years of development and one of the most active Apache projects
Camel Basics
Camel Routes

from("file:data/inbox")
  .to("jms:queue:order");

<route>
  <from uri="file:data/inbox"/>
  <to uri="jms:queue:order"/>
</route>
Basic Camel Example

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic
Basic Camel Example

```java
public final class CamelBasic {

    public static void main(String[] args) throws Exception {

    }
}

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic
```
Basic Camel Example

```java
public final class CamelBasic {

    public static void main(String[] args) throws Exception {

        // create a CamelContext
        CamelContext camel = new DefaultCamelContext();

    }

}

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic
```
public final class CamelBasic {

    public static void main(String[] args) throws Exception {
        // create a CamelContext
        CamelContext camel = new DefaultCamelContext();

        // add routes which can be inlined as anonymous inner class
        // (to keep all code in a single java file for this basic example)
        camel.addRoutes(new RouteBuilder() {
            @Override
            public void configure() throws Exception {
                from("timer:foo")
                    .log("Hello Camel");
            }
        });
    }
}

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic
Basic Camel Example

```java
public final class CamelBasic {

    public static void main(String[] args) throws Exception {
        // create a CamelContext
        CamelContext camel = new DefaultCamelContext;

        // add routes which can be inlined as anonymous inner class
        // (to keep all code in a single java file for this basic example)
        camel.addRoutes(new RouteBuilder() {
            @Override
            public void configure() throws Exception {
                from("timer:foo")
                    .log("Hello Camel");
            }
        });

        // start is not blocking
        camel.start();

        // so run for 10 seconds
        Thread.sleep(10000);

        // and then stop nicely
        camel.stop();
    }
}

https://github.com/apache/camel-examples/tree/master/examples/camel-example-basic
```
```java
@Component
public class MyCamelRouter extends RouteBuilder {

    // we can use spring dependency injection
    @Autowired
    MyBean myBean;

    @Override
    public void configure() throws Exception {
        // start from a timer
        from("timer:hello?period=\{\{myPeriod\}\}\).routeId("hello")
            // and call the bean
            .bean(myBean, "saySomething")
            // and print it to system out via stream component
            .to("stream:out");
    }
}

https://github.com/apache/camel-spring-boot-examples/tree/master/camel-example-spring-boot
```
Camel Quarkus

```
public class TimerRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("timer:foo?period=1000")
            .log("Hello World");
    }
}
```

https://github.com/apache/camel-quarkus/tree/master/examples/timer-log
@ApplicationScoped
public class TimerRoute extends RouteBuilder {

/** (@code timer.period) is defined in@code src/main/resources/application.properties**/
@ConfigProperty(name = "timer.period", defaultValue = "1000")
String period;

/** An injected bean */
@Inject
Counter counter;

@Override
public void configure() throws Exception {
    fromF("timer:foo?period=", period)
        .setBody(exchange -> "Incremented the counter: " + counter.increment())
        .to("log:example?showExchangePattern=false&showBodyType=false");
}

https://github.com/apache/camel-quarkus/tree/master/examples/timer-log-cdi
Camel 3 Projects
Apache Camel 3 - Projects

Camel

*Swiss knife of integration*
Apache Camel 3 - Projects

Camel
Swiss knife of integration

Camel Spring Boot
Camel on Spring Boot
Apache Camel 3 - Projects

Camel

Swiss knife of integration

Camel Karaf

Camel on Apache Karaf / OSGi

Camel Spring Boot

Camel on Spring Boot
Apache Camel 3 - Projects

- Camel K
  - Camel on Kubernetes & Knative

- Camel
  - Swiss knife of integration

- Camel Karaf
  - Camel on Apache Karaf / OSGi

- Camel Spring Boot
  - Camel on Spring Boot
Apache Camel 3 - Projects

**Camel K**
Camel on Kubernetes & Knative

**Camel**
Swiss knife of integration

**Camel Quarkus**
Optimized JVM & Native compiled Java (GraalVM)

**Camel Karaf**
Camel on Apache Karaf / OSGi

**Camel Spring Boot**
Camel on Spring Boot
Apache Camel 3 - Projects

- **Camel K**
  - Camel on Kubernetes & Knative

- **Camel**
  - Swiss knife of integration

- **Camel Quarkus**
  - Optimized JVM & Native compiled Java (GraalVM)

- **Camel Karaf**
  - Camel on Apache Karaf / OSGi

- **Camel Spring Boot**
  - Camel on Spring Boot

- **Camel Kafka Connector**
  - Kafka Connector with Camel
Camel Releases
<table>
<thead>
<tr>
<th>Release</th>
<th>Date</th>
<th>LTS</th>
<th>LTS EOL</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.25</td>
<td>Jan 2020</td>
<td>Yes</td>
<td>Jun 2021</td>
<td>8</td>
</tr>
<tr>
<td>3.0</td>
<td>Nov 2019</td>
<td>No</td>
<td></td>
<td>8,11</td>
</tr>
<tr>
<td>3.1</td>
<td>Feb 202</td>
<td>No</td>
<td></td>
<td>8,11</td>
</tr>
<tr>
<td>3.2</td>
<td>Apr 2020</td>
<td>No</td>
<td></td>
<td>8,11</td>
</tr>
<tr>
<td>3.3</td>
<td>May 2020</td>
<td>No</td>
<td></td>
<td>8,11</td>
</tr>
<tr>
<td><strong>3.4</strong></td>
<td><strong>Jun 2020</strong></td>
<td><strong>Yes</strong></td>
<td><strong>Jun 2021</strong></td>
<td><strong>8,11</strong></td>
</tr>
<tr>
<td>3.5</td>
<td>Aug 2020</td>
<td>No</td>
<td></td>
<td>8,11,14</td>
</tr>
<tr>
<td>3.6</td>
<td>Oct 2020</td>
<td>No</td>
<td></td>
<td>8,11,14</td>
</tr>
<tr>
<td><strong>3.7</strong></td>
<td><strong>Dec 2020</strong></td>
<td><strong>Yes</strong></td>
<td><strong>Dec 2021</strong></td>
<td><strong>8,11,14</strong></td>
</tr>
</tbody>
</table>

Schedule is subject for change
Camel K
What is Apache Camel K?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.
What is Apache Camel K?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.

Runs on “vanilla” Kubernetes (1) ...
What is Apache Camel K?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.

1. Runs on “vanilla” Kubernetes
2. Runs on OpenShift
What is Apache Camel K?

A lightweight integration platform, based on Apache Camel, born on Kubernetes, with serverless superpowers.

Runs on “vanilla” Kubernetes (1), Openshift (2) and gives its best on a Knative-powered cluster (3)!
How?
How?

1. Create a integration file (Java, Groovy, Kotlin, JS, XML...)

```java
from("telegram:bots/bot-id")
  .transform()...
  .to("kafka:topic");

from("kafka:topic")
  .to("http://my-host/api/path");
```

Camel DSL, based on EIPs...
How?

1. Create a integration file (Java, Groovy, Kotlin, JS, XML...)

   ```camel
camel
dsl
test

   from("telegram:bots/bot-id")
   .transform()...
   .to("kafka:topic");

camel
dsl
test

   from("kafka:topic")
   .to("http:my-host/api/path");
```

2. Run it

   ```bash
   $ kamel run integration.groovy
   ```
How?

1. Create an integration file (Java, Groovy, Kotlin, JS, XML...)

   ```
   from("telegram:bots/bot-id")
     .transform()
     .to("kafka:topic");

   from("kafka:topic")
     .to("http:my-host/api/path");
   ```

2. Run it

   ```
   $ kamel run integration.groovy
   ```

3. Runs on Openshift or Kubernetes

Camel DSL, based on EIPs...
Architecture of Camel K

Dev Environment

Remote Cloud

“Integration” Custom Resource

Camel K Operator

Running Pod

Live updates!

Tailored for cloud-native development experience

Fast redeploy! Less than 1 second!
Quick Camel K Demo
Camel K is not only Java

Table 1. Supported Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Integrations written in Java DSL are supported.</td>
</tr>
<tr>
<td>XML</td>
<td>Integrations written in plain XML DSL are supported.</td>
</tr>
<tr>
<td>YAML</td>
<td>Integrations written in YAML DSL are supported.</td>
</tr>
<tr>
<td>Groovy</td>
<td>Groovy <code>.groovy</code> files are supported (experimental).</td>
</tr>
<tr>
<td>Kotlin</td>
<td>Kotlin Script <code>.kts</code> files are supported (experimental).</td>
</tr>
<tr>
<td>JavaScript</td>
<td>JavaScript <code>.js</code> files are supported (experimental).</td>
</tr>
</tbody>
</table>
Camel
Quarkus
SAY JAVA IS SLOW
ONE MORE TIME
Java Density Problem

<table>
<thead>
<tr>
<th>Node</th>
<th>Node</th>
<th>Node</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Cloud-Native Java Stack</td>
<td>Traditional Cloud-Native Java Stack</td>
<td>Traditional Cloud-Native Java Stack</td>
<td>Traditional Cloud-Native Java Stack</td>
</tr>
<tr>
<td>NodeJS</td>
<td>NodeJS</td>
<td>NodeJS</td>
<td>NodeJS</td>
</tr>
<tr>
<td>NodeJS</td>
<td>NodeJS</td>
<td>NodeJS</td>
<td>NodeJS</td>
</tr>
<tr>
<td>NodeJS</td>
<td>NodeJS</td>
<td>NodeJS</td>
<td>NodeJS</td>
</tr>
</tbody>
</table>

CONTAINER ORCHESTRATION

https://developers.redhat.com/blog/2017/03/14/java-inside-docker/
Java Fast Scaling Problem

SCALE-UP

SCALE-DOWN

SCALE-TO-ZERO
What is Quarkus?
What is Quarkus?

Supersonic Subatomic Java
What is Quarkus?

Supersonic Subatomic Java

A Kubernetes Native Java stack tailored for GraalVM & OpenJDK HotSpot, crafted from the best of breed Java libraries and standards
Minimal Footprint
Apache Camel and Quarkus runs blazingly fast with native compiled binaries. Startup is 24 msec, 50mb binary file, and 27mb RSS memory used.

Compiled natively with GraalVM CE 20.2.0 with JDK 11.0.8 on Mac OSX
Camel
Kafka Connector
What is Apache Kafka?

- Kafka is known as
  - a distributed streaming platform
  - or a pub/sub messaging broker

- It’s an ecosystem
  - Multiple components part of Apache Kafka
  - A lot of 3rd party integrations
  - One of them is Kafka Connect
What is Kafka Connect?

- It is a **framework** that helps you to integrate Kafka with other systems
  - A user can define **source** and **sink** connectors to stream data in/out of Kafka brokers
  - Connectors are **plugable** - you can use one of the many connectors available or write your own
- **Distributed** and **scalable** by default
- **Automatic** **offset** management
- **Simple** transformations
- **Streaming** / **batch** integration
What is Camel Kafka Connector?

- A Kafka Connector built on top of Apache Camel
- Started as an internal proof-of-concept
  - A sub-project of the Apache Camel
  - Donated by Red Hat to the ASF on December 2019
- Reuses in a **simple way** most of the Camel components as Kafka sink and sources
- Latest release is 0.5.0 based on Camel 3.5.0
- Autogenerated documentation and connectors list:
  
  https://camel.apache.org/camel-kafka-connector/latest/connectors.html
Demo S3 to SQS

- S3 Source connector (camel-aws-s3)
- SQS Sink connector (camel-aws-sqs)
- A file loaded on a bucket -> file content in a SQS Queue

To run Kafka on Kubernetes
name=CamelAWSS3SourceConnector
connector.class=org.apache.camel.kafkaconnector.awss3.CamelAwss3SourceConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.camel.kafkaconnector.awss3.converters.S3ObjectConverter
camel.source.maxPollDuration=10000
topics=test1|
camel.source.url=aws-s3://camel-kafka-connector?autocloseBody=false
camel.component.aws-s3.access-key=xxxx
camel.component.aws-s3.secret-key=yyyy
camel.component.aws-s3.region=EU_WEST_1

name=CamelAWSSQSsinkConnector
connector.class=org.apache.camel.kafkaconnector.awssqs.CamelAwssqsSinkConnector
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
topics=test1|
camel.sink.path.queueNameOrArn=camel-1
camel.component.aws-sqs.access-key=xxxx
camel.component.aws-sqs.secret-key=yyyy
camel.component.aws-sqs.region=EU_WEST_1
### Visualizza/Elimina messaggi in camel-1

**Visualizza fino a:** 10 messaggi | **Polling della coda per:** 30 secondi

Polling per ciascun messaggio una volta ogni 2 secondi.

<table>
<thead>
<tr>
<th>Elimina</th>
<th>Corpo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hello from camel-kafka-connector!</td>
</tr>
</tbody>
</table>

**Progresso del polling:** 100%

*Attenzione*: dopo il polling della coda a 0.6 secondi/secondo per 10.2 secondi, i messaggi riportati sopra non sono disponibili per altri consumatori.
More Material
More Material

- Camel website
  https://camel.apache.org
- Quick Camel K demo (2 min video)
  https://www.youtube.com/watch?v=l-q2xt-Qcb8
- Run 100 Camels quickly with Came, Quarkus and GraalVM (10 min video)
  https://www.youtube.com/watch?v=4lXSf8DBQkQ
- Camel Kafka Connector (blog and 25 min video)
  https://developers.redhat.com/blog/2020/05/19/extending-kafka-connectivity-with-apache-camel-kafka-connectors
- Serverless Integration on Kubernetes with Camel K - KubeCon 2020 (35 min video)
  https://www.youtube.com/watch?v=beJOCndVfaU
Examples

- **Camel Examples**
  https://github.com/apache/camel-examples
- **Camel K Examples**
  https://github.com/apache/camel-k/tree/master/example
- **Camel K Examples (detailed as tutorials)**
  https://github.com/apache/camel-k-examples
- **Camel Quarkus Examples (on the way)**
  https://github.com/apache/camel-quarkus-examples
- **Camel Kafka Connector Examples**
  https://github.com/apache/camel-kafka-connector-examples
- **Camel Spring Boot Examples**
  https://github.com/apache/camel-spring-boot-examples
Any Questions?

Follow us on Twitter
@davsclaus
@oscerd2
@ApacheCamel

https://github.com/apache/camel
A star on github is appreciated