# Red Hat build of Apache Camel 4.0

# Migrating to Red Hat build of Apache Camel for Spring Boot

Migrating to Red Hat build of Apache Camel for Spring Boot

# Red Hat build of Apache Camel 4.0 Migrating to Red Hat build of Apache Camel for Spring Boot

Migrating to Red Hat build of Apache Camel for Spring Boot

## Legal Notice

## Abstract

This guide describes the settings for Red Hat build of Apache Camel components.

# Table of Contents

# PREFACE

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

# CHAPTER 1. ABOUT MIGRATION GUIDE

This guide details the changes in the Apache Camel components that you must consider when migrating your application. This guide provides information about following changes.

- Supported Java versions

- Changes to Apache Camel components and deprecated components

- Changes to APIs and deprecated APIs

- Updates to EIP

- Updated to tracing and health checks

# CHAPTER 2. MIGRATING TO APACHE CAMEL 4

This section provides information that can help you migrate your Apache Camel applications from version 3.20 or higher to 4.0. If you are upgrading from an older Camel 3.x release, such as 3.14, see the individual Upgrade guide to upgrade to the 3.20 release, before upgrading to Apache Camel 4.

## 2.1. JAVA VERSIONS

Apache Camel 4 supports Java 17. Support for Java 11 is dropped.

## 2.2. REMOVED COMPONENTS

The following components has been removed:

| Component | Alternative component(s) |
| --- | --- |
| camel-any23 | none |
| camel-atlasmap | none |
| camel-atmos | none |
| camel-caffeine-lrucache | camel-cache, camel-ignite, camel-infinispan |
| camel-cdi | camel-spring-boot, camel-quarkus |
| camel-corda | none |
| camel-directvm | camel-direct |
| camel-dozer | camel-mapstruct |
| camel-elasticsearch-rest | camel-elasticsearch |
| camel-gora | none |
| camel-hbase | none |
| camel-hyperledger-aries | none |
| camel-iota | none |
| camel-ipfs | none |
| camel-jbpm | none |
| camel-jclouds | none |

| Component | Alternative component(s) |
| --- | --- |
| camel-johnzon | camel-jackson, camel-fastjson, camel-gson |
| camel-microprofile-metrics | camel-micrometer, camel-opentelemetry |
| camel-milo | none |
| camel-opentracing | camel-micrometer, camel-opentelemetry |
| camel-rabbitmq | spring-rabbitmq-component |
| camel-rest-swagger | camel-openapi-rest |
| camel-restdsl-swagger-plugin | camel-restdsl-openapi-plugin |
| camel-resteasy | camel-cxf, camel-rest |
| camel-solr | none |
| camel-spark | none |
| camel-spring-integration | none |
| camel-swagger-java | camel-openapi-java |
| camel-websocket | camel-vertx-websocket |
| camel-websocket-jsr356 | camel-vertx-websocket |
| camel-vertx-kafka | camel-kafka |
| camel-vm | camel-seda |
| camel-weka | none |
| camel-xstream | camel-jacksonxml |
| camel-zipkin | camel-micrometer, camel-opentelemetry |

## 2.3. LOGGING

Camel 4 has upgraded logging facade API **slf4j-api** from 1.7 to 2.0.

## 2.4. JUNIT 4

All the **camel-test** modules that were JUnit 4.x based has been removed. All test modules now use JUnit 5.

## 2.5. API CHANGES

Following APIs are deprecated and removed from version 4:

- The **org.apache.camel.ExchangePattern** has removed **InOptionalOut**.

- Removed **getEndpointMap()** method from **CamelContext**.

- Removed **@FallbackConverter** as you should use **@Converter(fallback = true)** instead.

- Removed **uri** attribute on **@EndpointInject**, **@Produce**, and **@Consume** as you should use **value** (default) instead. For example **@Produce(uri = "kafka:cheese")** should be changed to **@Produce("kafka:cheese")**

- Removed **label** on **@UriEndpoint** as you should use **category** instead.

- Removed all **asyncCallback** methods on **ProducerTemplate**. Use **asyncSend** or **asyncRequest** instead.

- Removed **org.apache.camel.spi.OnCamelContextStart**. Use **org.apache.camel.spi.OnCamelContextStarting** instead.

- Removed **org.apache.camel.spi.OnCamelContextStop**. Use **org.apache.camel.spi.OnCamelContextStopping** instead.

- Decoupled the **org.apache.camel.ExtendedCamelContext** from the **org.apache.camel.CamelContext**.

- Replaced **adapt()** from **org.apache.camel.CamelContext** with **getCamelContextExtension**

- Decoupled the **org.apache.camel.ExtendedExchange** from the **org.apache.camel.Exchange**.

- Replaced **adapt()** from **org.apache.camel.ExtendedExchange** with **getExchangeExtension**

- Exchange failure handling status has moved from being a property defined as **ExchangePropertyKey.FAILURE_HANDLED** to a member of the ExtendedExchange, accessible via `isFailureHandled()`method.

- Removed **Discard** and **DiscardOldest** from **org.apache.camel.util.concurrent.ThreadPoolRejectedPolicy**.

- Removed **org.apache.camel.builder.SimpleBuilder**. Was mostly used internally in Camel with the Java DSL in some situations.

- Moved **org.apache.camel.support.IntrospectionSupport** to **camel-core-engine** for internal use only. End users should use **org.apache.camel.spi.BeanInspection** instead.

- Removed **archetypeCatalogAsXml** method from **org.apache.camel.catalog.CamelCatalog**.

- The **org.apache.camel.health.HealthCheck** method **isLiveness** is now default **false** instead of **true**.

- Added **position** method to **org.apache.camel.StreamCache**.

- The method **configure** from the interface **org.apache.camel.main.Listener** was removed

- The **org.apache.camel.support.EventNotifierSupport** abstract class now implements **CamelContextAware**.

- The type for **dumpRoutes** on **CamelContext** has changed from **boolean** to **String** to allow specifying either xml or yaml.
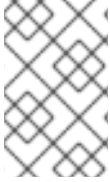
> **NOTE**
>
> The **org.apache.camel.support.PluginHelper** gives easy access to various extensions and context plugins, that was available previously in Camel v3 directly from **CamelContext**.

## 2.6. EIP CHANGES

- Removed **lang** attribute for the **<description>** on every EIPs.

- The **InOnly** and **InOut** EIPs has been removed. Instead, use **SetExchangePattern** or **To** where you can specify exchange pattern to use.

### 2.6.1. Poll Enrich EIP

The polled endpoint URI is now stored as property on the **Exchange** (with key **CamelToEndpoint**) like all other EIPs. Before the URI was stored as a message header.

### 2.6.2. CircuitBreaker EIP

The following options in **camel-resilience4j** was mistakenly not defined as attributes:

| Option |
| --- |
| bulkheadEnabled |
| bulkheadMaxConcurrentCalls |
| bulkheadMaxWaitDuration |
| timeoutEnabled |
| timeoutExecutorService |
| timeoutDuration |
| timeoutCancelRunningFuture |

These options were not exposed in YAML DSL, and in XML DSL you need to migrate from:

```
<circuitBreaker>
  <resilience4jConfiguration>
```

```
        <timeoutEnabled>true</timeoutEnabled>
        <timeoutDuration>2000</timeoutDuration>
     </resilience4jConfiguration>
  ...
  </circuitBreaker>
```

To use following attributes instead:

```
<circuitBreaker>
    <resilience4jConfiguration timeoutEnabled="true" timeoutDuration="2000"/>
  ...
</circuitBreaker>
```

## 2.7. XML DSL

The **<description>** to set a description on a route or node, has been changed from an element to an attribute.

**Example**

Changed from

```
<route id="myRoute">
  <description>Something that this route do</description>
  <from uri="kafka:cheese"/>
  ...
</route>
```

To

[source,xml

```
<route id="myRoute" description="Something that this route do">
  <from uri="kafka:cheese"/>
  ...
</route>
```

## 2.8. TYPE CONVERTER

The **String → java.io.File** converter has been removed.

## 2.9. TRACING

The Tracer and Backlog Tracer no longer includes internal tracing events from routes that was created by Rest DSL or route templates or Kamelets. You can turn this on, by setting **traceTemplates=true** in the tracer.

The Backlog Tracer has been enhanced and *fixed* to trace message headers (also streaming types). This means that previously headers of type **InputStream** was not traced before, but is now included. This could mean that the header stream is positioned at end, and logging the header afterward, may appear as the header value is empty.

## 2.10. USEORIGINALMESSAGE / USEORIGINALBODY

When **useOriginalMessage** or **useOriginalBody** is enabled in **OnException**, **OnCompletion** or error handlers, then the original message body is defensively copied and if possible converted to **StreamCache** to ensure the body can be re-read when accessed. Previously the original body was not converted to **StreamCache** which could lead to the body not able to be read or the stream has been closed.

## 2.11. CAMEL HEALTH

Health checks are now by default only readiness checks out of the box. Camel provides the **CamelContextCheck** as both readiness and liveness checks, so there is at least one of each out of the box. Only consumer based health-checks is enabled by default.

### 2.11.1. Producer Health Checks

The option **camel.health.components-enabled** has been renamed to **camel.health.producers-enabled**.

Some components (in particular AWS) provides also health checks for producers; in Camel 3.x these health checks did not work properly and has been disabled in the source. To continue this behaviour in Camel 4, the producer based health checks are disabled.

Notice that **camel-kafka** comes with producer based health-check that worked in Camel 3, and therefore this change in Camel 4, means that this health-check is disabled.

You **MUST** enable producer health-checks globally, such as in **application.properties**:

```
camel.health.producers-enabled = true
```

## 2.12. JMX

Camel now also include MBeans for **doCatch** and **doFinally** in the tree of processor MBeans.

The **ManagedChoiceMBean** have renamed **choiceStatistics** to **extendedInformation**. The **ManagedFailoverLoadBalancerMBean** have renamed **exceptionStatistics** to **extendedInformation**.

The **CamelContextMBean** and **CamelRouteMBean** has removed method **dumpRouteAsXml(boolean resolvePlaceholders, boolean resolveDelegateEndpoints)**.

## 2.13. YAML DSL

The backwards compatible mode Camel 3.14 or older, which allowed to have *steps* as child to *route* has been removed.

The old syntax:

```
- route:
    from:
      uri: "direct:info"
    steps:
    - log: "message"
```

should be changed to:

```
- route:
    from:
      uri: "direct:info"
      steps:
      - log: "message"
```

## 2.14. BACKLOG TRACING

The option **backlogTracing=true** is now automatically enabled to start the tracer on startup. In the previous versions the tracer was only made available, and had to be manually enabled afterwards. The old behavior can be archived by setting **backlogTracingStandby=true**.

Move the following class from **org.apache.camel.api.management.mbean.BacklogTracerEventMessage** in **camel-management-api** JAR to **org.apache.camel.spi.BacklogTracerEventMessage** in **camel-api** JAR.

The **org.apache.camel.impl.debugger.DefaultBacklogTracerEventMessage** has been refactored into an interface **org.apache.camel.spi.BacklogTracerEventMessage** with some additional details about traced messages. For example Camel now captures a *first* and *last* trace that contains the input and outgoing (if **InOut**) messages.

## 2.15. XML SERIALIZATION

The default xml serialization using **ModelToXMLDumper** has been improved and now uses a generated xml serializer located in the **camel-xml-io** module instead of the JAXB based one from **camel-jaxb**.

## 2.16. OPENAPI MAVEN PLUGIN

The **camel-restdsl-openapi-plugin** Maven plugin now uses **platform-http** as the default rest component in the generated Rest DSL code. Previously the default was servlet. However, platform-http is a better default that works out of the box with Spring Boot and Quarkus.

## 2.17. COMPONENT CHANGES

### 2.17.1. Category

The number of enums for **org.apache.camel.Category** has been reduced from 83 to 37, which means custom components that are using removed values need to choose one of the remainder values. We have done this to consolidate the number of categories of all components in the Camel community.

### 2.17.2. camel-openapi-rest-dsl-generator

This dsl-generator has updated the underlying model classes (**apicurio-data-models**) from 1.1.27 to 2.0.3.

### 2.17.3. camel-atom

The **camel-atom** component has changed the 3rd party atom client from Apache Abdera to RSSReader. This means the feed object is changed from **org.apache.abdera.model.Feed** to **com.apptasticsoftware.rssreader.Item**.

### 2.17.4. camel-azure-cosmosdb

The **itemPartitionKey** has been updated. It's now a String a not a PartitionKey anymore. More details in CAMEL-19222.

### 2.17.5. camel-bean

When using the **method** option to refer to a specific method, and using parameter types and values, such as: **"bean:myBean?method=foo(com.foo.MyOrder, true)"** then any class types must now be using **.class** syntax, i.e. **com.foo.MyOrder** should now be **com.foo.MyOrder.class**.

**Example**

> "bean:myBean?method=foo(com.foo.MyOrder.class, true)"

This also applies to Java types such as String, int.

> "bean:myBean?method=bar(String.class, int.class)"

### 2.17.6. camel-box

Upgraded from Box Java SDK v2 to v4, which have some method signature changes. The method to get a file thumbnail is no longer available.

### 2.17.7. camel-caffeine

The **keyType** parameter has been removed. The Key for the cache will now be only **String** type. More information in CAMEL-18877.

### 2.17.8. camel-fhir

The underlying **hapi-fhir** library has been upgraded from 4.2.0 to 6.2.4. Only the **Delete** API method has changed and now returns **ca.uhn.fhir.rest.api.MethodOutcome** instead of **org.hl7.fhir.instance.model.api.IBaseOperationOutcome**. See hapi-fhir for a more detailed list of underlying changes (only the hapi-fhir client is used in Camel).

### 2.17.9. camel-google

The API based components **camel-google-drive**, **camel-google-calendar**, **camel-google-sheets** and **camel-google-mail** has been upgraded from Google Java SDK v1 to v2 and to latest API revisions. The **camel-google-drive** and **camel-google-sheets** have some API methods changes, but the others are identical as before.

### 2.17.10. camel-http

The component has been upgraded to use Apache HttpComponents v5 which has an impact on how the underlying client is configured. There are 4 different timeouts (**connectionRequestTimeout**, **connectTimeout**, **soTimeout** and **responseTimeout**) instead of initially 3 (**connectionRequestTimeout**, **connectTimeout** and **socketTimeout**) and the default value of some of them has changed so please refer to the documentation for more details.

Please note that the **socketTimeout** has been removed from the possible configuration parameters of **HttpClient**, use **responseTimeout** instead.

Finally, the option **soTimeout** along with any parameters included into **SocketConfig**, need to be prefixed by **httpConnection.**, the rest of the parameters including those defined into **HttpClientBuilder** and **RequestConfig** still need to be prefixed by **httpClient.** like before.

### 2.17.11. camel-http-common

The API in **org.apache.camel.http.common.HttpBinding** has changed slightly to be more reusable. The **parseBody** method now takes in **HttpServletRequest** as input parameter. And all **HttpMessage** has been changed to generic **Message** types.

### 2.17.12. camel-kubernetes

The **io.fabric8:kubernetes-client** library has been upgraded and some deprecated API usage has been removed. Operations previously prefixed with **replace** are now prefixed with **update**.

For example **replaceConfigMap** is now **updateConfigMap**, **replacePod** is now **updatePod** etc. The corresponding constants in class **KubernetesOperations** are also renamed.
**REPLACE_CONFIGMAP_OPERATION** is now **UPDATE_CONFIGMAP_OPERATION**,
**REPLACE_POD_OPERATION** is now **UPDATE_POD_OPERATION** etc.

### 2.17.13. camel-main

The following constants has been moved from **BaseMainSupport** / **Main** to **MainConstants**:

| Old Name | New Name |
| --- | --- |
| Main.DEFAULT_PROPERTY_PLACEHOLDER_LOCATION | MainConstants.DEFAULT_PROPERTY_PLACEHOLDER_LOCATION |
| Main.INITIAL_PROPERTIES_LOCATION | MainConstants.INITIAL_PROPERTIES_LOCATION |
| Main.OVERRIDE_PROPERTIES_LOCATION | MainConstants.OVERRIDE_PROPERTIES_LOCATION |
| Main.PROPERTY_PLACEHOLDER_LOCATION | MainConstants.PROPERTY_PLACEHOLDER_LOCATION |

### 2.17.14. camel-micrometer

The metrics has been renamed to follow Micrometer naming convention .

| Old Name | New Name |
| --- | --- |
| CamelExchangeEventNotifier | camel.exchange.event.notifier |
| CamelExchangesFailed | camel.exchanges.failed |
| CamelExchangesFailuresHandled | camel.exchanges.failures.handled |

| CamelExchangesInflight | camel.exchanges.external.redeliveries |
| --- | --- |
| CamelExchangesSucceeded | camel.exchanges.succeeded |
| CamelExchangesTotal | camel.exchanges.total |
| CamelMessageHistory | camel.message.history |
| CamelRoutePolicy | camel.route.policy |
| CamelRoutePolicyLongTask | camel.route.policy.long.task |
| CamelRoutesAdded | camel.routes.added |
| CamelRoutesRunning | camel.routes.running |

### 2.17.15. camel-jbang

The command **camel dependencies** has been renamed to **camel dependency**.

In Camel CLI the **-dir** parameter for **init** and **run** goal has been renamed to require 2 dashes **--dir** like all the other options.

The **camel stop** command will now by default stop all running integrations (the option **--all** has been removed).

The *Placeholders substitutes* is changed to use **#name** instead of **$name** syntax.

### 2.17.16. camel-openapi-java

The **camel-openapi-java** component has been changed to use **io.swagger.v3** libraries instead of **io.apicurio.datamodels**. As a result, the return type of the public method org.apache.camel.openapi.RestOpenApiReader.read() is now **io.swagger.v3.oas.models.OpenAPI** instead of **io.apicurio.datamodels.openapi.models.OasDocument**. When an OpenAPI 2.0 (swagger) specification is parsed, it is automatically upgraded to OpenAPI 3.0.x by the swagger parser. This version also supports OpenAPI 3.1.x specifications. The related spring-boot starter components have been modified to use the new return type.

### 2.17.17. camel-salesforce

Property names of blob fields on generated DTOs no longer have 'Url' affixed. For example, the **ContentVersionUrl** property is now **ContentVersion**.

### 2.17.18. camel-slack

The default delay (on slack consumer) is changed from 0.5s to 10s to avoid being rate limited to often by Slack.

### 2.17.19. camel-spring-rabbitmq

The option **replyTimeout** in **camel-spring-rabbitmq** has been fixed and the default value from 5 to 30 seconds (this is the default used by Spring).

## 2.18. CAMEL SPRING BOOT

The **camel-spring-boot** dependency no longer includes **camel-spring-xml**. To use legacy Spring XML files **<beans>** with Camel on Spring Boot, then include the **camel-spring-boot-xml-starter** dependency.

### 2.18.1. Graceful Shutdown

Camel now shutdowns a bit later during Spring Boot shutdown. This allows Spring Boot graceful shutdown to complete first (stopping Spring Boot HTTP server gracefully), and then afterward Camel is doing its own Graceful Shutdown.

Technically **camel-spring** has changed **getPhase()** from returning **Integer.MAX_VALUE** to **Integer.MAX_VALUE - 2049**. This gives room for Spring Boot services to shut down first.

### 2.18.2. camel-micrometer-starter

The **uri** tags are now static instead of dynamic (by default), as potential too many tags generated due to URI with dynamic values. This can be enabled again by setting **camel.metrics.uriTagDynamic=true**.

### 2.18.3. camel-platform-http-starter

The **platform-http-starter** has been changed from using **camel-servlet** to use Spring HTTP server directly. Therefore, all the HTTP endpoints are no longer prefixed with the servlet context-path (default is **camel**).

For example:

```
from("platform-http:myservice")
  .to("...")
```

Then calling *myservice* would before require to include the context-path, such as http://localhost:8080/camel/myservice. Now the context-path is not in use, and the endpoint can be called with http://localhost:8080/myservice.

> **NOTE**
>
> The **platform-http-starter** can also be used with Rest DSL.

If the route or consumer is suspended then http status 503 is now returned instead of 404.

### 2.18.4. camel-twitter

The component was updated to use Twitter4j version 4.1.2, which has moved the packages used by a few of its classes. If accessing certain twitter-related data, such as the Twit status, you need to update the packages used from **twitter4j.Status** to **twitter4j.v1.Status**.

# CHAPTER 3. MIGRATING TO APACHE CAMEL 3

This guide provides information on migrating from Red Hat Fuse 7 to Camel 3 on Spring Boot.

**NOTE**

There are important differences between Fuse 7 and Camel 3 in the components, such as modularization and XML Schema changes. See each component section for details.

## 3.1. JAVA VERSIONS

Camel 3 supports Java 17 and Java 11 but not Java 8.

In Java 11 the JAXB modules have been **removed** from the JDK, therefore you will need to add them as Maven dependencies (if you use JAXB such as when using XML DSL or the camel-jaxb component):

```xml
<dependency>
    <groupId>javax.xml.bind</groupId>
    <artifactId>jaxb-api</artifactId>
    <version>2.3.1</version>
</dependency>

<dependency>
    <groupId>com.sun.xml.bind</groupId>
    <artifactId>jaxb-core</artifactId>
    <version>2.3.0.1</version>
</dependency>

<dependency>
    <groupId>com.sun.xml.bind</groupId>
    <artifactId>jaxb-impl</artifactId>
    <version>2.3.2</version>
</dependency>
```

**NOTE** : The Java Platform, Standard Edition 11 Development Kit (JDK 11) is deprecated in Camel Spring Boot 3.x release version and not supported from the further 4.x release versions.

## 3.2. MODULARIZATION OF CAMEL-CORE

In Camel 3.x, **camel-core** has been split into many JARs as follows:

- camel-api

- camel-base

- camel-caffeine-lrucache

- camel-cloud

- camel-core

- camel-jaxp

- camel-main

- camel-management-api

- camel-management

- camel-support

- camel-util

- camel-util-json

Maven users of Apache Camel can keep using the dependency **camel-core** which has transitive dependencies on all of its modules, except for **camel-main**, and therefore no migration is needed.

## 3.3. MODULARIZATION OF COMPONENTS

In Camel 3.x, some of the camel-core components are moved into individual components.

- camel-attachments

- camel-bean

- camel-browse

- camel-controlbus

- camel-dataformat

- camel-dataset

- camel-direct

- camel-directvm

- camel-file

- camel-language

- camel-log

- camel-mock

- camel-ref

- camel-rest

- camel-saga

- camel-scheduler

- camel-seda

- camel-stub

- camel-timer

- camel-validator

- camel-vm

- camel-xpath

- camel-xslt

- camel-xslt-saxon

- camel-zip-deflater

## 3.4. DEFAULT SHUTDOWN STRATEGY

Red Hat build of Apache Camel supports a shutdown strategy using **org.apache.camel.spi.ShutdownStrategy** which is responsible for shutting down routes in a graceful manner. Red Hat build of Apache Camel provides a default strategy in the **org.apache.camel.impl.engine.DefaultShutdownStrategy** to handle the graceful shutdown of the routes.

> **NOTE**
>
> The **DefaultShutdownStrategy** class has been moved from package **org.apache.camel.impl** to **org.apache.camel.impl.engine** in Apache Camel 3.x.

When you configure a simple scheduled route policy to stop a route, the route stopping algorithm is automatically integrated with the graceful shutdown procedure. This means that the task waits until the current exchange has finished processing before shutting down the route. You can set a timeout, however, that forces the route to stop after the specified time, irrespective of whether or not the route has finished processing the exchange.

During graceful shutdown, If you enable the DEBUG logging level on **org.apache.camel.impl.engine.DefaultShutdownStrategy**, then it logs the same inflight exchange information.

```
2015-01-12 13:23:23,656 [- ShutdownTask] INFO DefaultShutdownStrategy - There are 1 inflight
exchanges:
InflightExchange: [exchangeId=ID-test-air-62213-1421065401253-0-3, fromRouteId=route1,
routeId=route1, nodeId=delay1, elapsed=2007, duration=2017]
```

If you do not want to see these logs, you can turn this off by setting the option **logInflightExchangesOnTimeout** to false.

```
context.getShutdownStrategegy().setLogInflightExchangesOnTimeout(false);
```

## 3.5. MULTIPLE CAMELCONTEXTS PER APPLICATION NOT SUPPORTED

Support for multiple CamelContexts has been removed and only one CamelContext per deployment is recommended and supported. The **context** attribute on the various Camel annotations such as **@EndpointInject**, **@Produce**, **@Consume** etc. has therefore been removed.

## 3.6. DEPRECATED APIS AND COMPONENTS

All deprecated APIs and components from Camel 2.x have been removed in Camel 3.

### 3.6.1. Removed components

All deprecated components from Camel 2.x are removed in Camel 3.x, including the old **camel-http**, **camel-hdfs**, **camel-mina**, **camel-mongodb**, **camel-netty**, **camel-netty-http**, **camel-quartz**, **camel-restlet** and **camel-rx** components.

- Removed **camel-jibx** component.

- Removed **camel-boon** dataformat.

- Removed the **camel-linkedin** component as the Linkedin API 1.0 is no longer  supported. Support for the new 2.0 API is tracked by CAMEL-13813.

- The **camel-zookeeper** has its route policy functionality removed, instead use **ZooKeeperClusterService** or the **camel-zookeeper-master** component.

- The **camel-jetty** component no longer supports producer (which has been removed), use **camel-http** component instead.

- The **twitter-streaming** component has been removed as it relied on the deprecated Twitter Streaming API and is no longer functional.

### 3.6.2. Renamed components

Following components are renamed in Camel 3.x.

- The **Camel-microprofile-metrics** has been renamed to  **camel-micrometer**

- The **test** component has been renamed to  **dataset-test** and moved out of  **camel-core** into **camel-dataset** JAR.

- The **http4** component has been renamed to  **http**, and it's corresponding component package from **org.apache.camel.component.http4** to **org.apache.camel.component.http**. The supported schemes are now only **http** and **https**.

- The **hdfs2** component has been renamed to  **hdfs**, and it's corresponding component package from **org.apache.camel.component.hdfs2** to **org.apache.camel.component.hdfs**. The supported scheme is now **hdfs**.

- The **mina2** component has been renamed to  **mina**, and it's corresponding component package from **org.apache.camel.component.mina2** to **org.apache.camel.component.mina**. The supported scheme is now **mina**.

- The **mongodb3** component has been renamed to  **mongodb**, and it's corresponding component package from **org.apache.camel.component.mongodb3** to **org.apache.camel.component.mongodb**. The supported scheme is now **mongodb**.

- The **netty4-http** component has been renamed to  **netty-http**, and it's corresponding component package from **org.apache.camel.component.netty4.http** to **org.apache.camel.component.netty.http**. The supported scheme is now **netty-http**.

- The **netty4** component has been renamed to  **netty**, and it's corresponding component package from **org.apache.camel.component.netty4** to **org.apache.camel.component.netty**. The supported scheme is now **netty**.

- The **quartz2** component has been renamed to **quartz**, and it's corresponding component package from **org.apache.camel.component.quartz2** to **org.apache.camel.component.quartz**. The supported scheme is now **quartz**.

- The **rxjava2** component has been renamed to **rxjava**, and it's corresponding component package from **org.apache.camel.component.rxjava2** to **org.apache.camel.component.rxjava**.

- Renamed **camel-jetty9** to **camel-jetty**. The supported scheme is now **jetty**.

## 3.7. CHANGES TO CAMEL COMPONENTS

### 3.7.1. Mock component

The **mock** component has been moved out of **camel-core**. Because of this a number of methods on its *assertion clause builder* are removed.

### 3.7.2. ActiveMQ

If you are using the **activemq-camel** component, then you should migrate to use **camel-activemq** component, where the component name has changed from **org.apache.activemq.camel.component.ActiveMQComponent** to **org.apache.camel.component.activemq.ActiveMQComponent**.

### 3.7.3. AWS

The component **camel-aws** has been split into multiple components:

- camel-aws-cw

- camel-aws-ddb (which contains both ddb and ddbstreams components)

- camel-aws-ec2

- camel-aws-iam

- camel-aws-kinesis (which contains both kinesis and kinesis-firehose components)

- camel-aws-kms

- camel-aws-lambda

- camel-aws-mq

- camel-aws-s3

- camel-aws-sdb

- camel-aws-ses

- camel-aws-sns

- camel-aws-sqs

- camel-aws-swf

> **NOTE**
>
> It is recommended to add specific dependencies for these components.

### 3.7.4. Camel CXF

The **camel-cxf** JAR has been divided into SOAP vs REST and Spring and non Spring JARs. It is recommended to choose the specific JAR from the following list when migrating from **came-cxf**.

- **camel-cxf-soap**

- **camel-cxf-spring-soap**

- **camel-cxf-rest**

- **camel-cxf-spring-rest**

- **camel-cxf-transport**

- **camel-cxf-spring-transport**

For example, if you were using CXF for SOAP and with Spring XML, then select **camel-cxf-spring-soap** and **camel-cxf-spring-transport** when migrating from **camel-cxf**.

When using Spring Boot, choose from the following starter when you migrate from **camel-cxf-starter** to SOAP or REST:

- **camel-cxf-soap-starter**

- **camel-cxf-rest-starter**

#### 3.7.4.1. Camel CXF changed namespaces

The **camel-cxf** XML XSD schemas has also changed namespaces.

Table 3.1. Changes to namespaces

| Old Namespace | New Namespace |
| --- | --- |
| **http://camel.apache.org/schema/cxf** | **http://camel.apache.org/schema/cxf/jaxws** |
| **http://camel.apache.org/schema/cxf/camel-cxf.xsd** | **http://camel.apache.org/schema/cxf/jaxws/camel-cxf.xsd** |
| **http://camel.apache.org/schema/cxf** | **http://camel.apache.org/schema/cxf/jaxrs** |
| **http://camel.apache.org/schema/cxf/camel-cxf.xsd** | **http://camel.apache.org/schema/cxf/jaxrs/camel-cxf.xsd** |

The **camel-cxf** SOAP component is moved to a new **jaxws** sub-package, that is, **org.apache.camel.component.cxf** is now **org.apache.camel.component.cxf.jaws**. For example, the **CxfComponent** class is now located in **org.apache.camel.component.cxf.jaxws**.

### 3.7.5. FHIR

The camel-fhir component has upgraded it's hapi-fhir dependency to 4.1.0. The default FHIR version has been changed to R4. Therefore, if DSTU3 is desired it has to be explicitly set.

### 3.7.6. Kafka

The **camel-kafka** component has removed the options **bridgeEndpoint** and **circularTopicDetection** as this is no longer needed as the component is acting as bridging would work on Camel 2.x. In other words **camel-kafka** will send messages to the topic from the endpoint uri. To override this use the **KafkaConstants.OVERRIDE_TOPIC** header with the new topic. See more details in the **camel-kafka** component documentation.

### 3.7.7. Telegram

The **camel-telegram** component has moved the authorization token from uri-path to a query parameter instead, e.g. migrate

```
telegram:bots/myTokenHere
```

to

```
telegram:bots?authorizationToken=myTokenHere
```

### 3.7.8. JMX

If you run Camel standalone with just **camel-core** as a dependency, and you want JMX enabled out of the box, then you need to add **camel-management** as a dependency.

For using **ManagedCamelContext** you now need to get this extension from **CamelContext** as follows:

```
ManagedCamelContext managed = camelContext.getExtension(ManagedCamelContext.class);
```

### 3.7.9. XSLT

The XSLT component has moved out of camel-core into **camel-xslt** and **camel-xslt-saxon**. The component is separated so **camel-xslt** is for using the JDK XSTL engine (Xalan), and **camel-xslt-saxon** is when you use Saxon. This means that you should use **xslt** and **xslt-saxon** as component name in your Camel endpoint URIs. If you are using XSLT aggregation strategy, then use **org.apache.camel.component.xslt.saxon.XsltSaxonAggregationStrategy** for Saxon support. And use **org.apache.camel.component.xslt.saxon.XsltSaxonBuilder** for Saxon support if using xslt builder. Also notice that **allowStax** is also only supported in **camel-xslt-saxon** as this is not supported by the JDK XSLT.

### 3.7.10. XML DSL Migration

The XML DSL has been changed slightly.

The custom load balancer EIP has changed from **<custom>** to **<customLoadBalancer>**

The XMLSecurity data format has renamed the attribute **keyOrTrustStoreParametersId** to **keyOrTrustStoreParametersRef** in the **<secureXML>** tag.

The **<zipFile>** data format has been renamed to  **<zipfile>**.

## 3.8. MIGRATING CAMEL MAVEN PLUGINS

The **camel-maven-plugin** has been split up into two maven plugins:

**camel-maven-plugin**

camel-maven-plugin has the **run** goal, which is intended for quickly running Camel applications standalone. See https://camel.apache.org/manual/camel-maven-plugin.html for more information.

**camel-report-maven-plugin**

The **camel-report-maven-plugin** has the **validate** and **route-coverage** goals which is used for generating reports of your Camel projects such as validating Camel endpoint URIs and route coverage reports, etc. See https://camel.apache.org/manual/camel-report-maven-plugin.html for more information.