Red Hat build of Apache Camel 4.4

迁移到 Red Hat build of Apache Camel for Spring Boot

迁移到 Red Hat build of Apache Camel for Spring Boot

Last Updated: 2024-06-25

Red Hat build of Apache Camel 4.4 迁移到 Red Hat build of Apache Camel for Spring Boot

迁移到 Red Hat build of Apache Camel for Spring Boot

法律通告

摘要

本指南描述了红帽构建的 Apache Camel 组件的设置。

目录

前言

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

第 1 章 关于迁移指南

本指南详细介绍了迁移应用程序时必须考虑的 Apache Camel 组件中的更改。本指南提供有关以下更改的信息。

- 支持的 Java 版本
- 对 Apache Camel 组件的更改和已弃用的组件
- 对 API 和已弃用的 API 的更改
- EIP 更新
- 更新至追踪和健康检查

第 2 章 迁移到 APACHE CAMEL 4

本节提供了可帮助您将 Apache Camel 应用程序从 3.20 或更高版本迁移到 4.0 的信息。如果您要从旧的 Camel 3.x 版本（如 3.14）升级，请参阅 单个 升级指南 来升级到 3.20 版本，然后再升级到 Apache Camel 4。

注意

本发行版本的迁移指南中的信息不适用于 IBM Power 和 IBM Z。以后的发行版本中应该会改变。

2.1. JAVA 版本

Apache Camel 4 支持 Java 17。丢弃了对 Java 11 的支持。

2.2. 删除的组件

以下组件已被删除：

组件	其他组件
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

组件	其他组件
camel-jbpm	none
camel-jclouds	none
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. 日志记录

Camel 4 将日志 facade API **slf4j-api** 从 1.7 升级到 2.0。

2.4. JUNIT 4

所有基于 JUnit 4.x 的 **camel-test** 模块已被删除。所有测试模块现在都使用 JUnit 5。

2.5. API 更改

以下 API 已被弃用并从版本 4 中删除：

- **org.apache.camel.ExchangePattern** 已删除了 **InOptionalOut**。
- 从 **CamelContext** 中移除 **getEndpointMap （）** 方法。
- 删除了 **@FallbackConverter**，因为您应该改为使用 **@Converter (fallback = true)**。
- removed **uri** 属性 **@EndpointInject**,**@Produce**, 和 **@Consume** 替代 值 （默认）。例如，**@Produce (uri = "kafka:cheese")** 应更改为 **@Produce ("kafka:cheese")**
- 删除了 **@UriEndpoint 上的标签**，因为您应该使用 **类别**。
- 删除了 **ProducerTemplate** 上的所有 **asyncCallback** 方法。改为使用 **asyncSend** 或 **asyncRequest**。
- Removed **org.apache.camel.spi.OnCamelContextStart**.改为使用 **org.apache.camel.spi.OnCamelContextStarting**。
- Removed **org.apache.camel.spi.OnCamelContextStop**.改为使用 **org.apache.camel.spi.OnCamelContextStopping**。
- 将 **org.apache.camel.ExtendedCamelContext** 与 **org.apache.camel.CamelContext** 分离。
- 使用 **getCamelContextExtension**替换 **org.apache.camel.CamelContext** 的 **adapt （）**
- 将 **org.apache.camel.ExtendedExchange** 与 **org.apache.camel.Exchange** 分离。
- 使用 **getExchangeExtension**替换 **org.apache.camel.ExtendedExchange** 的 **adapt （）**
- 交换失败处理状态已从定义为 **ExchangePropertyKey.FAILURE_HANDLED** 的属性移到 ExtendedExchange 的成员，可通过 'isFailureHandled （）'method 访问。
- 从 **org.apache.camel.util.concurrent.ThreadPoolRejectedPolicy** 中删除了 **Discard** 和 **DiscardOldest**。
- 删除了 **org.apache.camel.builder.SimpleBuilder**。在某些情况下，通常在 Camel 内部使用 Java DSL。
- 将 **org.apache.camel.support.IntrospectionSupport** 设置为 **camel-core-engine**，仅供内部使用。最终用户应使用 **org.apache.camel.spi.BeanInspection** 替代。
- 从 **org.apache.camel.catalog.CamelCatalog** 中删除了 **archetypeCatalogAsXml** 方法。
- **org.apache.camel.health.HealthCheck** 方法现在默认为 **false**，而不是 **true**。
- **向 org.apache.camel.StreamCache 添加了 位置 方法。**

- 从接口 org.apache.camel.main.Listener 配置 的方法已被删除

- org.apache.camel.support.EventNotifierSupport abstract 类现在实现了 CamelContextAware。

- CamelContext 上的 dumpRoutes 的类型已从 布尔值 改为 String，以允许指定 xml 或 yaml。

注意

org.apache.camel.support.PluginHelper 提供对以前来自 CamelContext 的 Camel v3 中可用的各种扩展和上下文插件的简单访问。

2.6. EIP 更改

- 每个 EIPs 上删除了 &lt ;description> 的 lang 属性。

- InOnly 和 InOut EIPs 已被删除。反之，使用 SetExchangePattern 或 to 指定要使用的交换模式。

2.6.1. poll Enrich EIP

轮询的端点 URI 现在作为属性存储在 Exchange 上（带有键 CamelToEndpoint），与其他 EIPs 一样。在 URI 作为消息标头存储之前。

2.6.2. CircuitBreaker EIP

camel-resilience4j 中的以下选项被错误地定义为属性：

选项
bulkheadEnabled
bulkheadMaxConcurrentCalls
bulkheadMaxWaitDuration

timeoutEnabled
timeoutExecutorService
timeoutDuration
timeoutCancelRunningFuture

这些选项没有在 **YAML DSL 中公开，在您需要迁移的 XML DSL 中：**

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

使用以下属性：

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

2.7. XML DSL

在路由或节点上设置描述的 <description> 已从元素改为一个属性。

Example

从中修改

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

to

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

2.8. 类型 CONVERTER

String → java.io.File converter 已被删除。

2.9. TRACING

Tracer 和 Backlog Tracer 不再包含由 Rest DSL 或 route templates 或 Kamelets 创建的路由的内部追踪事件。您可以通过在 tracer 中设置 traceTemplates=true 来打开它。

Backlog Tracer 已被改进，并 *修复了* trace 消息标头（也流类型）。这意味着之前没有跟踪 InputStream 类型的标头，但现在包含。这可能意味着标头流在结尾处，并在后记录标头后可能会出现，因为标头值为空。

2.10. USEORIGINALMESSAGE / USEORIGINALBODY

当在 OnException、OnCompletion 或错误处理程序中启用 useOriginalMessage 或 useOriginalBody 时，原始消息正文会完全复制，如果可能转换为 StreamCache，以确保正文可以在访问时重新读取。在以前的版本中，原始正文没有转换为 StreamCache，这可能会导致正文无法读取或关闭流。

2.11. CAMEL HEALTH

现在，健康检查只会开箱即用的就绪度检查。Camel 提供 CamelContextCheck 作为就绪度和存活度检查，因此开箱即用至少一个。默认只启用基于消费者的健康检查。

2.11.1. 制作者健康检查

camel.health.components-enabled 选项已重命名为 camel.health.producers-enabled。

有些组件（特别是 AWS）也为生成者提供健康检查；在 Camel 3.x 中，这些健康检查无法正常工作，并在源中禁用。要在 Camel 4 中继续此行为，基于生成者的健康检查被禁用。

请注意，camel-kafka 附带基于制作者的健康检查，在 Camel 3 中工作，因此 Camel 4 中的这个更改意味着这个健康检查被禁用。

您必须在全局范围内启用制作者健康检查，例如在 application.properties 中：

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

2.12. JMX

Camel 现在还包括 doCatch 的 MBeans，并在处理器 MBeans 的树中进行 最后。

ManagedChoiceMBean 已将 choiceStatistics 重命名为 extendedInformation。ManagedFailoverLoadBalancerMBean 将 exceptionStatistics 重命名为 extendedInformation。

CamelContextMBean 和 CamelRouteMBean 删除了方法 dumpRouteAsXml （布尔值 resolvePlaceholders，布尔值 resolveDelegateEndpoints）。

2.13. YAML DSL

向后兼容模式 Camel 3.14 或更早版本，允许作为 *路由* 子级 *步骤* 已被删除。

旧语法：

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

应更改为：

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

2.14. BACKLOG TRACING

选项 backlogTracing=true 现在会自动启用来在启动时启动 tracer。在以前的版本中，tracer 仅可用，之后必须手动启用。可以通过设置 backlogTracingStandby=true 来归档旧行为。

将以下类从 camel-management-api JAR 中的 org.apache.camel.api.management.mbean.BacklogTracerEventMessage 移到 org.apache.camel.spi.BacklogTracerEventMessage。

org.apache.camel.impl.debugger.DefaultBacklogTracerEventMessage 已重构为接口 org.apache.camel.spi.BacklogTracerEventMessage，其中包含有关 traced 消息的一些额外详情。例如，Camel 现在捕获包含输入和输出（如果为 InOut）消息的 *第一个和最后一个* trace。

2.15. XML 序列化

使用 ModelToXMLDumper 的默认 xml 序列化已被改进，现在使用 camel-xml-io 模块中的生成的 xml serializer，而不是从 camel-jaxb 的一个 JAXB。

2.16. OPENAPI MAVEN 插件

camel-restdsl-openapi-plugin Maven 插件现在使用 platform-http 作为生成的 Rest DSL 代码中的默认其余组件。在以前的版本中，默认为 servlet。但是，platform-http 是一个更好的默认值，可用于 Spring Boot 和 Quarkus。

2.17. 组件更改

2.17.1. 类别

org.apache.camel.Category 的枚举数量已从 83 减少到 37，这意味着使用删除的值的自定义组件需要选择一个剩余的值。我们这样做是为了整合 Camel 社区中所有组件的类别。

2.17.2. camel-openapi-rest-dsl-generator

此 dsl-generator 已将底层模型类(apicurio-data-models)从 1.1.27 更新至 2.0.3。

2.17.3. camel-atom

camel-atom 组件已将 Apache Abdera 的第三方从 Apache Abdera 改为 RSSReader。这意味着源对象已从 org.apache.abdera.model.Feed 改为 com.apptasticsoftware.rssreader.Item。

2.17.4. camel-azure-cosmosdb

itemPartitionKey 已更新。现在，一个字符串 a not a PartitionKey。CAMEL-19222 中的更多详细信息。

2.17.5. camel-bean

当使用 method 选项引用特定方法时，并使用参数类型和值，例如 ："bean:myBean?method=foo(com.foo.MyOrder, true) "， 任何类类型现在都必须使用 .class 语法，即 com.foo.MyOrder.MyOrder。

Example

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

这也适用于 Java 类型，如 String、int。

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

2.17.6. camel-box

从 Box Java SDK v2 升级到 v4，其有一些方法签名更改。获取文件缩略图的方法不再可用。

2.17.7. camel-caffeine

keyType 参数已被删除。缓存的密钥现在仅是 String 类型。CAMEL-18877 中的更多信息。

2.17.8. camel-fhir

底层 hapi-fhir 库已从 4.2.0 升级到 6.2.4。只有 Delete API 方法已更改，现在返回 ca.uhn.fhir.rest.api.MethodOutcome 而不是

org.hl7.fhir.instance.model.api.IBaseOperationOutcome。有关底层更改的详细列表（只在 Camel 中使用 hapi-fhir 客户端，请参阅 hapi-fhir 客户端）。

2.17.9. camel-google

基于 API 的组件 camel-google-drive,camel-google-calendar,camel-google-sheets 和 camel-google-mail 已从 Google Java SDK v1 升级到 v2，以及最新的 API 修订。camel-google-drive 和 camel-google-sheets 有一些 API 方法更改，但其他方法与之前相同。

2.17.10. camel-http

组件已升级至使用 Apache HttpComponents v5，这会影响底层客户端的配置方式。有 4 个不同的超时(connectionRequestTimeout , connect Timeout,so Timeout, soTimeout)而不是最初 3 (connectionRequestTimeout,connectTimeout, socketTimeout)，以及其中一些的默认值已更改，因此请参阅文档了解更多详情。

请注意，socketTimeout 已从 HttpClient 的可能配置参数中删除，改为使用 responseTimeout。

最后，选项 soTimeout 以及 SocketConfig 中包含的任何参数，需要以 httpConnection. 前缀。（包括定义到 HttpClientBuilder 和 RequestConfig 的参数等）需要加上 httpClient 前缀。

2.17.11. camel-http-common

org.apache.camel.http.common.HttpBinding 中的 API 稍微更改为可重复利用。parseBody 方法现在使用 HttpServletRequest 作为输入参数。所有 HttpMessage 已更改为通用 消息类型。

2.17.12. camel-kubernetes

io.fabric8:kubernetes-client 库已被升级，一些已弃用的 API 用量已被删除。之前带有 replace 前缀的操作现在带有 update 前缀。

例如，replaceConfigMap 现在是 updateConfigMap，replacePod 现在为 updatePod 等。类 KubernetesOperations 中的对应常量也被重命名。REPLACE_CONFIGMAP_OPERATION 现在是 UPDATE_CONFIGMAP_OPERATION，REPLACE_POD_OPERATION 现在为 UPDATE_POD_OPERATION 等。

2.17.13. camel-main

*以下常数已从 **BaseMainSupport / Main to Main Constants** 移动：*

旧名称	新名称
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

*指标已被重命名为遵循 **Micrometer** 命名约定。*

旧名称	新名称
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

camel 依赖项 命令已被重命名为 camel 依赖项。

在 Camel CLI 中，init 和 run 目标的 -dir 参数已被重命名为需要 2 dashes --dir，与所有其他选项一样。

camel stop 命令现在默认停止所有正在运行的集成（删除了 --all 选项）。

Placeholders 替换 被修改为使用 #name 而不是 $name 语法。

2.17.16. camel-openapi-java

camel-openapi-java 组件已更改为使用 io.swagger.v3 库，而不是 io.apicurio.datamodels。因此，公共方法 org.apache.camel.openapi.RestOpenApiReader.read （）的返回类型是 io.swagger.v3.oas.models.OpenAPI，而不是 io.apicurio.datamodels.openapi.models.OasDocument。当解析 OpenAPI 2.0 (swagger)规格时，它由 swagger parser 自动升级到 OpenAPI 3.0.x。此版本还支持 OpenAPI 3.1.x 规格。相关的 spring-boot 入门组件已被修改为使用新的返回类型。

2.17.17. camel-salesforce

生成的 DTOs 上的 blob 字段的属性名称不再有 'Url' affixed。例如，ContentVersionUrl 属性现在是 ContentVersion。

2.17.18. camel-slack

默认延迟（在 slack 消费者上）从 0.5s 改为 10s，以避免被 Slack 的速率限制。

2.17.19. camel-spring-rabbitmq

camel-spring-rabbitmq 中的选项 replyTimeout 已被修复，默认值从 5 到 30 秒（这是 Spring 使用的默认值）。

2.18. CAMEL SPRING BOOT

camel-spring-boot 依赖项不再包含 camel-spring-xml。要使用传统的 Spring XML 文件 &lt

;beans& gt;，在 Spring Boot 上带有 Camel，然后包含 camel-spring-boot-xml-starter 依赖项。

2.18.1. 正常关闭

Camel 现在会在 Spring Boot 关闭过程中关闭一些时间。这允许 Spring Boot graceful shutdown 首先完成（正常停止 Spring Boot HTTP 服务器），然后在 Camel 执行自己的 Graceful Shutdown 后。

从技术上 camel-spring 已将 getPhase （） 从 Integer.MAX_VALUE 返回到 Integer.MAX_VALUE - 2049。这为 Spring Boot 服务提供了先关机的空间。

2.18.2. camel-micrometer-starter

uri 标签现在是静态的，而不是动态标签（默认情况下，因为 URI 使用动态值生成的标签太多）。这可以通过设置 camel.metrics.uriTagDynamic=true 来再次启用。

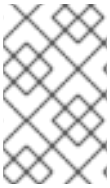
2.18.3. camel-platform-http-starter

platform-http-starter 已从使用 camel-servlet 改为直接使用 Spring HTTP 服务器。因此，所有 HTTP 端点不再以 servlet context-path 前缀（默认为 camel）。

例如：

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

然后，在需要包括 context-path 之前调用 myservice 才能包括 context-path，如 http://localhost:8080/camel/myservice。现在，context-path 没有被使用，端点可以使用 http://localhost:8080/myservice 调用。



注意

platform-http-starter 也可以用于 Rest DSL。

如果路由或消费者被暂停，则 http status 503 现在会返回，而不是 404。

2.18.4. camel-twitter

组件已更新为使用 Twitter4j 版本 4.1.2，它已移动了几个类使用 的软件包。如果访问某些与二者相关的数据，如 Twit 状态，您需要将从 twitter4j.Status 中使用的软件包更新为 twitter4j.v1.Status。

第 3 章 迁移到 APACHE CAMEL 3

本指南提供有关在 Spring Boot 上从 Red Hat Fuse 7 迁移到 Camel 3 的信息。

注意

在组件（如模块化和 XML 架构更改）中，Fuse 7 和 Camel 3 之间有重要的区别。详情请查看每个组件部分。

3.1. JAVA 版本

Camel 3 支持 Java 17 和 Java 11，但不支持 Java 8。

在 Java 11 中，JAXB 模块已从 JDK 中删除，因此您需要将它们添加为 Maven 依赖项（如果使用 JAXB 时，如使用 XML DSL 或 camel-jaxb 组件）：

```
<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>
```

注意 ：Java Platform，标准版本 11 开发套件(JDK 11)在 Camel Spring Boot 3.x 版本中已弃用，且不受进一步的 4.x 发行版本的支持。

3.2. CAMEL-CORE 的模块化

在 Camel 3.x 中，camel-core 已被分成多个 JAR，如下所示：

- *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*

Apache Camel 的 Maven 用户可以继续使用依赖项 camel-core，它对其所有模块有传输依赖项，但 camel-main 除外，因此不需要迁移。

3.3. 组件的模块化

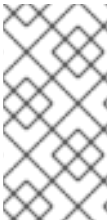
在 Camel 3.x 中，一些 camel-core 组件被移到各个组件中。

- *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. 默认关闭策略

红帽构建的 Apache Camel 支持使用 org.apache.camel.spi.ShutdownStrategy 的关闭策略，它负责以安全的方式关闭路由。红帽构建的 Apache Camel 在 org.apache.camel.impl.engine.DefaultShutdownStrategy 中提供默认策略，以处理路由的安全关闭。



注意

DefaultShutdownStrategy 类已从软件包 org.apache.camel.impl 移到 Apache Camel 3.x 中的 org.apache.camel.impl.engine。

当您配置一个简单的调度路由策略来停止路由时，路由停止算法会自动与安全关闭过程集成。这意味着，该任务在关闭路由前等待当前交换完成处理。您可以设置超时，它会强制路由在指定时间后停止，无论路由是否完成处理交换。

在安全关闭过程中，如果您在 org.apache.camel.impl.engine.DefaultShutdownStrategy 上启用 DEBUG 日志级别，则它会记录相同的 inflight Exchange 信息。

```
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]
```

如果您不想查看这些日志，您可以通过将 logInflightExchangesOnTimeout 选项设置为 false 来关闭这个日志。

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

3.5. 不支持每个应用程序有多个 CAMELCONTEXTS

对多个 CamelContexts 的支持已被删除，建议每个部署只有一个 CamelContext，并被支持。因此，各种 Camel 注释上的 context 属性（如 @EndpointInject、@Produce、@Consume 等）已被删除。

3.6. 弃用的 API 和组件

Camel 3 中删除了来自 Camel 2.x 的所有已弃用的 API 和组件。

3.6.1. 删除的组件

Camel 2.x 中的所有已弃用的组件都在 Camel 3.x 中删除，包括旧的 camel-http, camel-hdfs, camel-mina, camel-mongodb, camel-netty, camel-netty-http, camel-quartz, camel-restlet 和 camel-rx 组件。

- 删除了 camel-jibx 组件。
- 删除了 camel-boon 数据格式。
- 删除了 camel-linkedin 组件，因为 不再支持 Linkedin API 1.0。对新 2.0 API 的支持由 CAMEL-13813 跟踪。
- camel-zookeeper 删除了其路由策略功能，而是使用 ZooKeeperClusterService 或 camel-zookeeper-master 组件。
- camel-jetty 组件不再支持制作者（已被删除），改为使用 camel-http 组件。
- Twitter -streaming 组件已被删除，因为它依赖于已弃用的 Twitter Streaming API，不再可以正常工作。

3.6.2. 重命名组件

在 Camel 3.x 中重命名以下组件。

- camel -microprofile-metrics 已重命名为 camel-micrometer
- 测试 组件已重命名为 dataset-test，并从 camel-core 移到 camel-dataset JAR 中。
- http4 组件已重命名为 http，它对应于从 org.apache.camel.component.http4 到 org.apache.camel.component.http 的组件软件包。现在，支持的方案只能是 http 和 https。
- hdfs2 组件已重命名为 hdfs，它对应于从 org.apache.camel.component.hdfs2 到 org.apache.camel.component.hdfs 的组件软件包。现在支持的方案是 hdfs。
- mina2 组件已重命名为 mina，它对应于来自从 org.apache.camel.component.mina2 到 org.apache.camel.component.mina 2 的软件包。现在支持的方案是 mina。
-

mongodb3 组件已重命名为 mongodb，它对应于从 org.apache.camel.component.mongodb3 到 org.apache.camel.component.mongodb 的组件软件包。现在支持的方案是 mongodb。

- *netty4-http 组件已重命名为 netty-http，它对应于从 org.apache.camel.component.netty4.http 到 org.apache.camel.component.netty.http 的组件软件包。现在支持的方案是 netty-http。*
- *netty4 组件已重命名为 netty，它对应于从 org.apache.camel.component.netty4 到 org.apache.camel.component.netty 的组件软件包。现在支持的方案是 netty。*
- *quartz2 组件已重命名为 quartz，它对应于从 org.apache.camel.component.quartz2 到 org.apache.camel.component.quartz 的组件软件包。现在支持的方案是 quartz。*
- *rxjava2 组件已重命名为 rxjava，它对应于从 org.apache.camel.component.rxjava2 到 org.apache.camel.component.rxjava 的组件软件包。*
- *将 camel-jetty9 重命名为 camel-jetty。现在，支持的方案是 jetty。*

3.7. CAMEL 组件的更改

3.7.1. Mock 组件

mock 组件已从 camel-core 移出。由于其 assertion 子句构建器 上的许多方法已被删除。

3.7.2. ActiveMQ

如果您使用 activemq-camel 组件，则应迁移到使用 camel-activemq 组件，其中组件名称已从 org.apache.activemq.camel.component.ActiveMQComponent 改为 org.apache.camel.component.activemq.ActiveMQComponent。

3.7.3. AWS

组件 camel-aws 被分成多个组件：

- *camel-aws-cw*
- *camel-aws-ddb （包含 ddb 和 ddbstreams 组件）*
- *camel-aws-ec2*
- *camel-aws-iam*
- *camel-aws-kinesis （其中包含 kinesis 和 kinesis-firehose 组件）*
- *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*

注意

建议为这些组件添加特定的依赖项。

3.7.4. Camel CXF

camel-cxf JAR 已分为 SOAP 与 REST 和 Spring JAR。当从 came-cxf 进行迁移时，建议从以下列表中选择特定的 JAR。

- camel-cxf-soap
- camel-cxf-spring-soap
- camel-cxf-rest
- camel-cxf-spring-rest
- camel-cxf-transport
- camel-cxf-spring-transport

例如，如果您使用 CXF 用于 SOAP 并使用 Spring XML，那么在从 camel-cxf 进行迁移时，请选择 camel-cxf-spring-soap 和 camel-cxf-spring-transport。

使用 Spring Boot 时，当您从 camel-cxf-starter 迁移到 SOAP 或 REST 时，从以下入门中选择：

- camel-cxf-soap-starter
- camel-cxf-rest-starter

3.7.4.1. Camel CXF 更改命名空间

camel-cxf XML XSD 模式也更改了命名空间。

表 3.1. 对命名空间的更改

旧命名空间	新命名空间
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

camel-cxf SOAP 组件被移到一个新的 jaxws 子软件包，即 org.apache.camel.component.cxf 现在是 org.apache.camel.component.cxf.jaws。例如， CxfComponent 类现在位于 org.apache.camel.component.cxf.jaxws。

3.7.5. FHIR

camel-fhir 组件已将其 hapi-fhir 依赖项升级到 4.1.0。默认 FHIR 版本已改为 R4。因此，如果需要 DSTU3，则必须明确设置它。

3.7.6. Kafka

camel-kafka 组件删除了选项 bridgeEndpoint 和 circularTopicDetection，因为组件不再需要，因为组件在 Camel 2.x 上可以正常工作。换句话说，camel-kafka 将从 endpoint uri 发送消息到主题。要覆盖它，请使用带有新主题的 KafkaConstants.OVERRIDE_TOPIC 标头。请参阅 camel-kafka 组件文档以了解更多信息。

3.7.7. telegram

camel-telegram 组件已将授权令牌从 uri-path 移到查询参数，如 migrate

```
telegram:bots/myTokenHere
```

to

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

3.7.8. JMX

如果您只使用 camel-core 作为依赖项运行 Camel 独立，并且希望开箱即用启用 JMX，则需要将 camel-management 添加为依赖项。

对于使用 ManagedCamelContext，您需要从 CamelContext 获取此扩展，如下所示：

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

3.7.9. XSLT

XSLT 组件已从 camel-core 移到 camel-xslt 和 camel-xslt-saxon。组件被分开，因此 camel-xslt 用于使用 JDK XSTL 引擎(Xalan)，camel-xslt-saxon 是使用 Saxon 时的。这意味着，您应该在 Camel 端点 URI 中使用 xslt 和 xslt-saxon 作为组件名称。如果您使用 XSLT 聚合策略，则使用 org.apache.camel.component.xslt.saxon.XsltSaxonAggregationStrategy 进行 Saxon 支持。并使用 org.apache.camel.component.xslt.saxon.XsltSaxonBuilder 进行 Saxon 支持（如果使用 xslt 构建器）。另请注意，只有 camel-xslt-saxon 中也支持 allowStax，因为 JDK XSLT 不支持它。

3.7.10. XML DSL 迁移

XML DSL 稍微改变。

自定义负载均衡器 EIP 已从 < custom> 改为 < customLoadBalancer>

在 <secureXML> tag 中，XMLSecurity 数据格式将属性 keyOrTrustStoreParametersId 重新命名为 keyOrTrustStoreParametersRef。

& lt;zipFile& gt; 数据格式已重命名为 < zipfile>。

3.8. 迁移 CAMEL MAVEN 插件

camel-maven-plugin 已分成两个 maven 插件：

camel-maven-plugin

camel-maven-plugin 具有 run 目标，旨在单独运行 Camel 应用程序。如需更多信息，请参阅 https://camel.apache.org/manual/camel-maven-plugin.html。

camel-report-maven-plugin

camel-report-maven-plugin 具有 validate 和 route-coverage 目标，用于生成 Camel 项目报告，如验证 Camel 端点 URI 和路由覆盖报告等。如需更多信息，请参阅 https://camel.apache.org/manual/camel-report-maven-plugin.html。