



Red Hat build of Apache Camel 4.8

Release Notes for Red Hat build of Apache Camel for Spring Boot

What's new in Red Hat build of Apache Camel

Red Hat build of Apache Camel 4.8 Release Notes for Red Hat build of Apache Camel for Spring Boot

What's new in Red Hat build of Apache Camel

Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Describes the Red Hat build of Apache Camel product and provides the latest details on what's new in this release.

Table of Contents

CHAPTER 1. RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT 4.8 RELEASE NOTES	3
1.1. FEATURES IN RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT	3
1.2. SUPPORTED PLATFORMS, CONFIGURATIONS, DATABASES, AND EXTENSIONS FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT	3
1.3. THE JAVAX TO JAKARTA PACKAGE NAMESPACE CHANGE	3
1.3.1. Migration tools	3
1.4. IMPORTANT NOTES FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT	4
1.4.1. Support for IBM Power and IBM Z	4
1.4.2. Changes to the snowdrop groupId	4
1.4.3. Using the automatic Camel context reloading on Secret Refresh feature of AWS Secret Manager component starter	4
1.5. FIXED ISSUES FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT	4
1.5.1. Red Hat build of Apache Camel for Spring Boot version 4.8.3 fixed issues	4
1.5.2. Red Hat build of Apache Camel for Spring Boot version 4.8.0 fixed issues	5
1.6. KNOWN ISSUES FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT	7
1.6.1. Red Hat build of Apache Camel for Spring Boot version 4.8.3 known issues	7
1.6.2. Red Hat build of Apache Camel for Spring Boot version 4.8.0 known issues	8
CHAPTER 2. ADDITIONAL RESOURCES	10

CHAPTER 1. RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT 4.8 RELEASE NOTES

1.1. FEATURES IN RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT

Red Hat build of Apache Camel for Spring Boot introduces Camel support for Spring Boot which provides auto-configuration of Camel, and starters for many Camel components. The opinionated auto-configuration of the Camel context auto-detects Camel routes available in the Spring context and registers key Camel utilities (like producer template, consumer template and the type converter) as beans.

1.2. SUPPORTED PLATFORMS, CONFIGURATIONS, DATABASES, AND EXTENSIONS FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT

- For information about supported platforms, configurations, and databases in Red Hat build of Apache Camel for Spring Boot, see the [Supported Configuration](#) page on the Customer Portal (login required).
- For a list of Red Hat Red Hat build of Apache Camel for Spring Boot extensions, see the [Red Hat build of Apache Camel for Spring Boot Reference](#) (login required).

1.3. THE JAVAX TO JAKARTA PACKAGE NAMESPACE CHANGE

The Java EE move to the Eclipse Foundation and the establishment of Jakarta EE, since Jakarta EE 9, packages used for all EE APIs have changed to **jakarta.***

Code snippets in documentation have been updated to use the **jakarta.*** namespace, but you of course need to take care and review your own applications.



NOTE

This change does not affect javax packages that are part of Java SE.

When migrating applications to EE 10, you need to:

- Update any import statements or other source code uses of EE API classes from the **javax** package to **jakarta**.
- Change any EE-specified system properties or other configuration properties whose names begin with **javax.** to begin with **jakarta..**
- Use the **META-INF/services/jakarta.[rest_of_name]** name format to identify implementation classes in your applications that use the implement EE interfaces or abstract classes bootstrapped with the **java.util.ServiceLoader** mechanism.

1.3.1. Migration tools

- Source code migration: [How to use Red Hat Migration Toolkit for Auto-Migration of an Application to the Jakarta EE 10 Namespace](#)

- Bytecode transforms: For cases where source code migration is not an option, the open source [Eclipse Transformer](#)

Additional resources

- Background: [Update on Jakarta EE Rights to Java Trademarks](#)
- Red Hat Customer Portal: [Red Hat JBoss EAP Application Migration from Jakarta EE 8 to EE 10](#)
- Jakarta EE: [Javax to Jakarta Namespace Ecosystem Progress](#)

1.4. IMPORTANT NOTES FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT

1.4.1. Support for IBM Power and IBM Z

Red Hat build of Camel Spring Boot is now supported on IBM Power and IBM Z.

1.4.2. Changes to the snowdrop groupId

The snowdrop groupId is changed from **me.snowdrop** to **dev.snowdrop**. You must update the **pom.xml** file accordingly.

1.4.3. Using the automatic Camel context reloading on Secret Refresh feature of AWS Secret Manager component starter

To use the [Automatic Camel context reloading on Secret Refresh](#) feature, the secret update has to be done either via UI or via API call with operation **PutSecretValue**. The camel context reload will not be triggered with executing updateSecret via Camel.

1.5. FIXED ISSUES FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT

The following sections list the issues that have been resolved in Red Hat build of Apache Camel for Spring Boot.

- [Section 1.5.1, "Red Hat build of Apache Camel for Spring Boot version 4.8.3 fixed issues"](#)
- [Section 1.5.2, "Red Hat build of Apache Camel for Spring Boot version 4.8.0 fixed issues"](#)

1.5.1. Red Hat build of Apache Camel for Spring Boot version 4.8.3 fixed issues

The following sections list the issues that have been resolved in Red Hat build of Apache Camel for Spring Boot version 4.8.3.

Table 1.1. Red Hat build of Apache Camel for Spring Boot version 4.8.3 resolved issues

Issue	Description
CSB-5877	Camel JBang Export/Run are not working based on camel-version setting

Issue	Description
CSB-6253	Platform-http doesn't remove methods on Camel context reload
CSB-6259	CVE-2024-53990 org.asynchttpclient/async-http-client: AsyncHttpClient (AHC) library's CookieStore replaces explicitly defined `Cookie`s
CSB-6284	camel-platform-http-starter HttpBinding does not support concurrent multipart/form-data requests with the same key id
CSB-6292	CVE-2024-12798 ch.qos.logback/logback-core: arbitrary code execution via JaninoEventEvaluator
CSB-6295	CVE-2024-52046 org.apache.mina/mina-core: Apache MINA: applications using unbounded deserialization may allow RCE
CSB-6298	Camel Opentelemetry, add the RouteID attribute to every span.
CSB-6309	GZIPOutInterceptor : ensure the CXF headers that GZIPOutInterceptor needs to resize is modifiable
CSB-6348	CXF opentelemetry - using same trace id from different http requests

1.5.2. Red Hat build of Apache Camel for Spring Boot version 4.8.0 fixed issues

The following sections list the issues that have been resolved in Red Hat build of Apache Camel for Spring Boot version 4.8.0.

Table 1.2. Red Hat build of Apache Camel for Spring Boot version 4.8.0 resolved issues

Issue	Description
CSB-3066	Add support for camel-opensearch
CSB-3887	Request BeanIO counting character length by bytes
CSB-4068	Implement the kafka consumer's "offsetsForTimes" method on the Kafka consumer
CSB-4452	Improve logging of offliner tool
CSB-4600	Allow to configure Offsets position and Offsets timestamp as property for Camel Kafka Component
CSB-4608	support Hashicorp vault
CSB-4672	Define Agroal version in CSB platform BOM

Issue	Description
CSB-4781	Marshaling the surrogate pair characters by camel-jackson results with Garbled characters
CSB-4868	Lack of jolokia in the base image causes hawtio-online connection error "Jolokia Connect Error - Bad Gateway (502)"
CSB-4978	Refactor and remove the cxf-rt-transport-jetty/cxf-rt-transport-netty-server/cxf-rt-transport-undertow
CSB-5066	Circular dependency error when restConfiguration is defined in a Spring beans XML
CSB-5076	OpenTelemetry missing traces, spans and or context
CSB-5338	[CAMEL-20790]kafka batching consumer polls randomly failing with NPE under load
CSB-5382	camel-rest - Code first should use actual values for property placeholders in dumped API spec
CSB-5559	CVE-2024-7254 protobuf: StackOverflow vulnerability in Protocol Buffers
CSB-5572	Create docs for kafka consumer's "offsetsForTimes" method
CSB-5584	Excessive locking in camel jaxb under load
CSB-5587	support component camel-azure-key-vault
CSB-5592	support component camel-google-secret-manager
CSB-5597	support component camel-aws-secrets-manager
CSB-5603	CVE-2021-44549 org.eclipse.angus/angus-mail: Enabling Secure Server Identity Checks for Safer SMTPS Communication
CSB-5663	[CAMEL-21300]camel-platform-http - Consumer should have option to control if writing response failing should cause Exchange to fail
CSB-5668	Camel-rest with undertow: Occasional ConcurrentModificationException
CSB-5673	Address CXF Async Calls with OpenTelemetry
CSB-5748	Camel-CICS - CTG6662E This JavaGateway instance is already open
CSB-5749	[CAMEL-21329] camel-zipfile - Null body is not supported by ZipAggregationStrategy

Issue	Description
CSB-5812	CVE-2024-38816 org.springframework/spring-webmvc: Path Traversal Vulnerability in Spring Applications Using RouterFunctions and FileSystemResource
CSB-5815	CVE-2024-47561 org.apache.avro/avro: Schema parsing may trigger Remote Code Execution (RCE)
CSB-5880	JsonPath cant read message body coming from platform-http by default
CSB-5887	platform-http Large File Streaming
CSB-5890	platform-http Write Response Handler Error
CSB-5955	camel-http does not support socketTimeout option anymore
CSB-5958	For camel-crypto component, it is not possible to use "inline" with "AES/GCM/NoPadding"
CSB-6126	CVE-2024-31141 org.apache.kafka/kafka-clients: privilege escalation to filesystem read-access via automatic ConfigProvider
CSB-6170	camel-opentelemetry: the camel spans are not present anymore on the traces
CSB-6269	platform-http-starter handle attachments
CSB-6270	platfrom-http-starter enforce produces and consumes configuration

1.6. KNOWN ISSUES FOR RED HAT BUILD OF APACHE CAMEL FOR SPRING BOOT

The following sections list known issues for Red Hat build of Apache Camel for Spring Boot.

1.6.1. Red Hat build of Apache Camel for Spring Boot version 4.8.3 known issues

[CSB-6437](#) CXF opentelemetry - using same trace id from different http requests on RHEL9

This issue only occurs with CXF and OpenTelemetry with the custom tracing configuration on RHEL 9 platform. In case of CXF and opentelemetry with the custom opentelemetry tracer defined, when you call multiple requests, the trace ID seems to be reused in the Camel routes. The result is that new spans are added in the existing traces for each http request, instead of creating one new trace ID for each http request. The workaround is to explicitly define the **ContextPropagators** to **W3CTraceContextPropagator** as shown in the example below:

```
@Bean
ContextPropagators contextPropagators() {
    // return
```

```
ContextPropagators.create(TextMapPropagator.composite(W3CBaggagePropagator.getInstance()));
    return ContextPropagators.create(W3CTraceContextPropagator.getInstance());
}
```

1.6.2. Red Hat build of Apache Camel for Spring Boot version 4.8.0 known issues

CSB-4318 Fail to deploy on OCP using Openshift Maven Plugin if `spring.boot.actuator.autoconfigure` is not in the dependencies

JKube maven plugin uses the following condition to check if the application exposes health endpoint (using **SpringBootHealthCheckEnricher**). Both classes are in the classpath:

- **org.springframework.boot.actuate.health.HealthIndicator**
- **org.springframework.web.context.support.GenericWebApplicationContext**

However, the `/actuator/health` will be not exposed without the configuration of the actuator. This creates discordance between the readiness/liveness probes configured by JKube (they both uses the above endpoint) and what the application is exposing.

This misconfiguration causes a failing deployment config on OpenShift Container Platform since the generated pod will never be in Ready status since the probe's call for an endpoint is not configured. So in order to make the application work on OpenShift Container Platform, which is deployed using JKube (openshift-maven-plugin), it is necessary to have both web and actuator autoconfiguration in the dependencies.

Following example shows how to configure web and actuator autoconfiguration.

Example

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

Update the archetype as shown below. The applications built from the following archetype will be deployed correctly using JKube.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-undertow</artifactId>
```

```
</dependency>  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>
```

This issue affects the custom applications with missing one of the above dependencies.

CHAPTER 2. ADDITIONAL RESOURCES

- [Supported Configurations](#)
- [Getting Started with 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 for Spring Boot Reference](#)