Red Hat build of Apache Camel 4.4

Release Notes for Red Hat build of Apache Camel for Quarkus

What's new in Red Hat build of Apache Camel

Last Updated: 2024-07-04

Red Hat build of Apache Camel 4.4 Release Notes for Red Hat build of Apache Camel for Quarkus

What's new in Red Hat build of Apache Camel

Legal Notice

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. RELEASE NOTES FOR RED HAT BUILD OF APACHE CAMEL FOR QUARKUS 3.8 / 3.8.4.SP2 . 3

1.1. RED HAT BUILD OF APACHE CAMEL FOR QUARKUS FEATURES 3

1.2. SUPPORTED PLATFORMS, CONFIGURATIONS, DATABASES, AND EXTENSIONS 3

1.3. BOM FILES FOR RED HAT BUILD OF APACHE CAMEL FOR QUARKUS 3

1.4. TECHNOLOGY PREVIEW EXTENSIONS 3

1.5. KNOWN ISSUES 4

1.5.1. Issues with Quarkus on AArch64 systems 4

1.5.2. Websocket + Knative does not work with HTTP2 4

1.5.3. Other known issues 4

1.6. KNOWN QUARKUS CXF ISSUES 4

1.6.1. Code generation fails with in-app with SmallRye Config SPI 4

1.6.2. Name clash between Service methods with the same name in one Java package 5

1.6.3. Beans not injected into a service implementation with @CXFEndpoint("/my-path") 5

1.7. IMPORTANT NOTES 5

1.7.1. The javax to jakarta Package Namespace Change 5

1.7.1.1. Migration tools 6

1.7.2. Support for IBM Power and IBM Z 6

1.7.3. Minimum Java version - JDK 17 6

1.7.4. Support for OpenJDK 6

1.7.5. Support for AdoptiumJDK 6

1.7.6. Upgrades 6

1.7.7. Camel upgraded from version 4.0 to version 4.4 6

1.7.8. Camel Quarkus upgraded from version 3.2 to version 3.8 7

1.8. RESOLVED ISSUES 7

1.8.1. Previous releases 7

1.9. DEPRECATED FEATURES IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8 8

1.9.1. Openapi-java support for Openapi v2 8

1.10. EXTENSIONS REMOVED IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8 8

1.11. EXTENSIONS ADDED IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8 8

1.12. EXTENSIONS WITH CHANGED SUPPORT IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8 9

1.13. DATA FORMATS ADDED IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8 9

1.14. ADDITIONAL RESOURCES 9

CHAPTER 1. RELEASE NOTES FOR RED HAT BUILD OF APACHE CAMEL FOR QUARKUS 3.8 / 3.8.4.SP2

1.1. RED HAT BUILD OF APACHE CAMEL FOR QUARKUS FEATURES

Fast startup and low RSS memory

Using the optimized build-time and ahead-of-time (AOT) compilation features of Quarkus, your Camel application can be pre-configured at build time resulting in fast startup times.

Application generator

Use the Quarkus application generator to bootstrap your application and discover its extension ecosystem.

Highly configurable

All the important aspects of a Red Hat build of Apache Camel for Quarkus application can be set up programmatically with CDI (Contexts and Dependency Injection) or by using configuration properties. By default, a CamelContext is configured and automatically started for you.
Check out the Configuring your Quarkus applications by using a properties file guide for more information on the different ways to bootstrap and configure an application.

Integrates with existing Quarkus extensions

Red Hat build of Apache Camel for Quarkus provides extensions for libraries and frameworks that are used by some Camel components which inherit native support and configuration options.

1.2. SUPPORTED PLATFORMS, CONFIGURATIONS, DATABASES, AND EXTENSIONS

- For information about supported platforms, configurations, and databases in Red Hat build of Apache Camel for Quarkus version 3.8, see the Supported Configuration page on the Customer Portal (login required).

- For a list of Red Hat Red Hat build of Apache Camel for Quarkus extensions and the Red Hat support level for each extension, see the Extensions Overview chapter of the *Red Hat build of Apache Camel for Quarkus Reference* (login required).

1.3. BOM FILES FOR RED HAT BUILD OF APACHE CAMEL FOR QUARKUS

- To configure your Red Hat Red Hat build of Apache Camel for Quarkus version 3.8 projects to use the supported extensions, use the latest Bill Of Materials (BOM) version **3.8.4.SP2-redhat-00001** or newer, from the Redhat Maven Repository.

For more information about BOM dependency management, see Developing Applications with Red Hat build of Apache Camel for Quarkus

1.4. TECHNOLOGY PREVIEW EXTENSIONS

Items designated as *Technology Preview* in the Extensions Overview chapter of the *Red Hat build of Apache Camel for Quarkus Reference* have limited supportability, as defined by the Technology Preview Features Support Scope.

1.5. KNOWN ISSUES

1.5.1. Issues with Quarkus on AArch64 systems

There currently are problems and limitations with Quarkus 3.8 on AArch64 systems. For more information, see the release notes for Red Hat build of Quarkus 3.8:

- Missing native library for the Kafka Streams extension on AArch64
- AArch64 support limitations in JVM mode testing on OpenShift

1.5.2. Websocket + Knative does not work with HTTP2

We support both **camel-quarkus-grpc** and **camel-vertx-websocket** with Knative.

gRPC needs HTTP2 (you can find instructions on how to enable it here: HTTP2 on Knative).

Unfortunately, Websockets with Knative does not work with HTTP2 (see Ingress Operator in OpenShift Container Platform).

Consequently, if you have an application that is intended to accept WebSocket connections, it must not allow negotiating the HTTP/2 protocol or else clients will fail to upgrade to the WebSocket protocol.

1.5.3. Other known issues

Moving from smallrye-metrics to camel-quarkus-micrometer requires manual registration of beans

If you are migrating to **micrometer** from **smallrye-metrics**, you may need to manually define some beans as scoped.
In **smallrye-metrics**, classes that are registered for metrics (for example with **@COUNTED**, **@METRIC**), but not registered as scoped beans, are registered automatically. This does not happen in **micrometer**.

In **micrometer** you need to manually register beans accessed via CDI, by for example adding a **@Dependent** annotation.

Camel-quarkus-snmp not supported in Native

In Red Hat build of Apache Camel for Quarkus we support the **camel-quarkus-snmp** component in JVM mode only.

1.6. KNOWN QUARKUS CXF ISSUES

NOTE

CXF is fully supported, but the following issues remain with this release of Red Hat build of Apache Camel for Quarkus.

1.6.1. Code generation fails with in-app with SmallRye Config SPI

SmallRye Config supports implementing custom secret key expressions of the form **${handler::value}**, where the handler is the name of a **io.smallrye.config.SecretKeysHandler** to decode or decrypt the value separated by a double colon **::**.

If the **SecretKeysHandler** is implemented in the user application, then, if the wsdl2Java code generation is enabled, it fails with an error message like the following:

```
Failed to execute goal io.quarkus.platform:quarkus-maven-plugin:3.8.0:generate-code (default) on
project fallgruppenbestimmung: Quarkus code generation phase has failed:
InvocationTargetException: io.smallrye.config.SecretKeysHandler: Provider
org.acme.MySecretKeyHandler not found
```

1.6.2. Name clash between Service methods with the same name in one Java package

If there are two SEIs in one Java package, both having a **@WebMethod** with the same name but different signature, then the default name for the generated request, response and possibly other classes is the same for both methods of both classes.

As of Quarkus CXF 3.8.3, no exception is thrown when this happens during the class generation at build time. At runtime, only one set of those classes is present and therefore calls to one of the clients fail inevitably.

1.6.3. Beans not injected into a service implementation with @CXFEndpoint("/my-path")

The **@CXFEndpoint** annotation was mistakenly introduced in Red Hat build of Apache Camel for Quarkus 3.8.4.SP2 and Quarkus CXF 3.8.4. It is reverted in the of Red Hat build of Apache Camel for Quarkus 3.8 and Quarkus CXF 3.8.5.

The annotation allows you to specify CXF service endpoint paths through the new annotation **@CXFEndpoint("/myPath")**.

This does not work well for service implementation classes having both **@WebService** and **@CXFEndpoint** annotations. In those cases, if the service has some **@Inject** fields, those fields are left blank and the service call throws a **NullPointerException**.

Service implementations that do not have the **@CXFEndpoint** annotations are unaffected.

We recommend that you continue to specify service endpoint paths in **application.properties** as before:

Example

```
quarkus.cxf.endpoint."/myPath".implementor = org.acme.MyServiceImpl
```

1.7. IMPORTANT NOTES

1.7.1. 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.

1.7.1.1. 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

1.7.2. Support for IBM Power and IBM Z

Red Hat build of Apache Camel for Quarkus is now supported on IBM Power and IBM Z.

1.7.3. Minimum Java version - JDK 17

Red Hat build of Apache Camel for Quarkus version 3.8 requires JDK 17 or newer.

1.7.4. Support for OpenJDK

Red Hat build of Apache Camel for Quarkus version 3.8 includes support for OpenJDK 21.

1.7.5. Support for AdoptiumJDK

Red Hat build of Apache Camel for Quarkus version 3.8 includes support for AdoptiumJDK 17 and AdoptiumJDK 21.

1.7.6. Upgrades

1.7.7. Camel upgraded from version 4.0 to version 4.4

Red Hat build of Apache Camel for Quarkus version 3.8 has been upgraded from Camel version 4.0 to Camel version 4.4. For additional information about each intervening Camel patch release, refer to the following:

- Apache Camel 4.0.1 Release Notes
- Apache Camel 4.0.2 Release Notes
- Apache Camel 4.0.3 Release Notes
- Apache Camel 4.0.4 Release Notes
- Apache Camel 4.0.5 Release Notes
- Apache Camel 4.1.0 Release Notes
- Apache Camel 4.2.0 Release Notes
- Apache Camel 4.3.0 Release Notes
- Apache Camel 4.4.0 Release Notes

1.7.8. Camel Quarkus upgraded from version 3.2 to version 3.8

Red Hat build of Apache Camel for Quarkus version 3.8 has been upgraded from Camel Quarkus version 3.2 to Camel Quarkus version 3.8. For additional information about each intervening Camel Quarkus patch release, refer to the following:

- Apache Camel Quarkus 3.2.1 Release Notes
- Apache Camel Quarkus 3.2.2 Release Notes
- Apache Camel Quarkus 3.2.3 Release Notes
- Apache Camel Quarkus 3.4.0 Release Notes
- Apache Camel Quarkus 3.5.0 Release Notes
- Apache Camel Quarkus 3.6.0 Release Notes
- Apache Camel Quarkus 3.7.0 Release Notes
- Apache Camel Quarkus 3.8.0 Release Notes

1.8. RESOLVED ISSUES

The following lists shows known issues that were affecting Red Hat build of Apache Camel for Quarkus, which have been fixed in Red Hat build of Apache Camel for Quarkus version 3.8.

Resolved issues

CEQ-8857

Camel-http producer sets "Content-Encoding=UTF-8"

1.8.1. Previous releases

For details of issues resolved between Camel Quarkus 3.2 and Camel Quarkus 3.8, see the Release Notes for each patch release.

1.9. DEPRECATED FEATURES IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8

The following capabilities are not available in the next major release of Red Hat build of Apache Camel for Quarkus, and are deprecated in this release.

1.9.1. Openapi-java support for Openapi v2

Deprecated features

OpenApi V2 is deprecated in 3.8, due to dropped support in Openapi-java with Camel 4.5.x.

1.10. EXTENSIONS REMOVED IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8

No extensions are removed in the Red Hat build of Apache Camel for Quarkus version 3.8 release.

1.11. EXTENSIONS ADDED IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8

The following table lists the extensions added in the Red Hat build of Apache Camel for Quarkus version 3.8 release .

Table 1.1. Added extensions

Extension	Artifact	Description
Jasypt	**camel-quarkus-jasypt**	Security using Jasypt
JSON Path	**camel-quarkus-jsonpath**	Evaluate a JSONPath expression against a JSON message body
JT400	**camel-quarkus-jt400**	Exchanges messages with an IBM i system using data queues, message queues, or program call. IBM i is the replacement for AS/400 and iSeries servers.
Kudu	**camel-quarkus-kudu**	Interact with Apache Kudu, a free and open source column-oriented data store of the Apache Hadoop ecosystem.
LRA	**camel-quarkus-lra**	Camel saga binding for Long-Running-Action framework.
Saga	**camel-quarkus-saga**	Execute custom actions within a route using the Saga EIP.

Extension	Artifact	Description
Splunk HEC	**camel-quarkus-splunk-hec**	The splunk component allows to publish events in Splunk using the HTTP Event Collector.
XJ	**camel-quarkus-xj**	Transform JSON and XML message using a XSLT.

1.12. EXTENSIONS WITH CHANGED SUPPORT IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8

No extensions have changed support levels in the Red Hat build of Apache Camel for Quarkus version 3.8 release.

NOTE

For information about support levels, see Red Hat build of Apache Camel for Quarkus Extensions

1.13. DATA FORMATS ADDED IN RED HAT BUILD OF APACHE CAMEL FOR QUARKUS VERSION 3.8

No data formats have been added in the Red Hat build of Apache Camel for Quarkus version 3.8 release..

1.14. ADDITIONAL RESOURCES

- Supported Configurations
- Red Hat build of Apache Camel for Quarkus Extensions
- Getting Started with Red Hat build of Apache Camel for Quarkus
- Developing Applications with Red Hat build of Apache Camel for Quarkus