Red Hat support for Spring Boot 2.3

Release Notes for Spring Boot 2.3

For use with Spring Boot 2.3.10

Red Hat support for Spring Boot 2.3 Release Notes for Spring Boot 2.3

For use with Spring Boot 2.3.10

Legal Notice

Abstract

This Release Note contains important information related to Spring Boot 2.3.10

Table of Contents

PREFACE

Date of release: 2021-09-08

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments.

This section explains how to submit feedback.

Prerequisites

- You are logged in to the Red Hat Customer Portal.

- In the Red Hat Customer Portal, view the document in **Multi-page HTML** format.

Procedure

To provide your feedback, perform the following steps:

1. Click the **Feedback** button in the top-right corner of the document to see existing feedback.

   > **NOTE**
   >
   > The feedback feature is enabled only in the **Multi-page HTML** format.

2. Highlight the section of the document where you want to provide feedback.

3. Click the **Add Feedback** pop-up that appears near the highlighted text.
   A text box appears in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click **Submit**.
   A documentation issue is created.

5. To view the issue, click the issue tracker link in the feedback view.

CHAPTER 1. REQUIRED INFRASTRUCTURE COMPONENT VERSIONS

Red Hat does not provide support for components listed below, with the exception of components explicitly designated as supported.

Component name	Version
Maven	3.6.0
Fabric8 Maven Plugin	4.4.1
JDK[a][b]	OpenJDK 8, OpenJDK 11[c]
Red Hat Enterprise Linux 7[d]	7.7
Red Hat Enterprise Linux 8[e]	8.1
OpenShift Container Platform (OCP)[f]	3.11, 4.8
Minishift	1.34.2 or later
CDK[g]	3.11.0
git	2.0 or later
oc command line tool	3.11 or later[h]

[a] A full JDK installation is required, as JRE does not provide tools for compiling Java applications from source.

[b] Red Hat OpenJDK is supported by Red Hat

[c] OpenJDK 9 is not supported by Red Hat.

[d] For deploying applications based on CNR on stand-alone RHEL in a production environment.

[e] For deploying applications based on CNR on stand-alone RHEL in a production environment.

[f] OCP is supported by Red Hat

[g] CDK is supported by Red Hat

[h] The version of the **OC** CLI tool should correspond to the version of OCP that you are using.

CHAPTER 2. SUPPORTED SPRING BOOT RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS

The following resource defines the supported configurations and integrations of Red Hat products with Spring Boot:

- For a list of technologies that are supported for integration with Spring Boot in production environments see the Supported Spring Boot configurations and integrations.

- For a list of Maven artifacts that Red Hat provides support for in Spring Boot 2.3.10 see the corresponding section of the component details overview.

CHAPTER 3. FEATURES

3.1. NEW AND CHANGED FEATURES

3.1.1. Deploying example applications on OpenShift provisioned on IBM Power Systems infrastructure

To deploy the example applications on OpenShift environments provisioned on IBM Power Systems infrastructure, specify the relevant IBM Power Systems image name in the **pom.xml** file and commands.

Some of the example applications also require other products, such as Red Hat Data Grid to demonstrate the workflows. In this case, you must also change the image names of these products to their relevant IBM Power Systems image names in the YAML file of the example applications.

3.1.2. New Spring Boot OAuth2 Client and Resource Server Starters

A tested and verified build of OAuth2 Spring Security Starters replaces the Spring Boot Keycloak Starter included with previous Spring Boot releases. The set of new Starters introduced in this release includes the Spring Security OAuth2 Client Starter and the Spring Security OAuth2 Resource Server Starter. Both of the new Starters are supported by Red Hat.

3.1.3. Support for OpenJDK 8 and OpenJDK 11 RHEL 8 Universal Base Images (UBI8)

Spring Boot 2.3 introduces support for building and deploying Spring Boot applications to OpenShift with OCI-compliant Universal Base Images for Red Hat OpenJDK 8 and Red Hat OpenJDK 11 on RHEL 8. The RHEL 8 OpenJDK Universal Base Images replace the RHEL 8 OpenJDK builder images supported by Red Hat for use with earlier releases of Red Hat Build of Spring Boot. The previously supported RHEL 8 OpenJDK base images are no longer supported for use with Red Hat Build of Spring Boot.

3.1.4. Dekorate version upgraded to 2.4.1

In Spring Boot 2.3, the version of Dekorate included with the Spring Boot BOM is upgraded to 2.4.1.

New features in Dekorate 2.4.1

- Configurable logging threshold
  You can control how verbose the Dekorate output is by setting the log-level threshold, using the **io.dekorate.log.level** system property-drawer. The following values are supported:

  - **OFF**

  - **ERROR**

  - **WARN**

  - **INFO** (default)

  - **DEBUG**

- Configurable **vcs** properties
  In Dekorate 1.x, the **vcs** labels that were added to the generated manifests contained the property **remote** hard-coded as **origin** and URL converted to the **http** protocol.

In Dekorate 2.4.1, the property **remote** is configurable and the URL is not converted to HTTP unless explicitly specified. You can configure these programmatically using **io.dekorate.option.annotation.VcsOptions** as follows:

```
public @interface VcsOptions {  /**
  * The remote to use.
  *
  * @return the name of the remote, defaults to 'origin'.
  */
  String remote() default "origin";  /**
  * Flag that specifies that https is preferred.
  * When use vcs url that use 'git+ssh' will be converted to https.
  *
  * @return true, if https is prefered.
  */
  boolean httpsPreferred() default false;
}
```

You can also configure the property as follows:

```
dekorate.option.vcs.remote=<your remote here>
dekorate.option.vcs.https-preferred=true
```

- Changes to port mapping
  The earlier versions of Dekorate did not allow port mapping on Service resources, except **Ingress** and **Route** resources. Because of this, the internal consumers of the Service had to know the exact **containerPort** an application used. Dekorate 2.4.1 provides port mapping to allow loose-coupling for HTTP and HTTPS.

  http services are now mapped to port **80** and **https** services to port **443**. Port mapping is done using the following criteria:

  - Ports named **http**, **web** and **http1** are mapped to port **80**.

  - Ports named **https** or **h2c** are mapped to port **443**.

  - Ports numbered **8080** are mapped to port **80**.

  - Ports numbered **8443** are mapped to port **443**.

Changes to annotation naming in Dekorate 2.4.1

The following table lists the changes in annotation names:

Dekorate 1.x	Dekorate 2.4.1
DockerBuild	**EnableDockerBuild**
S2iBuild	**EnableS2iBuild**
JibBuild	**EnableJibBuild**

Modules removed in Dekorate 2.4.1

The following modules have been removed from Dekorate in version 2.4.1:

- service-catalog

- halkyon

- application-crd

- CRD generator
  This functionality has been moved to the **fabric8 kubernetes-client**.

- dependencies uberjar
  The earlier versions of Dekorate provided shadowed uberjar with all dependencies. From Dekorate 2.4.1 onward, the shadowed uberjar containing all dependencies is no longer available.

  If you use the libraries **io.dekorate.deps.xxx**, you must update these to the original packages.

3.1.5. Spring Boot metering labels for OpenShift

You can add metering labels to your Spring Boot pods and check Red Hat subscription details with the OpenShift Metering Operator.

> **NOTE**
>
> Do not add metering labels to any pods that an operator deploys and manages.

Spring Boot should use the following metering labels:

- **com.redhat.component-name: "Spring_Boot"**

- **com.redhat.component-type: application**

- **com.redhat.component-version: 2.3.10**

- **com.redhat.product-name: "Red_Hat_Runtimes"**

- **com.redhat.product-version: 2021-Q3**

See Metering documentation for more information.

For more information on labels, see Understanding how to update labels on nodes .

3.1.6. Support for Spring Boot Runtime on IBM Power Systems

The Red Hat support for Spring Boot for ppc64le platform is supported only in OpenShift environments provisioned on IBM Power Systems infrastructure. Running an Spring Boot application on a stand-alone installation of RHEL on IBM Power Systems is not supported.

OpenJ9 Java images for IBM Power Systems and new images for products supported on IBM Power Systems are available in the Red Hat Container Catalog .

3.2. DEPRECATED FEATURES

No features or functionalities are marked as deprecated in this release.

3.3. TECHNOLOGY PREVIEW

3.3.1. Dekorate build hooks for deploying Spring Boot applications to OpenShift Container Platform

You can use Dekorate to configure a Source-to-image build of your application that starts automatically after you compile your application with Maven. This functionality is provided as Technology Preview in Dekorate version 1.0.0 and later. Red Hat does not provide support for using this functionality in a production environment.

CHAPTER 4. RELEASE COMPONENTS

For a complete list of release components included in this release, and for information about the current support status of these components, see the Spring Boot 2.3.10 component details overview.

CHAPTER 5. FIXED ISSUES

This Spring Boot release incorporates all bugfixes from the upstream release. Issues resolved in the community release are listed in the Spring Boot 2.3.10 Release Notes .

CHAPTER 6. KNOWN ISSUES

- Red Hat AMQ Streams images are not available for IBM Z and IBM Power Systems
  The Red Hat AMQ Streams Operator and Kafka images are not available for IBM Z and IBM Power Systems. Since the images are not available, the starter **vertx-spring-boot-starter-kafka** is not certified to work with AMQ Streams on IBM Z and IBM Power Systems.

- ENTSBT-850: Spring Boot Validation Starter Exception: Package **javax.validation.constraints** does not exist.

- SB-379: Missing APR/native library in the **openshift-openjdk** image.

- SB-1165: Database application fails to run because **org.apache.tomcat.jdbc.pool.DataSource** can not be found.

- ENTSBT-202: Mutual TLS authentication in Spring Boot Webflux AMQP does not work.

- ENTSBT-366: Infinispan Hotrod Client Starter: **org.infinispan.client.hotrod.exceptions.HotRodClientException:: ISPN004034: Unable to unmarshall bytes** when the **infinispan.remote.java-serial-whitelist=<your_class_name>** property is not set in **application.properties**.

- ENTSBT-367: Remote communication between Red Hat Spring Boot 2.3.10 with Infinispan/Red Hat Data Grid 7.3 does not work without setting the **infinispan.remote.protocol-version=2.6** property.

- ENTSBT-912: Examples applications not supported on OpenShift Container Platform 4.x.
  The Spring Boot runtime provides example applications, which are accessed on Developer Launcher. These example applications are not supported on OpenShift Container Platform 4.x.

- ENTSBT-1131: There are two **io.vertx:vertx-dependencies** BOM files in the release:

  - **3.9.5.redhat-00002.pom**

  - **3.9.8.redhat-00004.pom**.

  Use **3.9.8.redhat-00004.pom** with Eclipse Vert.x Spring Boot Starter because the dependencies for the starter are included only in the **3.9.8.redhat-00004.pom** file.

CHAPTER 7. ADVISORIES RELATED TO THIS RELEASE

The following advisories have been issued to document enhancements, bugfixes, and CVE fixes included in this release.

- RHSA-2021:3425-03