



# AMQ Spring Boot Starter 3.4

## 使用 AMQ Spring Boot Starter

为 JMS (jakarta)开发 Spring Boot 应用程序



## AMQ Spring Boot Starter 3.4 使用 AMQ Spring Boot Starter

---

为 JMS (jakarta)开发 Spring Boot 应用程序

## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

本指南介绍了如何安装和配置库，运行实践示例，并将您的客户端与其他组件一起使用。

---

## Table of Contents

<b>使开源包含更多</b> .....	<b>3</b>
<b>第 1 章 概述</b> .....	<b>4</b>
1.1. 主要特性	4
1.2. 支持的标准和协议	4
1.3. 支持的配置	4
1.4. 文档惯例	4
<b>第 2 章 安装</b> .....	<b>5</b>
2.1. 先决条件	5
2.2. 使用 RED HAT MAVEN 存储库	5
2.3. 安装本地 MAVEN 存储库	5
<b>第 3 章 开始使用</b> .....	<b>7</b>
3.1. 先决条件	7
3.2. 运行 HELLO WORLD	7
3.3. 其他示例	9
<b>第 4 章 配置</b> .....	<b>10</b>
4.1. 连接选项	10
4.2. 池选项	10
<b>附录 A. 使用您的订阅</b> .....	<b>12</b>
A.1. 访问您的帐户	12
A.2. 激活订阅	12
A.3. 下载发行文件	12
A.4. 为系统注册软件包	12
<b>附录 B. 使用 RED HAT MAVEN 存储库</b> .....	<b>14</b>
B.1. 使用在线软件仓库	14
B.2. 使用本地存储库	15
<b>附录 C. 在示例中使用 AMQ BROKER</b> .....	<b>17</b>
C.1. 安装代理	17
C.2. 启动代理	17
C.3. 创建队列	17
C.4. 停止代理	17



## 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。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

# 第 1 章 概述

AMQ Spring Boot Starter 是一个适配器，用于创建使用 AMQ 消息传递的基于 Spring 的应用程序。它提供了一个 Spring Boot starter 模块，供您构建独立 Spring 应用程序。初学者使用红帽构建的 Apache Qpid JMS 客户端来使用 AMQP 1.0 协议进行通信。

此发行版本支持 **jakarta.jms**，且需要 Java 版本 17 或更高版本。有关 **javax.jms** 支持，请查看 AMQ Spring Boot Starter 的 2.x 版本。

AMQ Spring Boot Starter 基于 [AMQP 1.0 JMS Spring Boot](#) 项目。

## 1.1. 主要特性

- 使用内置消息传递快速构建独立 Spring 应用程序
- 自动配置 JMS 资源
- 可配置 JMS 连接和会话的池

## 1.2. 支持的标准和协议

- Spring Boot API 的版本 3.4
- [Java 消息服务](#) API 的版本 2.0
- [高级消息队列协议 \(AMQP\)](#) 的版本 1.0

## 1.3. 支持的配置

有关 [AMQ Spring Boot Starter 支持的配置](#)，请参阅红帽客户门户网站中的 Red Hat AMQ 支持的配置。

## 1.4. 文档惯例

### sudo 命令

在本文档中，**sudo** 用于任何需要 root 特权的命令。使用 **sudo** 时请小心谨慎，因为任何更改都可能影响整个系统。有关 **sudo** 的详情，[请参考使用 sudo 命令](#)。

### 文件路径

在本文档中，所有文件路径都对 Linux、UNIX 和类似操作系统有效（例如 `/home/andrea`）。在 Microsoft Windows 上，您必须使用等效的 Windows 路径（例如 `C:\Users\andrea`）。

### 变量文本

本文档包含代码块，其变量必须替换为特定于您的环境的值。变量文本用箭头括起，样式化为字典 monospace。例如，使用以下命令将 `<project-dir>` 替换为您的环境的值：

```
$ cd <project-dir>
```

## 第 2 章 安装

本章介绍了在环境中安装 AMQ Spring Boot Starter 的步骤。

### 2.1. 先决条件

- 您必须有 [订阅](#) 才能访问 AMQ 发行文件和存储库。
- 要使用 AMQ Spring Boot Starter 构建程序，您必须安装 [Apache Maven](#)。
- 要使用 AMQ Spring Boot Starter，您必须安装 Java。

### 2.2. 使用 RED HAT MAVEN 存储库

配置 Maven 环境，以从 Red Hat Maven 存储库下载客户端库。

#### 流程

1. 将红帽存储库添加到 Maven 设置或 POM 文件中。有关配置文件示例，请参阅 [第 B.1 节“使用在线软件仓库”](#)。

```
<repository>
  <id>red-hat-ga</id>
  <url>https://maven.repository.redhat.com/ga</url>
</repository>
```

2. 将库依赖项添加到 POM 文件中。

```
<dependency>
  <groupId>org.amqphub.spring</groupId>
  <artifactId>amqp-10-jms-spring-boot-starter</artifactId>
  <version>3.4.0.redhat-00001</version>
</dependency>
```

客户端现在在您的 Maven 项目中可用。

### 2.3. 安装本地 MAVEN 存储库

作为在线存储库的替代选择，AMQ Spring Boot Starter 可以安装到本地文件系统中，作为基于文件的 Maven 存储库。

#### 流程

1. [使用您的订阅](#) 下载 **AMQ Spring Boot Starter 3.4.0 Maven 存储库.zip** 文件。
2. 将文件内容提取到您选择的目录。  
在 Linux 或 UNIX 上，使用 **unzip** 命令提取文件内容。

```
$ unzip amq-spring-boot-starter-3.4.0-maven-repository-maven-repository.zip
```

在 Windows 上，右键单击 .zip 文件并选择 **Extract All**。

3. 将 Maven 配置为使用提取安装目录中的 **maven-repository** 目录中的存储库。如需更多信息，请参阅 [第 B.2 节“使用本地存储库”](#)。

## 第 3 章 开始使用

本章指导您执行设置环境并运行一个简单的消息传递程序的步骤。

### 3.1. 先决条件

- 若要构建示例，必须将 Maven 配置为使用 [红帽存储库](#) 或 [本地存储库](#)。
- 您必须有一个消息代理侦听 `localhost` 上的连接。它必须启用匿名访问。如需更多信息，请参阅 [启动代理](#)。
- 您必须有一个名为 `example` 的队列。如需更多信息，请参阅 [创建队列](#)。

### 3.2. 运行 HELLO WORLD

Hello World 示例创建一个与代理的连接，发送含有问 `示例` 队列的消息，并将它接收。成功时，它会接收到的消息输出到控制台。

**示例：发送和接收 "Hello World!" - HelloWorld.java**

```
package net.example;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.jms.annotation.EnableJms;
import org.springframework.jms.annotation.JmsListener;
import org.springframework.jms.core.JmsTemplate;

@EnableJms
@SpringBootApplication
public class HelloWorld implements CommandLineRunner {
    @Autowired
    private JmsTemplate jmsTemplate;

    public static void main(String[] args) {
        SpringApplication.run(HelloWorld.class, args);
    }

    @Override
    public void run(String... strings) throws Exception {
        sendMessage("Hello World!");
    }

    public void sendMessage(String text) {
        System.out.println(String.format("Sending '%s'", text));
        this.jmsTemplate.convertAndSend("example", text);
    }

    @JmsListener(destination = "example")
    public void receiveMessage(String text) {
```

```

        System.out.println(String.format("Received '%s'", text));
    }
}

```

## 运行示例

要编译并运行示例程序，请使用以下流程：

## 流程

1. 创建新项目目录。在以下步骤中，这被称为 `<project-dir>`。
2. 将示例列表复制到以下位置：

```
<project-dir>/src/main/java/net/example/HelloWorld.java
```

3. 使用文本编辑器创建新的 `<project-dir>/pom.xml` 文件。在其中添加以下 XML：

```

<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>net.example</groupId>
  <artifactId>example</artifactId>
  <version>1.0.0-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.0</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.amqphub.spring</groupId>
      <artifactId>amqp-10-jms-spring-boot-starter</artifactId>
      <version>3.4.0.redhat-00001</version>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>

```

4. 切换到项目目录，并使用 `mvn` 命令来编译程序。

```
$ mvn clean package
```

5. 使用 `java` 命令运行程序。



## 第 4 章 配置

以下选项可用于 **application-properties** 文件来配置 Spring Boot 应用程序。

### 4.1. 连接选项

这些选项决定了 AMQ Spring Boot Starter 如何建立到远程 AMQP 对等点的连接。初学者使用红帽构建的 Apache Qpid JMS 通过网络进行通信。如需更多信息，[请参阅使用 Apache Qpid JMS](#)。

#### **amqphub.amqp10jms.remoteUrl**

红帽构建的 Apache Qpid JMS 客户端用来建立新连接的连接 URI。

#### 连接 URI 格式

```
amqp[s]://host:port[?option=value[&option2=value...]]
```

如需更多信息，请参阅红帽构建的 Apache Qpid JMS 中的[连接 URI](#)。

#### **amqphub.amqp10jms.username**

用于验证连接的用户名。

#### **amqphub.amqp10jms.password**

用于验证连接的密码。

#### **amqphub.amqp10jms.clientId**

应用到连接的客户端 ID。

#### **amqphub.amqp10jms.receiveLocalOnly**

如果启用，则使用 `timeout` 参数 **接收的** 调用只检查消费者的本地消息缓冲。否则，也会检查远程 peer，以确保没有可用的消息。它默认是禁用的。

#### **amqphub.amqp10jms.receiveNoWaitLocalOnly**

如果启用，则只调用 **receiveNoWait** 检查消费者的本地消息缓冲区。否则，也会检查远程 peer，以确保没有可用的消息。它默认是禁用的。

### 4.2. 池选项

这些选项决定了 AMQ Spring Boot Starter 如何缓存 JMS 连接和会话。

#### **amqphub.amqp10jms.pool.enabled**

控制是否启用了池。它默认是禁用的。

#### **amqphub.amqp10jms.pool.maxConnections**

单个池的最大连接数。默认值为 1。

#### **amqphub.amqp10jms.pool.maxSessionsPerConnection**

每个连接的会话的最大数量。默认值为 500。负值会删除任何限制。如果超过限制，请创建 **Session** () 或抛出异常，具体取决于配置。

#### **amqphub.amqp10jms.pool.blockIfSessionPoolsFull**

如果启用，调用 **createSession** () 块，直到池中有会话可用。它会被默认启用。如果禁用，如果没有会话可用，则调用 **createSession** () 会抛出 **IllegalStateException**。

**amqphub.amqp10jms.pool.blockIfSessionPoolsFullTimeout**

阻塞调用 **createSession ()** 前的时间 (毫秒) 会抛出 **IllegalStateException**。默认值为 -1, 表示调用块(ever)。

**amqphub.amqp10jms.pool.connectionIdleTimeout**

当前没有位于 loan 上的连接可以从池中驱除时间 (毫秒)。默认值为 30 秒。0 代表禁用超时。

**amqphub.amqp10jms.pool.connectionCheckInterval**

定期检查过期的连接之间的时间 (毫秒)。默认值为 0, 表示检查被禁用。

**amqphub.amqp10jms.pool.useAnonymousProducers**

如果启用, 则使用单个匿名 JMS **MessageProducer** 进行所有调用 来创建 **Producer ()**。它会被默认启用。

在个别情况下, 此行为是不正常的。如果禁用, 则每个 **创建Producer ()** 调用都会生成一个新的 **MessageProducer** 实例。

**amqphub.amqp10jms.pool.explicitProducerCacheSize**

如果不使用匿名制作者, 则 JMS 会话 可以配置为缓存具有特定目的地的 **MessageProducer** 对象。随着新制作者创建与缓存的生成者不匹配, 缓存中的最旧的条目会被驱除。

**amqphub.amqp10jms.pool.useProviderJMSContext**

如果启用, 则使用底层 JMS 提供程序的 **JMSContext** 类。它默认是禁用的。

在正常操作中, 池使用自己的通用 **JMSContext** 实施来嵌套来自池的连接, 而不使用提供程序实施。通用实施可能会限制供应商实施。但是, 在启用时, 来自 **JMSContext** API 的连接不由池管理。

## 附录 A. 使用您的订阅

AMQ 通过软件订阅提供。要管理您的订阅，请访问红帽客户门户中的帐户。

### A.1. 访问您的帐户

#### 流程

1. 转至 [access.redhat.com](https://access.redhat.com)。
2. 如果您还没有帐户，请创建一个帐户。
3. 登录到您的帐户。

### A.2. 激活订阅

#### 流程

1. 转至 [access.redhat.com](https://access.redhat.com)。
2. 导航到 **My Subscriptions**。
3. 导航到 **激活订阅** 并输入您的 16 位激活号。

### A.3. 下载发行文件

要访问 .zip、.tar.gz 和其他发布文件，请使用客户门户查找要下载的相关文件。如果您使用 RPM 软件包或 Red Hat Maven 存储库，则不需要这一步。

#### 流程

1. 打开浏览器并登录红帽客户门户网站 **产品下载页面**，网址为 [access.redhat.com/downloads](https://access.redhat.com/downloads)。
2. 查找 **INTEGRATION** 目录中的红帽 AMQ 条目。
3. 选择所需的 AMQ 产品。此时会打开 **Software Downloads** 页面。
4. 单击组件的 **Download** 链接。

### A.4. 为系统注册软件包

要在 Red Hat Enterprise Linux 上安装此产品的 RPM 软件包，必须注册您的系统。如果您使用下载的发行文件，则不需要这一步。

#### 流程

1. 转至 [access.redhat.com](https://access.redhat.com)。
2. 进入 **Registration Assistant**。
3. 选择您的操作系统版本，再继续到下一页。
4. 使用您的系统终端中列出的命令完成注册。

有关注册您的系统的更多信息，请参阅以下资源之一：

- [Red Hat Enterprise Linux 7 - 注册系统并管理订阅](#)
- [Red Hat Enterprise Linux 8 - 注册系统并管理订阅](#)

## 附录 B. 使用 RED HAT MAVEN 存储库

本节论述了如何在软件中使用红帽提供的 Maven 存储库。

### B.1. 使用在线软件仓库

红帽维护一个中央 Maven 存储库，用于基于 Maven 的项目。如需更多信息，请参阅 [存储库欢迎页面](#)。

将 Maven 配置为使用红帽存储库的方法有两种：

- [将存储库添加到您的 Maven 设置中](#)
- [将存储库添加到 POM 文件中](#)

在您的 Maven 设置中添加存储库

这种配置方法适用于用户拥有的所有 Maven 项目，只要 POM 文件没有覆盖存储库配置，并且启用了包含的配置集。

#### 流程

1. 找到 Maven **settings.xml** 文件。它通常位于用户主目录中的 **.m2** 目录中。如果文件不存在，请使用文本编辑器创建该文件。

对于 Linux 或 UNIX：

```
/home/<username>/.m2/settings.xml
```

在 Windows 上：

```
C:\Users\<username>\.m2\settings.xml
```

2. 在 **settings.xml** 文件的 **profile** 元素中添加包含红帽存储库的新配置集，如下例所示：

**示例：包含红帽存储库的 Maven settings.xml 文件**

```
<settings>
  <profiles>
    <profile>
      <id>red-hat</id>
      <repositories>
        <repository>
          <id>red-hat-ga</id>
          <url>https://maven.repository.redhat.com/ga</url>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>red-hat-ga</id>
          <url>https://maven.repository.redhat.com/ga</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>false</enabled>
          </snapshots>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

```

        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
  <activeProfiles>
    <activeProfile>red-hat</activeProfile>
  </activeProfiles>
</settings>

```

有关 Maven 配置的更多信息，请参阅 [Maven 设置参考](#)。

将存储库添加到 POM 文件中

要直接在项目中配置存储库，请在 POM 文件的 **repositories** 元素中添加一个新的条目，如下例所示：

**示例：包含红帽存储库的 Maven pom.xml 文件**

```

<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>example-app</artifactId>
  <version>1.0.0</version>

  <repositories>
    <repository>
      <id>red-hat-ga</id>
      <url>https://maven.repository.redhat.com/ga</url>
    </repository>
  </repositories>
</project>

```

有关 POM 文件配置的更多信息，请参阅 [Maven POM 参考](#)。

## B.2. 使用本地存储库

红帽为其某些组件提供基于文件的 Maven 存储库。它们作为可下载归档进行交付，您可以提取到本地文件系统中。

要将 Maven 配置为使用本地提取的存储库，请在 Maven 设置或 POM 文件中应用以下 XML：

```

<repository>
  <id>red-hat-local</id>
  <url>${repository-url}</url>
</repository>

```

**`${repository-url}`** 必须是包含提取存储库的本地文件系统路径的文件 URL。

表 B.1. 本地 Maven 存储库的 URL 示例

操作系统	文件系统路径	URL
Linux 或 UNIX	<code>/home/alice/maven-repository</code>	<code>file:/home/alice/maven-repository</code>

操作系统	文件系统路径	URL
Windows	<b>C:\repos\red-hat</b>	<b>file:C:\repos\red-hat</b>



运行完示例后，使用 **artemis stop** 命令来停止代理。

```
$ <broker-instance-dir>/bin/artemis stop
```

更新于 2025-09-04