



## Red Hat build of Quarkus 3.8

使用 YAML 文件配置红帽构建的 Quarkus 应用程序



## Red Hat build of Quarkus 3.8 使用 YAML 文件配置红帽构建的 Quarkus 应用程序

---

## 法律通告

Copyright © 2024 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.

## 摘要

本指南论述了如何使用 YAML 文件配置红帽构建的 Quarkus 应用程序。

---

## 目录

提供有关红帽构建的 QUARKUS 文档的反馈 .....	3
使开源包含更多 .....	4
第 1 章 使用 YAML 文件配置红帽构建的 QUARKUS 应用程序 .....	5
1.1. 红帽配置选项 .....	5
1.2. 添加 YAML 配置支持 .....	6
1.3. 属性表达式 .....	8
1.4. 用于在运行时配置属性的外部 APPLICATION.YAML 文件 .....	9
1.5. 管理配置属性冲突 .....	9
1.6. 其他资源 .....	10



## 提供有关红帽构建的 QUARKUS 文档的反馈

要报告错误或改进文档，请登录到 Red Hat JIRA 帐户并提交问题。如果您没有 Red Hat Jira 帐户，则会提示您创建一个帐户。

### 步骤

1. 单击以下链接 [以创建 ticket](#)。
2. 在 **Summary** 中输入问题的简短描述。
3. 在 **Description** 中提供问题或功能增强的详细描述。包括一个指向文档中问题的 URL。
4. 点 **Submit** 创建问题，并将问题路由到适当的文档团队。

## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中有问题的语言。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 使用 YAML 文件配置红帽构建的 QUARKUS 应用程序

作为应用程序开发人员，您可以使用 Red Hat build of Quarkus 创建使用 Java 编写的基于微服务的应用程序，这些应用程序在 OpenShift Container Platform 和无服务器环境中运行。编译到原生可执行文件的应用程序具有较少的内存占用率和快速启动时间。

通过更新 **application.yaml** 文件来配置 Quarkus 应用程序来应用结构化配置。



## 注意

另外，您可以通过在 **application.properties** 文件中设置属性来配置 Quarkus 应用程序。如需更多信息，[请参阅设置配置属性](#)。

该流程包括使用 Quarkus **config-quickstart** 练习创建的配置示例。



## 注意

对于入门练习的已完成示例，请下载 [Quarkus Quickstarts](#) 归档或克隆 [Quarkus Quickstarts](#) Git 存储库，再前往 **getting-started** 目录。

## 先决条件

- 已安装 OpenJDK 17 或 21，并设置 **JAVA\_HOME** 环境变量来指定 Java SDK 的位置。
  - 要下载红帽构建的 OpenJDK，请登录到红帽客户门户网站，再进入 [Software Downloads](#)。
- 已安装 Apache Maven 3.8.6 或更高版本。
  - 从 [Apache Maven Project](#) 网站下载 Maven。
- 您已将 Apache Maven 配置为使用 [Quarkus Maven](#) 存储库中的工件。
  - 要了解如何配置 Apache Maven 设置，[请参阅开始使用 Quarkus](#)。

## 1.1. 红帽配置选项

您可以使用配置选项在单个配置文件中更改应用程序的设置。Red Hat build of Quarkus 支持用于对相关属性进行分组的配置文件，并根据需要在配置集间切换。

默认情况下，Quarkus 从 **src/main/resources** 目录中的 **application.properties** 文件中读取属性。您还可以将 Quarkus 配置为从 YAML 文件中读取属性。

当您将 **quarkus-config-yaml** 依赖项添加到项目 **pom.xml** 文件时，您可以在 **application.yaml** 文件中配置和管理应用程序属性。如需更多信息，[请参阅添加 YAML 配置支持](#)。

红帽构建的 Quarkus 还支持 MicroProfile Config，可用于从其他来源加载应用的配置。

您可以使用 Eclipse MicroProfile 项目的 [MicroProfile Config](#) 规范，将配置属性注入到应用中，并使用代码中定义的方法访问它们。

Quarkus 也可以从不同的来源读取应用程序属性，包括以下源：

- 文件系统
- 数据库

- Kubernetes 或 OpenShift Container Platform **ConfigMap** 或 Secret 对象
- Java 应用程序可以加载的任何源

## 1.2. 添加 YAML 配置支持

红帽构建的 Quarkus 通过 Eclipse MicroProfile Config 的 **SmallRye Config** 实现支持 YAML 配置文件。您可以添加 **Quarkus Config YAML** 扩展，并在属性文件中使用 YAML 配置文件进行配置。Quarkus 支持使用 **application.yml** 和 **application.yaml** 作为 YAML 文件的名称。

YAML 配置文件优先于 **application.properties** 文件。要避免错误，您可以删除 **application.properties** 文件，并仅使用一种配置文件。

### 步骤

1. 使用以下方法之一在项目中添加 YAML 扩展：

- 打开 **pom.xml** 文件，将 **quarkus-config-yaml** 扩展添加为依赖项：

#### pom.xml 文件示例

```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-config-yaml</artifactId>
</dependency>
```

- 要从命令行添加 **quarkus-config-yaml** 扩展，请在项目目录中输入以下命令：

#### 添加 quarkus-config-yaml 扩展

```
./mvnw quarkus:add-extension -Dextensions="quarkus-config-yaml"
```

### 1.2.1. 将嵌套对象配置与 YAML 搭配使用

您可以使用 **application.yaml** 配置文件为红帽构建的 Quarkus 应用程序定义嵌套配置属性。

#### 先决条件

- 您有一个 Quarkus Maven 项目。
- 您有一个 PostgreSQL 数据源。
- 在项目的 **pom.xml** 文件中有以下扩展作为依赖项：
  - **quarkus-resteasy-client**
  - **quarkus-jdbc-postgresql**
  - **quarkus-config-yaml**

### 步骤

1. 打开 **src/main/resources/application.yaml** 配置文件。

- 在 `application.yaml` 文件中添加嵌套类配置属性，如下例所示：

### `application.yaml` 文件示例

```
# Properties that configure the JDBC data source driver of your PostgreSQL data source
quarkus:
  datasource:
    db-kind: postgresql
    jdbc:
      url: jdbc:postgresql://localhost:5432/quarkus_test
      username: quarkus_test
      password: quarkus_test

# Property that configures the URL of the endpoint to which the REST client sends requests
quarkus:
  rest-client:
    org.acme.rest.client.ExtensionsService:
      url: https://stage.code.quarkus.io/api

# Property that configures the log message level for your application
# For configuration property names that use quotes, do not split the string inside the quotes
quarkus:
  log:
    category:
      "io.quarkus.category":
        level: INFO
```



#### 警告

对于生产环境，请不要在配置文件中设置用户名和密码，如上例中所示。这仅用于说明目的。相反，请在环境变量中设置它们。如需更多信息，请参阅“使用 [属性文件配置红帽构建的 Quarkus 应用程序指南](#)中的设置配置属性部分。

与 `application.properties` 文件类似，您可以使用注释以 YAML 格式描述您的配置属性。



#### 注意

始终使用空格来缩进 YAML 配置文件中的属性。YAML 不支持将标签页用于缩进。

## 1.2.2. 使用 YAML 设置自定义配置配置集

使用 Quarkus，您可以设置特定于应用程序不同配置配置集的配置属性和值。您可以使用特定的配置文件启动应用程序，以访问特定的配置。此流程演示了如何以 YAML 格式为特定配置集提供配置。

### 先决条件

- 您有一个 Quarkus Maven 项目，配置为使用带有 JDBC 数据源驱动程序的 PostgreSQL 数据源。
- 您有 `quarkus-jdbc-postgresql` 和 `quarkus-config-yaml` 扩展作为项目的 `pom.xml` 文件中的依赖项。

## 步骤

1. 打开项目的配置文件 `src/main/resources/application.yaml`。
2. 要设置独立于配置集的配置，请在使用 `"%<profile_name>"` 语法定义键值对前添加配置集名称。确保将配置集名称放在引号中。

## 提示

在 YAML 中，您必须将以特殊字符开头的所有字符串放在引号内。

在以下示例中，当您以开发模式启动 Quarkus 应用程序时，PostgreSQL 数据库被配置为位于 `jdbc:postgresql://localhost:5432/some-database` URL：

### `src/main/resources/application.yaml`

```
"%dev":
  quarkus:
    datasource:
      db-kind: postgresql
      jdbc:
        url: jdbc:postgresql://localhost:5432/quarkus_test
        username: quarkus_test
        password: quarkus_test
```



### 警告

对于生产环境，请不要在配置文件中设置用户名和密码，如上例中所示。这仅用于说明目的。相反，请在环境变量中设置它们。如需更多信息，请参阅[使用属性文件配置红帽构建的 Quarkus 应用程序指南中的设置配置属性部分](#)。

## 1.3. 属性表达式

您可以将属性引用和文本字符串合并到属性表达式中，并将这些表达式用作 Quarkus 配置中的值。

与变量一样，属性表达式会动态替换配置属性值，避免了硬编码的值。

您可以在一个配置源中扩展表达式，并在另一个配置源中定义一个值。

当 `java.util.Properties` 从配置源读取属性值时，应用程序会解析属性表达式：在编译时从编译时读取，并在运行时覆盖（如果在该点被覆盖）。

如果应用无法解析表达式中属性值，并且属性没有默认值，则应用会抛出 `NoSuchElementException` 错误。

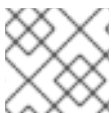
### 1.3.1. 示例：YAML 文件中的属性表达式

以下示例演示了如何将属性表达式用于 Quarkus 应用程序的灵活配置。

#### application.yaml 文件示例

```
mach: 3
x:
  factor: 2.23694

display:
  mach: ${mach}
  unit:
    name: "mph"
    factor: ${x.factor}
```



#### 注意

要引用嵌套属性，请使用 `.` (dot) 分隔符，如 `{x.factor}` 所述。

#### 其他资源

- 有关属性表达式的更多信息，请参阅 [属性表达式](#)。
- 有关在属性文件中使用属性表达式的示例，请参阅 [属性表达式的使用示例](#)。

## 1.4. 用于在运行时配置属性的外部 APPLICATION.YAML 文件

要在运行时配置应用程序属性，请将 `application.yaml` 文件添加到 `config` 目录中。

当 `config/application.yaml` 和 `src/main/resources/application.yaml` 共享属性时，`config/application.yaml` 的值会覆盖 `src/main/resources/application.yaml` 中的值。

确保 `config/application.yaml` 文件位于相对于 Quarkus 应用程序运行程序的工作目录的根目录中，如下例所示：

```
├── config
│   └── application.yaml
└── my-app-runner
```

#### 其他资源

- 有关添加 YAML 配置支持的更多信息，请参阅 [添加 YAML 配置支持](#)。

## 1.5. 管理配置属性冲突

结构化格式，如 YAML 只支持可能配置命名空间的子集。以下流程演示了如何在两个配置属性 `quarkus.http.cors` 和 `quarkus.http.cors.methods` 之间解决冲突，其中一个属性是另一个属性的前缀。

## 先决条件

- 您有一个 Quarkus 项目，它被配置为读取 YAML 配置文件。

## 步骤

1. 打开 YAML 配置文件。
2. 要将 YAML 属性定义为另一个属性的前缀，请在属性范围内添加波形符(~)，如下例所示：

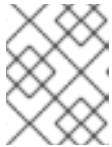
### 将 YAML 属性定义为前缀的示例

```
quarkus:  
  http:  
    cors:  
      ~: true  
      methods: GET,PUT,POST
```

3. 要以开发模式编译 Quarkus 应用程序，请从项目目录中输入以下命令：

### 编译应用程序

```
./mvnw quarkus:dev
```



### 注意

您可以在任何级别上将 YAML 键用于冲突的配置键，因为它们没有包含在配置属性名称的装配中。

## 1.6. 其他资源

- [使用属性文件配置 Quarkus 应用程序](#)