Red Hat Fuse 7.5

开始使用

开始使用红帽 Fuse！

Last Updated: 2023-07-25

Red Hat Fuse 7.5 开始使用

开始使用红帽 Fuse！

法律通告

摘要

开始使用 Fuse on Spring Boot、Apache Karaf 上的 Fuse 和 JBoss Enterprise Application Platform 上的 Fuse。

目录

前言

要开始使用 Fuse，您需要为所需容器下载并安装文件，无论是 Spring Boot、JBoss EAP 或 Apache Karaf。此处的信息和说明指导您为每个容器安装、开发和构建第一个 Fuse 应用程序。

第 1 章 在 SPRING BOOT 中使用 FUSE

要在 Spring Boot 上开发 Fuse 应用程序，请首先生成和构建在 Spring Boot 上运行的 Fuse 示例提升项目。以下主题提供详情：

- 第 1.1 节 “关于 Spring Boot 上的 Fuse”
- 第 1.2 节 “生成您的 booster 项目”
- 第 1.3 节 “构建您的 booster 项目”

1.1. 关于 SPRING BOOT 上的 FUSE

Spring Boot 是知名 Spring 容器的一个演进。Spring Boot 容器的一个独特的质量是容器功能被分成几个小块，可以独立部署。这样，您可以部署一个资源占用少、对于特定类型的服务专用的容器，这就是您需要对微服务架构的范例相符。

此容器技术的主要特性是：

- *特别适合在可扩展的云平台(Kubernetes 和 OpenShift)上运行。*
- *占用空间较小（适用于微服务架构）。*
- *与 配置相比，针对约定进行了优化。*
- *不需要应用服务器。您可以在 JVM 中直接运行 Spring Boot 应用程序 Jar。*

1.2. 生成您的 BOOSTER 项目

Fuse booster 项目存在，可帮助开发人员开始使用运行独立应用程序。此处提供的说明指导您生成其中一个增强项目，即 Circuit Breaker booster。此练习演示了 Spring Boot 上 Fuse 的有用组件。

Netflix/Hystrix 断路器使分布式应用能够处理对后端服务的网络连接和临时不可用的中断。断路器模式的基本理念是，自动检测到依赖服务的丢失，如果后端服务暂时不可用，可以编程替代行为。

Fuse 断路器提升器由两个相关服务组成：

- *名称服务，将 **名称** 返回到 greet 的后端服务。*
- *一个 **问候** 服务，即调用 **name** 服务的 frontend 服务以获取名称，然后返回字符串 **Hello, NAME**。*

*在此提升器演示中，Hystrix circuit breaker 在 **问候器** 服务和 **名称服务** 之间插入。如果 **后端名称** 服务不可用，则 **问候** 服务可能会回退到替代的行为并立即响应客户端，而不是在等待名称服务重启时被阻断。*

先决条件

- ***您必须有权访问 Red Hat Developer Platform。***
- ***您必须有受支持的 Java Developer Kit (JDK)版本。详情请查看 支持的配置页面。***

- 您必须具有 Apache Maven 3.3.x 或更高版本。

流程

1. 进入 https://developers.redhat.com/launch。
2. 单击 START。

   启动程序向导会提示您登录到您的红帽帐户。
3. 单击 登录或注册 按钮，然后登录。
4. 在 Launcher 页面上，单击 Deploy an Example Application 按钮。
5. 在 Create Example Application 页面上，在 Create Example Application 字段中输入 name fuse-circuit-breaker。
6. 点 Select an Example。
7. 在 Example 对话框中，选择 Circuit Breaker 选项。此时会出现 选择运行时 下拉菜单。
   a. 从 Select a Runtime 下拉菜单中选择 Fuse。
   b. 从版本下拉菜单中，选择 7.5 （红帽 Fuse） （不要选择 2.21.2 (Community) 版本。
   c. 点 Save。
8. 在 Create Example Application 页面中，点 Download。
9. 当您看到您的应用程序 是 Ready 对话框时，点 Download.zip。您的浏览器下载生成的 booster 项目（打包为 ZIP 文件）。

10. 使用存档实用程序将生成的项目提取到本地文件系统中的方便位置。

1.3. 构建您的 BOOSTER 项目

这些说明指导您在 Spring Boot 上使用 Fuse 构建 Circuir Breaker booster。

先决条件

- 您必须已通过 Red Hat Developer Portal 生成并下载您的 booster 项目。
- 您必须有受支持的 Java Developer Kit (JDK)版本。详情请查看 支持的配置页面。
- 您必须具有 Apache Maven 3.3.x 或更高版本。

流程

1. 打开 shell 提示符并使用 Maven 从命令行构建项目：

```
cd fuse-circuit-breaker
```

```
mvn clean package
```

Maven 构建项目后，它会显示 Build Success 消息。

2. 打开一个新的 shell 提示符并启动名称服务，如下所示：

```
cd name-service
```

```
mvn spring-boot:run -DskipTests -Dserver.port=8081
```

当 Spring Boot 启动时，您应该看到类似如下的输出：

```
...
```

```
2019-05-06 20:19:59.401  INFO 9553 --- [           main] o.a.camel.spring.SpringCamelContext
: Route: route1 started and consuming from: servlet:/name?httpMethodRestrict=GET
2019-05-06 20:19:59.402  INFO 9553 --- [           main] o.a.camel.spring.SpringCamelContext
: Total 1 routes, of which 1 are started
2019-05-06 20:19:59.403  INFO 9553 --- [           main] o.a.camel.spring.SpringCamelContext
: Apache Camel 2.21.0.fuse-730078-redhat-00001 (CamelContext: camel-1) started in 0.287
seconds
2019-05-06 20:19:59.406  INFO 9553 --- [           main] o.a.c.c.s.CamelHttpTransportServlet
: Initialized CamelHttpTransportServlet[name=CamelServlet, contextPath=]
2019-05-06 20:19:59.473  INFO 9553 --- [           main]
b.c.e.u.UndertowEmbeddedServletContainer : Undertow started on port(s) 8081 (http)
2019-05-06 20:19:59.479  INFO 9553 --- [           main]
com.redhat.fuse.boosters.cb.Application  : Started Application in 5.485 seconds (JVM
running for 9.841)
```

3. **打开一个新的 shell 提示符并启动问候服务，如下所示：**

```
cd greetings-service
```

```
mvn spring-boot:run -DskipTests
```

当 Spring Boot 启动时，您应该看到类似如下的输出：

```
...
2019-05-06 20:22:19.051  INFO 9729 --- [           main] o.a.c.c.s.CamelHttpTransportServlet
: Initialized CamelHttpTransportServlet[name=CamelServlet, contextPath=]
2019-05-06 20:22:19.115  INFO 9729 --- [           main]
b.c.e.u.UndertowEmbeddedServletContainer : Undertow started on port(s) 8080 (http)
2019-05-06 20:22:19.123  INFO 9729 --- [           main]
com.redhat.fuse.boosters.cb.Application  : Started Application in 7.68 seconds (JVM running
for 12.66)
```

greetings 服务通过 http://localhost:8080/camel/greetings URL 公开 REST 端点。

4. **通过在 web 浏览器中打开 URL 或打开另一个 shell 提示符并输入以下 curl 命令来调用 REST 端点：**

```
curl http://localhost:8080/camel/greetings
```

以下是响应：

```
{"greetings":"Hello, Jacopo"}
```

5.

要演示 Camel Hystrix 提供的断路器功能，请在名称服务的 shell 提示符窗口中键入 Ctrl-C 来终止后端名称服务。

现在，名称服务不可用，断路器在 中启动，以防止在调用时间候服务挂起。

6. 在网页浏览器中打开 http://localhost:8080/camel/greetings，或者在另一个 shell 提示符窗口中输入以下 curl 命令来调用问候 REST 端点：

```
curl http://localhost:8080/camel/greetings
```

以下是响应：

```
{"greetings":"Hello, default fallback"}
```

在运行 greetings 服务的窗口中，日志会显示以下信息序列：

```
2019-05-06 20:24:16.952  INFO 9729 --- [-CamelHystrix-2] route2                                  : Try to call name Service
2019-05-06 20:24:16.956  INFO 9729 --- [-CamelHystrix-2] o.a.c.httpclient.HttpMethodDirector     : I/O exception (java.net.ConnectException) caught when processing request: Connection refused (Connection refused)
2019-05-06 20:24:16.956  INFO 9729 --- [-CamelHystrix-2] o.a.c.httpclient.HttpMethodDirector     : Retrying request
2019-05-06 20:24:16.957  INFO 9729 --- [-CamelHystrix-2] o.a.c.httpclient.HttpMethodDirector     : I/O exception (java.net.ConnectException) caught when processing request: Connection refused (Connection refused)
2019-05-06 20:24:16.957  INFO 9729 --- [-CamelHystrix-2] o.a.c.httpclient.HttpMethodDirector     : Retrying request
2019-05-06 20:24:16.957  INFO 9729 --- [-CamelHystrix-2] o.a.c.httpclient.HttpMethodDirector     : I/O exception (java.net.ConnectException) caught when processing request: Connection refused (Connection refused)
2019-05-06 20:24:16.957  INFO 9729 --- [-CamelHystrix-2] o.a.c.httpclient.HttpMethodDirector     : Retrying request
2019-05-06 20:24:16.964  INFO 9729 --- [-CamelHystrix-2] route2                                  : We are falling back!!!!
```

7. 有关此示例的更多信息，请打开 Circuit Breaker - Red Hat Fuse 页面，地址为 http://localhost:8080/ （同时 问候服务正在运行 ）。此页面包含一个到监控断路器状态的 Hystrix 仪表板的链接。

第 2 章 在 KARAF 上使用 FUSE

要了解 Fuse on Karaf 上的 Fuse 以及在 Karaf 容器上安装、开发和构建第一个 Fuse 应用程序的信息，此处的信息和说明可帮助您实现这一目的。详情请查看以下主题：

- 第 2.1 节 “关于 Karaf 上的 Fuse”
- 第 2.2 节 “在 Karaf 上安装 Fuse”
- 第 2.3 节 “在 Karaf 上构建第一个 Fuse 应用程序”

2.1. 关于 KARAF 上的 FUSE

Apache Karaf 基于 OSGi 联盟中的 OSGi 标准。OSGi 源自电信行业，它用于开发可即时升级的网关服务器，而无需关闭服务器（称为 热代码交换）。因此，SGiOS 容器技术发现了各种其他用途，并被模块化应用程序（例如，Eclipse IDE）。

此容器技术的主要特性是：

- 特别适用于在独立模式运行。
- 对模块化(OSGi 捆绑包)的强支持，具有复杂的类加载支持。
- 容器中的多个依赖项版本可以并行部署（但在实践中需要注意）。
- 热代码交换功能，允许您在不关闭容器的情况下升级或替换模块。这是一个独特的功能，但为了使其正常工作，需要大量的努力。

注意：不支持 Spring Dynamic Modules (Spring-DM)（将 Spring XML 与 Apache Karaf 中的 OSGi 服务层集成）。反之，您应该使用 Blueprint 框架。使用 Blueprint XML 不会阻止您使用 Spring 框架中的 Java 库：最新版本的 Spring 与 Blueprint 兼容。

2.2. 在 KARAF 上安装 FUSE

FUSE 7.5 上的 Fuse 7.5 的标准安装软件包可从红帽客户门户下载。它安装 datacenter 容器的标准装配，并提供完整的 Fuse 技术堆栈。

先决条件

- 在红帽客户门户网站中 您需要一个全订阅帐户。
- 您必须登录到客户门户网站。
- 您必须已下载 CodeReady Studio 安装程序。
- 您必须在 Karaf 安装程序中下载 Fuse。

流程

1. 将 Apache Karaf 上 Fuse 下载的 .zip 归档文件解压缩到您的文件系统中 FUSE_INSTALL 的便捷位置。
2. 将管理员用户添加到 Fuse 运行时。
   a. 在文本编辑器中打开 FUSE_INSTALL/etc/users.properties 文件。
   b. 删除以 # admin = admin 开头的行开头的 # 字符。
   c. 删除以 # g\:admingroup 开头的行开头的 # 字符。
   d. 自定义用户条目的用户名、USERNAME 和密码 PASSWORD，以便您有一个用户条目和 admin 组条目，如下所示（连续行）：

```
USERNAME = PASSWORD,_g_:admingroup
_g_\:admingroup = group,admin,manager,viewer,systembundles,ssh
```

e. 保存 etc/users.properties 文件。

3. 运行 CodeReady Studio 安装程序，如下所示：

```
java -jar DOWNLOAD_LOCATION/codereadystudio-12.13.0.GA-installer-standalone.jar
```

4. 在安装过程中：

a. 接受条款和条件。

b. 选择您首选的安装路径。

c. 选择 Java 8 JVM。

d. 在 Select Platforms and Servers 步骤中，通过单击 Add 并浏览到 FUSE_INSTALL 目录的位置，配置 FUSE 运行时上的 Fuse。

e. 在 Select Additional Features to Install 步骤中，选择 Red Hat Fuse Tooling。

5. CodeReady Studio 启动。出现 Searching for runtime 对话框时，单击 OK 以在 Karaf 运行时上创建 Fuse。

6. （可选）为了从命令行使用 Apache Maven，您需要安装和配置 Maven。

注意

如果您只使用 CodeReady Studio，则不需要安装 Maven，因为 CodeReady Studio 已为您预安装并配置了 Maven。但是，如果您计划从命令行调用 Maven，则需要执行此步骤。

2.3. 在 KARAF 上构建第一个 FUSE 应用程序

这组说明可帮助您在 Karaf 上构建第一个 Fuse 应用程序。

先决条件

- 在红帽客户门户网站中 您需要一个全订阅帐户。
- 您必须登录到客户门户网站。
- 您必须已下载 CodeReady Studio 安装程序。
- 您必须已在 Karaf 上下载并成功安装了 Fuse。

流程

1. 在 CodeReady Studio 中，创建一个新项目，如下所示：
   a. 选择 File→New→Fuse Integration Project。
   b. 在 Project Name 字段中输入 fuse-camel-cbr。
   c. 点击 Next。
   d. 在 Select a Target Environment 窗格中选择以下设置：
      - 选择 Standalone 作为部署平台。
      - 选择 Karaf 上的 Karaf/Fuse 作为运行时环境，并使用 Runtime （可选） 下拉菜单选择 Red Had JBoss Middleware> 红帽 Fuse 7+ Runtime 服务器作为目标运行时。
   e. 选择目标运行时后，会自动为您选择 Camel Version，字段将灰掉。

f. 点击 **Next**。

g. 在 **Advanced Project Setup** 窗格中，选择 **Beginner→Content Based Router - Blueprint DSL** 模板。

h. 点 **Finish**。

i. 如果系统提示您打开关联的 **Fuse** 集成透视图，请单击 **Yes**。

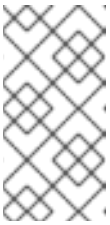
j. 等待 **CodeReady Studio** 下载所需的工件，并在后台构建项目。

重要

如果您在 **CodeReady Studio** 中首次构建 **Fuse** 项目时，向导将需要几分钟才能完成生成项目，因为它会从远程 **Maven** 存储库下载依赖项。在项目在后台构建时，请勿尝试中断向导或关闭 **CodeReady Studio**。

2. 将项目部署到服务器，如下所示：

a. 在 **Servers** 视图(**Fuse Integration** 视角的左下角)中，如果服务器尚未启动，请选择 **fuse-karaf-7.5.0.fuse-750035-redhat-00001 Runtime Server** 服务器，然后单击绿色箭头来启动它。



注意

如果您看到对话框，警告：无法建立主机"localhost"的真实性。 单击 **Yes** 以连接到服务器并访问 **Karaf** 控制台。

b. 等待 **Console** 视图中看到类似如下的消息：

```
Karaf started in 1s. Bundle stats: 12 active, 12 total
```

c. 服务器启动后，切回到 **Servers** 视图，右键单击服务器，然后从上下文菜单中选择 **Add and Remove**。

d.

在 Add and Remove 对话框中，选择 fuse-camel-cbr 项目，然后点击 Add > 按钮。

e.

点 Finish。

f.

您可以通过进入 Terminal 视图并输入 bundle:list | tail 来检查项目的 OSGi 捆绑包是否已启动。您应该看到类似如下的输出：

```
...
228 | Active |  80 | 1.0.0.201505202023          | org.osgi:org.osgi.service.j
232 | Active |  80 | 1.0.0.SNAPSHOT              | Fuse CBR Quickstart
```

注意

Camel 路由启动后，它将在 fuse-camel-cbr 项目中创建一个目录 work/cbr/input。

3.

在 Project Explorer 视图中，单击 Refresh 以查看新创建的 work/cbr/input 目录。

4.

将项目的 src/main/data 目录中找到的文件复制到 work/cbr/input 目录中。

5.

等待几分钟，然后再次刷新 Project Explorer 视图，以查看 work/cbr/output 目录下的国家（地区）组织相同的文件：

a.

work/cbr/output/others中的 order1.xml

b.

work/cbr/output/uk中的 order2.xml 和 order4.xml

c.

order3.xml 和 order5.xml in work/cbr/output/us

6.

取消部署项目，如下所示：

a.

在 Servers 视图中，选择 Red Hat Fuse 7+ Runtime Server 服务器。

b.

右键单击服务器，然后从上下文菜单中选择 Add and Remove。

c.

在 Add and Remove 对话框中，选择 fuse-camel-cbr 项目，再单击 &lt ; Remove 按钮。

d.

点 Finish。

第 3 章 在 JBOSS EAP 上使用 FUSE

本章介绍了 JBoss EAP 上的 Fuse，并解释了如何在 JBoss EAP 容器上安装、开发和构建第一个 Fuse 应用程序。

详情请查看以下主题：

- 第 3.1 节 “关于 JBoss EAP 上的 Fuse”
- 第 3.2 节 “在 JBoss EAP 上安装 Fuse”
- 第 3.3 节 “在 JBoss EAP 上构建第一个 Fuse 应用程序”

3.1. 关于 JBOSS EAP 上的 FUSE

JBoss Enterprise Application Platform (EAP)基于 Eclipse Foundation 中的 Jakarta EE 技术（之前是 Java EE），最初创建为开发企业应用程序的用例而创建。JBoss EAP 的特征是实施服务和标准化的 Java API （例如，持久性、消息传递、安全性等）的定义模式。近年来，此技术已演变为更加轻量级，为企业 Java Bean 的依赖性注入和简化的注解推出 CDI。

此容器技术的主要特性是：

- 特别适用于在独立模式运行。
- 许多标准服务（如持久性、消息传递、安全性等）预先配置并提供开箱即用。
- 应用程序 WAR 通常比较小且轻量级（因为容器中预装了许多依赖项）。
- 标准化、向后兼容的 Java API.

3.2. 在 JBOSS EAP 上安装 FUSE

JBoss EAP 上 Fuse 7.5 的标准安装软件包可从红帽客户门户下载。它安装 JBoss EAP 容器的标准装配，并提供完整的 Fuse 技术堆栈。

先决条件

- 您必须 在红帽客户门户网站中 有一个完整的订阅帐户。
- 您必须登录到客户门户网站。
- 您必须已下载了 JBoss EAP 和 JBoss EAP 7.2 Update 04。
- 您必须在 JBoss EAP 上下载 Fuse。
- 您必须已下载 CodeReady Studio 安装程序。

流程

1. 从 shell 提示符运行 JBoss EAP 安装程序，如下所示：

```
java -jar DOWNLOAD_LOCATION/jboss-eap-7.2.0-installer.jar
```

2. 在安装过程中：
   a. 接受条款和条件。
   b. 为 JBoss EAP 运行时选择您首选的安装路径 EAP_INSTALL。
   c. 创建管理用户，并小心记录这些管理用户凭证。
   d. 您可以在剩余的屏幕上接受默认设置。

3. 打开 shell 提示符，并将目录更改为 EAP_INSTALL。

4. 在 EAP_INSTALL 目录中，应用 JBoss EAP 7.2 Update 04。例如：

```
bin/jboss-cli.sh "patch apply jboss-eap-7.2.4-patch.zip"
```

5. 在 EAP_INSTALL 目录中，在 EAP 安装程序上运行 Fuse，如下所示：

```
java -jar DOWNLOAD_LOCATION/fuse-eap-installer-7.5.0.jar
```

6. 运行 CodeReady Studio 安装程序，如下所示：

```
java -jar DOWNLOAD_LOCATION/codereadystudio-12.13.0.GA-installer-standalone.jar
```

7. 在安装过程中：

   a. 接受条款和条件。

   b. 选择您首选的安装路径。

   c. 选择 Java 8 JVM。

   d. 在 Select Platforms and Servers 步骤中，单击 Add 并浏览到 EAP_INSTALL 目录的位置来配置 JBoss EAP 运行时。

   e. 在 Select Additional Features to Install 步骤中，选择 Red Hat Fuse Tooling。

8. CodeReady Studio 启动。出现 搜索运行时 对话框时，单击 OK 以创建 JBoss EAP 运行时。

9. （可选）为了从命令行使用 Apache Maven，您需要安装和配置 Maven。

注意

如果您只使用 CodeReady Studio，则不需要安装 Maven，因为 CodeReady Studio 已预安装并配置了 Maven。但是，如果您计划从命令行调用 Maven，则必须执行此步骤。

3.3. 在 JBOSS EAP 上构建第一个 FUSE 应用程序

这组说明可帮助您在 JBoss EAP 上构建第一个 Fuse 应用程序。

先决条件

- 在红帽客户门户网站中您需要一个全订阅帐户。
- 您必须登录到客户门户网站。
- 您必须在 JBoss EAP 上下载并成功安装了 Fuse。
- 您必须已下载并成功安装了 CodeReady Studio 安装程序。

流程

1. 在 CodeReady Studio 中，创建一个新项目，如下所示：
   a. 选择 File→New→Fuse Integration Project。
   b. 在 Project Name 字段中，输入 eap-camel。
   c. 点击 Next。
   d. 在 Select a Target Environment 窗格中选择以下设置：

- 选择 **Standalone** 作为部署平台。

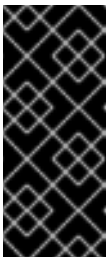
- 选择 **EAP 上的 Wildfly/Fuse** 作为运行时环境，并使用 **Runtime （可选） 下拉菜单选择 JBoss EAP 7.x Runtime** 服务器作为目标运行时。

e. 选择目标运行时后，会自动为您选择 **Camel Version，字段将灰掉。**

f. **点击 Next。**

g. **在 Advanced Project Setup 窗格中，选择 Spring Bean - Spring DSL 模板。**

h. **点 Finish。**



重要

如果您在 CodeReady Studio 中首次构建 Fuse 项目时，向导将需要几分钟时间才能完成生成项目。这是因为它从远程 Maven 存储库下载依赖项。在项目在后台构建时，不要中断向导或关闭 CodeReady Studio。

i. **如果系统提示您打开关联的 Fuse 集成透视图，请单击 Yes。**

j. **等待 CodeReady Studio 下载所需的工件，并在后台构建项目。**

2. **将项目部署到服务器，如下所示：**

a. **在 Servers 视图(Fuse Integration 视角的左下角)中，如果服务器尚未启动，请选择 Red Hat JBoss EAP 7.2 Runtime 服务器，然后单击绿色箭头以启动它。**

b. **等待 Console 视图中看到类似如下的消息：**

> *14:47:07,283 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: JBoss EAP 7.2.0.GA (WildFly Core 6.0.11.Final-redhat-00001) started in 13948ms - Started 495 of 680 services (326 services are lazy, passive or on-demand)*

c. *服务器启动后，切回到 Servers 视图，右键单击服务器，然后从上下文菜单中选择 Add and Remove。*

d. *在 Add and Remove 对话框中，选择 eap-camel 项目并点 Add >。*

e. *点 Finish。*

3. *验证项目是否正常工作，如下所示：*

a. *浏览到以下 URL 以访问 eap-camel 项目中运行的服务：http://localhost:8080/camel-test-spring?name=Kermit*

b. *浏览器窗口应当显示响应 Hello Kermit。*

4. *取消部署项目，如下所示：*

a. *在 Servers 视图中，选择 Red Hat JBoss EAP 7.2 Runtime 服务器。*

b. *右键单击服务器，然后从上下文菜单中选择 Add and Remove。*

c. *在 Add and Remove 对话框中，选择 eap-camel 项目，再点 < Remove。*

d. *点 Finish。*

第 4 章 在本地设置 MAVEN

典型的 Fuse 应用程序开发使用 Maven 来构建和管理项目。

以下主题描述了如何在本地设置 Maven：

- 第 4.1 节 “准备设置 Maven”
- 第 4.2 节 “将红帽软件仓库添加到 Maven”
- 第 4.3 节 “使用本地 Maven 存储库”
- 第 4.4 节 “关于 Maven 工件和协调”

4.1. 准备设置 MAVEN

Maven 是一个来自 Apache 的免费开源构建工具。通常，您可以使用 Maven 构建 Fuse 应用程序。

流程

1. 从 Maven 下载页面 下载最新版本的 Maven。
2. 确定您的系统已连接到互联网。

   在构建项目时，默认行为是 Maven 搜索外部存储库并下载所需的工件。Maven 查找可通过互联网访问的存储库。

   您可以更改此行为，以便 Maven 仅搜索位于本地网络上的存储库。也就是说，Maven 可以在离线模式下运行。在离线模式中，Maven 会在其本地存储库中查找工件。请参阅 第 4.3 节 “使用本地 Maven 存储库”。

4.2. 将红帽软件仓库添加到 MAVEN

要访问位于 Red Hat Maven 存储库中的工件，您需要将这些存储库添加到 Maven 的 settings.xml 文件中。Maven 在用户主目录的 .m2 目录中查找 settings.xml 文件。如果没有用户指定的 settings.xml 文件，Maven 将使用 M2_HOME/conf/ settings.xml 中的系统级 settings.xml 文件。

前提条件

- 您知道要在其中添加红帽软件仓库的 settings.xml 文件的位置。

流程

1. 在 settings.xml 文件中，为红帽 软件仓库 添加软件仓库元素，如下例所示：

```
<?xml version="1.0"?>
<settings>

  <profiles>
    <profile>
      <id>extra-repos</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <repositories>
       <repository>
          <id>redhat-ga-repository</id>
          <url>https://maven.repository.redhat.com/ga</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>false</enabled>
          </snapshots>
       </repository>
       <repository>
          <id>redhat-ea-repository</id>
          <url>https://maven.repository.redhat.com/earlyaccess/all</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>false</enabled>
          </snapshots>
       </repository>
       <repository>
        <id>jboss-public</id>
        <name>JBoss Public Repository Group</name>
        <url>https://repository.jboss.org/nexus/content/groups/public/</url>
       </repository>
      </repositories>
      <pluginRepositories>
       <pluginRepository>
          <id>redhat-ga-repository</id>
```

```
        <url>https://maven.repository.redhat.com/ga</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>false</enabled>
        </snapshots>
      </pluginRepository>
      <pluginRepository>
        <id>redhat-ea-repository</id>
        <url>https://maven.repository.redhat.com/earlyaccess/all</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>false</enabled>
        </snapshots>
      </pluginRepository>
      <pluginRepository>
        <id>jboss-public</id>
        <name>JBoss Public Repository Group</name>
        <url>https://repository.jboss.org/nexus/content/groups/public</url>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>

<activeProfiles>
  <activeProfile>extra-repos</activeProfile>
</activeProfiles>

</settings>
```

4.3. 使用本地 MAVEN 存储库

如果您在没有互联网连接的情况下运行 Apache Karaf 容器，并且您需要部署一个具有离线不可用依赖项的应用程序，您可以使用 Maven 依赖项插件将应用的依赖项下载到 Maven 离线存储库中。然后，您可以将此自定义 Maven 离线存储库分发到没有互联网连接的机器。

流程

1. 在包含 pom.xml 文件的项目目录中，运行以下命令来下载 Maven 项目的存储库，如下所示：

```
mvn org.apache.maven.plugins:maven-dependency-plugin:3.1.0:go-offline -
Dmaven.repo.local=/tmp/my-project
```

在本例中，构建项目所需的 Maven 依赖项和插件将下载到 /tmp/my-project 目录中。

2. 编辑 etc/org.ops4j.pax.url.mvn.cfg 文件，将 org.ops4j.pax.url.mvn.offline 设置为 true。这可启用离线模式：

```
##
# If set to true, no remote repository will be accessed when resolving artifacts
#
org.ops4j.pax.url.mvn.offline = true
```

3. 将此自定义 Maven 离线存储库在内部分发到没有互联网连接的任何机器。

4.4. 关于 MAVEN 工件和协调

在 Maven 构建系统中，基本构建块是一个 工件。构建后，工件的输出通常是一个存档，如 JAR 或 WAR 文件。

Maven 的一个关键方面是能够找到工件和管理它们之间的依赖关系。Maven 协调 是一组用于标识特定工件位置的值。基本协调有三个值，格式为：

groupId:artifactId:version

有时，Maven 通过 打包值或使用打包 值和 分类器 值增加一个基本的协调。Maven 协调可以具有以下形式之一：

```
groupId:artifactId:version
groupId:artifactId:packaging:version
groupId:artifactId:packaging:classifier:version
```

以下是值的描述：

groupdId

定义工件名称的范围。您通常使用所有或部分软件包名称作为组 ID。例如，org.fusesource.example。

artifactId

定义相对于组 ID 的工件名称。

version

指定工件的版本。版本号最多可有四个部分：n.n.n.n，其中版本号的最后一部分可以包含非数字字符。例如，1.0-SNAPSHOT 的最后一部分是字母数字字符子字符串 0-SNAPSHOT。

打包

定义构建项目时生成的打包实体。对于 OSGi 项目，打包是 捆绑包。默认值为 jar。

classifier

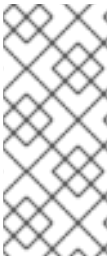
可让您区分从同一 POM 构建但具有不同内容的工件。

工件的 POM 文件中的元素定义工件的组 ID、工件 ID、打包和版本，如下所示：

```
<project ... >
  ...
  <groupId>org.fusesource.example</groupId>
  <artifactId>bundle-demo</artifactId>
  <packaging>bundle</packaging>
  <version>1.0-SNAPSHOT</version>
  ...
</project>
```

要定义对上述工件的依赖项，您可以在 POM 文件中添加以下 dependencies 元素：

```
<project ... >
  ...
  <dependencies>
    <dependency>
      <groupId>org.fusesource.example</groupId>
      <artifactId>bundle-demo</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>
  </dependencies>
  ...
</project>
```



注意

不需要在前面的依赖项中指定 bundle 软件包类型，因为捆绑包只是特定类型的 JAR 文件，jar 是默认的 Maven 软件包类型。但是，如果您需要在依赖项中明确指定打包类型，您可以使用 type 元素。