



# Red Hat Fuse 7.8

## 管理 Fuse

使用 Fuse 控制台管理 Fuse 应用程序



## Red Hat Fuse 7.8 管理 Fuse

---

使用 Fuse 控制台管理 Fuse 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

部署 Fuse 应用程序时，您可以使用 Fuse 控制台监控并与红帽 Fuse 集成交互。

# 目录

前言 .....	4
<b>第 1 章 在 OPENSIFT 中监控和管理红帽 FUSE 应用程序 .....</b>	<b>5</b>
1.1. 关于 FUSE 控制台	5
1.2. 在 OPENSIFT 4.X 中设置 FUSE 控制台	5
1.3. 在 OPENSIFT 3.11 中设置 FUSE 控制台	20
1.4. 查看容器和应用程序	26
1.5. 查看并管理 APACHE CAMEL 应用程序	26
1.6. 查看 AMQ BROKER	32
1.7. 查看并管理 JMX 域和 MBEANS	33
1.8. 查看诊断	34
1.9. 查看线程	35
1.10. 确保数据在 FUSE 控制台中正确显示	35
<b>第 2 章 在 SPRING BOOT 独立上监控和管理 RED HAT FUSE 应用程序 .....</b>	<b>37</b>
2.1. 关于 FUSE 控制台	37
2.2. 访问 SPRING BOOT 2.X 的 FUSE 控制台	37
2.3. 自定义 FUSE 控制台品牌	40
2.4. 保护 FUSE 控制台	41
2.5. 确保数据在 FUSE 控制台中正确显示	43
2.6. 连接到远程 FUSE 应用程序	43
2.7. 查看并管理 APACHE CAMEL 应用程序	45
2.8. 查看并管理 JMX 域和 MBEANS	51
2.9. 查看诊断	52
2.10. 查看线程	53
<b>第 3 章 在 KARAF 独立上监控和管理红帽 FUSE 应用程序 .....</b>	<b>54</b>
3.1. 关于 FUSE 控制台	54
3.2. 访问 FUSE 控制台	54
3.3. 保护 FUSE 控制台	55
3.4. 基于角色的访问控制参考	57
3.5. 自定义 FUSE 控制台品牌	59
3.6. 确保数据在 FUSE 控制台中正确显示	61
3.7. 禁用 FUSE 控制台	62
3.8. 连接到远程 FUSE 应用程序	62
3.9. 查看并管理 APACHE CAMEL 应用程序	66
3.10. 查看并管理 JMX 域和 MBEANS	72
3.11. 查看和管理您的 OSGI 环境	73
3.12. 查看诊断	74
3.13. 查看线程	75
3.14. 查看日志条目	75
3.15. 启用 PROMETHEUS 指标	77
<b>第 4 章 在 EAP 独立上监控和管理红帽 FUSE 应用程序 .....</b>	<b>79</b>
4.1. 关于 FUSE 控制台	79
4.2. 访问 FUSE 控制台	79
4.3. 自定义 FUSE 控制台品牌	80
4.4. 保护 FUSE 控制台	82
4.5. 确保数据在 FUSE 控制台中正确显示	83
4.6. 禁用 FUSE 控制台	84
4.7. 连接到远程 FUSE 应用程序	84
4.8. 查看并管理 APACHE CAMEL 应用程序	88

4.9. 查看并管理 JMX 域和 MBEANS	94
4.10. 查看诊断	95
4.11. 查看线程	96
4.12. 查看日志条目	96
<b>附录 A. FUSE 控制台配置属性</b> .....	<b>99</b>



## 前言

Red Hat Fuse 为查看和管理 Fuse 集成提供了两个企业监控工具：

- Fuse 控制台是一个基于 Web 的控制台，您可以从浏览器访问以监控和管理正在运行的 Fuse 容器。Fuse 控制台基于 Hawtio 开源软件(<https://hawt.io/>)。本指南论述了如何使用 Fuse 控制台。
- Prometheus 为 Fuse 发行版存储系统和集成级别指标。您可以使用图形分析界面（如 Grafana）来查看和分析存储的历史数据。如需了解更多有关使用 Prometheus 的信息，请参阅以下文档：
  - [Prometheus 文档](#)
  - [OpenShift 上的 Fuse 指南](#)
  - [在 OpenShift Container Platform 上安装和操作 Fuse Online](#)

本指南对象是红帽 Fuse 管理员。本指南假定您已熟悉红帽 Fuse 平台、Apache Camel 和组织的处理要求。



# 第 1 章 在 OPENSIFT 中监控和管理红帽 FUSE 应用程序

## 1.1. 关于 FUSE 控制台

红帽 Fuse 控制台是一个基于 Hawtio 开源软件的 Web 控制台。如需支持的浏览器列表，请访问 [支持的配置](#)。

Fuse 控制台提供了一个中央接口，用于检查和管理一个或多个部署 Fuse 容器的详细信息。您还可以监控红帽 Fuse 和系统资源，执行更新，以及启动或停止服务。

安装 Red Hat Fuse 独立或使用 OpenShift 中的 Fuse 时，可以使用 Fuse 控制台。您可以在 Fuse 控制台中查看和管理的集成取决于正在运行的插件。可能的插件包括：

- Camel
- JMX
- OSGI
- Runtime
- Logs

## 1.2. 在 OPENSIFT 4.X 中设置 FUSE 控制台

在 OpenShift 4.x 中，设置 Fuse 控制台涉及保护、安装和部署它。您可以选择安装和部署 Fuse 控制台：

- [第 1.2.1 节 “使用 OperatorHub 在 OpenShift 4.x 上安装和部署 Fuse 控制台”](#)  
您可以使用 Fuse Console Operator 来安装和部署 Fuse 控制台，以便可以访问特定命名空间中的 Fuse 应用程序。
- [第 1.2.2 节 “使用命令行在 OpenShift 4.x 上安装和部署 Fuse 控制台”](#)  
您可以使用命令行和 Fuse 控制台模板之一来安装和部署 Fuse 控制台，以便它能够在 OpenShift 集群或特定命名空间中访问 Fuse 应用程序。

另外，您可以为 Fuse 控制台实施基于角色的访问控制(RBAC)，如 [第 1.2.3 节 “OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制”](#) 所述。

### 1.2.1. 使用 OperatorHub 在 OpenShift 4.x 上安装和部署 Fuse 控制台

要在 OpenShift 4.x 上安装 Fuse 控制台，您可以使用 OpenShift OperatorHub 中提供的 Fuse Console Operator。要部署 Fuse 控制台，请创建一个已安装 Operator 实例。

#### 前提条件

如果要为 Fuse 控制台自定义基于角色的访问控制(RBAC)，则必须在安装 Fuse Console Operator 的同一 OpenShift 命名空间中有一个 RBAC 配置映射文件。如果要使用默认的 RBAC 行为，如 [OpenShift 4.x 上 Fuse Console 的基于角色的访问控制](#) 中所述，则不需要提供配置映射文件。

#### 流程

要安装和部署 Fuse 控制台：

1. 以具有集群管理员访问权限的用户身份登录 Web 浏览器中的 OpenShift 控制台。

2. 点 **Operators**，然后点 **OperatorHub**。
3. 在搜索字段窗口中，键入 **Fuse Console** 来过滤操作器列表。
4. 单击 **Fuse Console Operator**。
5. 在 **Fuse Console Operator** 安装窗口中，单击 **Install**。

**Create Operator Subscription** 表单将打开。

- 对于 **更新频道**，请选择 **fuse-console-7.8.x**。
- 对于 **Installation Mode**，接受 **default**（集群中的特定命名空间）。

请注意，在安装 **Operator** 后，当部署 **Fuse** 控制台时，您可以选择监控集群中的所有命名空间中的应用程序，或者仅监控安装 **Fuse Console operator** 的命名空间中的应用程序。

- 对于 **Installed Namespace**，选择要在其中安装 **Fuse Console Operator** 的命名空间。
- 对于 **批准策略**，您可以选择 **Automatic** 或 **Manual** 来配置 **OpenShift** 如何处理对 **Fuse Console Operator** 的更新。
  - 如果选择 **Automatic** 更新，当 **Fuse Console Operator** 的新版本可用时，**OpenShift Operator Lifecycle Manager (OLM)** 将自动升级 **Fuse** 控制台的运行实例，而无需人为干预。
  - 如果选择手动更新，则当有新版 **Operator** 可用时，**OLM** 会创建更新请求。作为集群管理员，您必须手动批准该更新请求，才能将 **Fuse Console Operator** 更新至新版本。

6. 点 **Install**。

OpenShift 在当前命名空间中安装 Fuse Console Operator。

7. 要验证安装，点 Operators，然后点 Installed Operators。您可以在 Operator 列表中看到 Fuse Console。

8. 使用 OpenShift Web 控制台部署 Fuse 控制台：

- a. 在 Installed Operators 列表中，单击 Name 列下的 Fuse Console。
- b. 在 Provided APIs 下的 Operator Details 页面中，点 Create Instance。

接受配置默认值或选择性地编辑它们。

对于 Replicas，如果要提高 Fuse 控制台性能（例如在高可用性环境中），您可以增加分配给 Fuse 控制台的 pod 数量。

对于 Rbac（基于角色的访问控制），如果要为 Fuse 控制台启用 RBAC，在 Form 视图图中将 rbac > Enabled 选项设置为 true。另外，在 YAML 视图中，编辑该文件以在 spec 下包含以下条目：

```
rbac:  
  enabled: true
```

如果要自定义默认的 RBAC 行为，且 ConfigMap 文件已在安装 Fuse Console Operator 的命名空间中存在时，才在 config Map 字段中指定一个值。如需有关 RBAC 的更多信息，请参阅 [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)。

- c. 点 Create。

Fuse Console Operator Details 页面将打开，并显示部署的状态。

9. 打开 Fuse 控制台：

- a.

对于命名空间部署：在 OpenShift Web 控制台中，打开安装 **Fuse Console operator** 的项目，然后选择 **Overview**。在 **Project Overview** 页面中，向下滚动到 **Launcher** 部分，再单击 **Fuse Console URL** 将它打开。

对于集群部署，在 OpenShift Web 控制台的标题栏中，点网格图标(



)。在弹出菜单中，在红帽应用程序下，单击 **Fuse Console URL** 链接。

b.

登录 **Fuse** 控制台。

在列出所需权限的浏览器中打开 **Authorize Access** 页面。

c.

单击 **Allow selected permissions**。

**Fuse Console** 在浏览器中打开，并显示您具有访问权限的 **Fuse** 应用容器集。

10.

点您要查看的应用程序的 **Connect**。

这时将打开一个新的浏览器窗口，显示 **Fuse** 控制台中的应用程序。

### 1.2.2. 使用命令行在 OpenShift 4.x 上安装和部署 Fuse 控制台

在 OpenShift 4.x 中，您可以选择其中一个部署选项来从命令行安装和部署 **Fuse** 控制台：

- **集群 - Fuse Console** 可以发现并连接到 OpenShift 集群上的多个命名空间（项目）部署的 **Fuse** 应用程序。若要部署此模板，您必须具有 OpenShift 集群的管理员角色。
- **具有基于角色的访问控制的集群 - 具有可配置基于角色的访问控制(RBAC)的集群模板**。如需更多信息，请参阅 [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)。
- **namespace - Fuse 控制台** 有权访问特定的 OpenShift 项目（命名空间）。若要部署此模板，您必须具有 OpenShift 项目的管理员角色。
- **具有基于角色的访问控制的命名空间 - 带有可配置 RBAC 的命名空间模板**。如需更多信息，请参阅 [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)。

要查看 **Fuse Console** 模板的参数列表，请运行以下命令：

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fuse-console-namespace-os4.json
```

### 先决条件

- 在安装并部署 **Fuse** 控制台之前，您必须生成使用服务签名证书颁发机构签名的客户端证书，如 [生成证书来保护 OpenShift 4.x 上的 Fuse 控制台](#) 中所述。
- 具有 **OpenShift** 集群的集群管理员角色。
- 已安装 **Fuse** 控制台镜像流（以及其他 **Fuse** 镜像流），如在 [OpenShift 4.x 服务器上安装 Fuse 镜像流和模板](#) 中所述。

### 流程

1. 使用以下命令检索所有模板的列表，验证是否安装了 **Fuse Console** 镜像流：

```
oc get template -n openshift
```

2. 另外，如果要使用新的发行版本标签更新已安装的镜像流，请使用以下命令将 **Fuse Console** 镜像导入到 **openshift** 命名空间：

```
oc import-image fuse7/fuse7-console:1.8 --from=registry.redhat.io/fuse7/fuse-console:1.8 --confirm -n openshift
```

3. 运行以下命令来获取 **Fuse Console APP\_NAME** 值：

```
oc process --parameters -f TEMPLATE-FILENAME
```

其中 **TEMPLATE-FILENAME** 是以下模板之一：

- 集群模板：  
  
<https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fuse-console-cluster-os4.json>
- 带有可配置 RBAC 的集群模板：  
  
<https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fuse-console-cluster-rbac.yml>
- 命名空间模板：  
  
<https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fuse-console-namespace-os4.json>
- 带有可配置 RBAC 的命名空间模板：  
  
<https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fuse-console-namespace-rbac.yml>

例如，对于带有可配置 RBAC 的集群模板，运行以下命令：

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fuse-console-cluster-rbac.yml
```

4. 从您在 [OpenShift 4.x 上保护 Fuse 控制台](#) 生成的证书，使用以下命令创建 `secret` 并将其挂载到 Fuse 控制台中（其中 `APP_NAME` 是 Fuse 控制台应用程序的名称）。

```
oc create secret tls APP_NAME-tls-proxying --cert server.crt --key server.key
```

5. 运行以下命令，根据 Fuse Console 模板的本地副本创建一个新应用程序（其中 `myproject` 是 OpenShift 项目的名称），`mytemp` 是包含 Fuse 控制台模板的本地目录的路径，`myhost` 是访问 Fuse 控制台的主机名：

- 对于集群模板：

```
oc new-app -n myproject -f {templates-base-url}/fuse-console-cluster-os4.json -p
ROUTE_HOSTNAME=myhost"
```

- 对于使用 RBAC 模板的集群：

```
oc new-app -n myproject -f {templates-base-url}/fuse-console-cluster-rbac.yml -p
ROUTE_HOSTNAME=myhost"
```

- 对于命名空间模板：

```
{templates-base-url}/fuse-console-namespace-os4.json
```

- 对于使用 RBAC 模板的命名空间：

```
oc new-app -n myproject -f {templates-base-url}/fuse-console-namespace-rbac.yml
```

6. 要配置 Fuse 控制台使其能够打开 OpenShift Web 控制台，请运行以下命令设置 `OPENSIFT_WEB_CONSOLE_URL` 环境变量：

```
oc set env dc/${APP_NAME} OPENSIFT_WEB_CONSOLE_URL=`oc get -n openshift-
config-managed cm console-public -o jsonpath={.data.consoleURL}`
```

7. 运行以下命令，获取 Fuse Console 部署的状态和 URL：

```
oc status
```

8. 要从浏览器访问 Fuse 控制台，请使用步骤 7 返回的 URL（例如 <https://fuse-console.192.168.64.12.nip.io>）。

### 1.2.2.1. 生成证书来保护 OpenShift 4.x 上的 Fuse 控制台

在 OpenShift 4.x 中，若要在 Fuse Console 代理和 Jolokia 代理安全之间保持连接，必须在部署 Fuse 控制台前生成客户端证书。服务签名证书颁发机构私钥必须用于为客户端证书签名。

只有在 使用命令行安装和部署 Fuse 控制台时，您必须按照以下步骤操作。如果使用 Fuse Console Operator，它会为您处理此任务。



### 重要

您必须为每个 OpenShift 集群生成并签署单独的客户端证书。不要将同一证书用于多个集群。

### 先决条件

- 具有 集群管理员 对 OpenShift 集群的访问权限。
- 如果您要为多个 OpenShift 集群生成证书，并且之前在当前目录中为不同的集群生成证书，请执行以下操作之一，以确保为当前集群生成不同的证书：
  - 从当前目录中删除现有证书文件（如 ca.crt、ca.key、ca.key 和 ca.srl）。
  - 更改到其他工作目录。例如，如果您的当前工作目录命名为 cluster1，请创建一个新的 cluster2 目录，并将工作目录改为该目录：

```
mkdir ../cluster2
```

```
cd ../cluster2
```

### 流程

1. 以具有集群管理员访问权限的用户身份登录 OpenShift：

```
oc login -u <user_with_cluster_admin_role>
```

2. 通过执行以下命令来检索服务签名证书颁发机构密钥：

- 检索证书：



```
oc get secrets/signing-key -n openshift-service-ca -o "jsonpath={.data['tls.crt']}" | base64
--decode > ca.crt
```

- 检索私钥：

```
oc get secrets/signing-key -n openshift-service-ca -o "jsonpath={.data['tls.key']}" |
base64 --decode > ca.key
```

3.

使用 **easyrsa**、**openssl** 或 **cfssl** 生成客户端证书，如 [Kubernetes 证书管理](#) 中所述。

以下是使用 **openssl** 的示例命令：

a.

生成私钥：

```
openssl genrsa -out server.key 2048
```

b.

编写 **CSR** 配置文件。

```
cat <<EOT >> csr.conf
[ req ]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn

[ dn ]
CN = fuse-console.fuse.svc

[ v3_ext ]
authorityKeyIdentifier=keyid,issuer:always
keyUsage=keyEncipherment,dataEncipherment,digitalSignature
extendedKeyUsage=serverAuth,clientAuth
EOT
```

在这里，**CN** 参数中的值指的是应用程序使用的应用程序名称和命名空间。

c.

生成 **CSR**：

```
openssl req -new -key server.key -out server.csr -config csr.conf
```

d.

发布签名证书：

```
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt  
-days 10000 -extensions v3_ext -extfile csr.conf
```

### 后续步骤

您需要此证书来为 Fuse 控制台创建 secret，如 [使用命令行在 OpenShift 4.x 上安装和部署 Fuse 控制台](#) 中所述。

### 1.2.3. OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制

Fuse 控制台提供基于角色的访问控制(RBAC)，它根据 OpenShift 提供的用户授权来推断访问。在 Fuse 控制台中，RBAC 决定用户对 pod 执行 MBean 操作的能力。

如需有关 OpenShift 授权的信息，请参阅 [OpenShift 文档中的使用 RBAC 定义和应用权限](#) 部分。

当使用 Operator 在 OpenShift 上安装 Fuse 控制台时，默认禁用基于角色的访问控制。您可以选择在使用 Operator 期间或安装后启用 RBAC，如 [使用 OperatorHub 在 OpenShift 4.x 上安装和部署 Fuse 控制台](#) 中所述。

如果要为 Fuse 控制台实施基于角色的访问控制，则必须使用通过 RBAC 配置的模板之一(fuse-console-cluster-rbac.yml 或 fuse-console-namespace-rbac.yml)来安装 Fuse 控制台，如 [在 OpenShift 4.x 上安装 Fuse 控制台](#) 中所述。

Fuse Console RBAC 利用 OpenShift 中 pod 资源的操作动词访问权限来确定用户对 Fuse 控制台中 pod 的 MBean 操作的访问权限。默认情况下，Fuse 控制台有两个用户角色：

- **admin**

如果用户可以在 OpenShift 中更新 pod，则用户会延迟 Fuse 控制台的 admin 角色。用户可以在 Fuse 控制台中为 pod 执行写入 MBean 操作。

- **viewer**

如果用户可以在 OpenShift 中获取 pod，则用户会延迟 Fuse 控制台的 viewer 角色。用户可以在 Fuse 控制台中为 pod 执行只读 MBean 操作。



## 注意

如果您没有使用 Fuse Console Operator 配置启用 RBAC，或者您没有使用 RBAC 模板安装 Fuse 控制台，只有 pod 资源上赋予 update 动词的 OpenShift 用户才会被授权执行 Fuse Console MBeans 操作。授予 pod 资源的 get 动词的用户可以查看 pod，但不能执行任何 Fuse 控制台操作。

## 其他资源

- [确定 OpenShift 4.x 上 Fuse 控制台的访问角色](#)
- [在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问](#)
- [在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制](#)

### 1.2.3.1. 确定 OpenShift 4.x 上 Fuse 控制台的访问角色

Fuse Console 基于角色的访问控制是从 Pod 的 OpenShift 权限中推断出来的。要确定授予特定用户的 Fuse 控制台访问角色，获取赋予 Pod 用户的 OpenShift 权限。

## 先决条件

- 您知道用户名。
- 您知道 pod 的名称。

## 流程

- 要确定用户是否有 pod 的 Fuse Console admin 角色，请运行以下命令来查看用户是否可以更新 OpenShift 上的 pod：

```
oc auth can-i update pods/<pod> --as <user>
```

如果响应是 **yes**，则用户具有 Pod 的 Fuse Console admin 角色。用户可以在 Fuse 控制台中为 pod 执行写入 MBean 操作。

- 要确定用户是否有 pod 的 Fuse Console viewer 角色，请运行以下命令来查看用户是否可以在 OpenShift 上获取 pod：

```
oc auth can-i get pods/<pod> --as <user>
```

如果响应是 **yes**，用户具有 Pod 的 Fuse Console viewer 角色。用户可以在 Fuse 控制台中为 pod 执行只读 MBean 操作。根据上下文，Fuse 控制台可防止具有 viewer 角色的用户执行写入 MBean 操作，方法是禁用一个选项，或者在用户尝试写 MBean 操作时显示 "operation not allowed for this user" 消息。

如果响应不是，则用户没有绑定到任何 Fuse 控制台角色，用户无法在 Fuse 控制台中查看 pod。

#### 其他资源

- [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)
- [在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问](#)
- [在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制](#)

#### 1.2.3.2. 在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问

如果您使用 OperatorHub 安装 Fuse 控制台，并且启用了 [基于角色的访问控制\(RBAC\)](#)，[Fuse Console Operator](#) 会提供默认的 RBAC 行为，如 [OpenShift 4.x 上 Fuse Console](#) 的基于角色的访问控制中所述。如果要自定义 Fuse 控制台 RBAC 行为，在部署 Fuse 控制台前，您必须提供一个 ConfigMap 文件（定义自定义 RBAC 行为）。您必须将自定义 ConfigMap 文件放在安装 Fuse Console Operator 的同一命名空间中。

如果使用命令行模板安装 Fuse 控制台，则 `deployment-cluster-rbac.yml` 和 `deployment-namespace-rbac.yml` 模板会创建一个包含配置文件(ACL.yml)的 ConfigMap。配置文件定义 MBean 操作允许的角色。

#### 前提条件

- 您使用 OperatorHub 或使用其中一个 Fuse 控制台 RBAC 模板(`deployment-cluster-rbac.yml` 或 `deployment-namespace-rbac.yml`)安装 Fuse 控制台。

## 流程

### 自定义 Fuse 控制台 RBAC 角色：

1. 如果使用命令行安装 Fuse 控制台，则安装模板包含一个默认的 ConfigMap 文件，以便您可以跳至下一步。

如果使用 OperatorHub 安装 Fuse 控制台，在部署 Fuse 控制台前，要创建一个 RBAC ConfigMap：

- a. 确保当前的 OpenShift 项目是您要安装 Fuse 控制台的项目。例如，如果要在 fusetest 项目中安装 Fuse 控制台，请运行以下命令：

```
oc project fusetest
```

- b. 要从模板创建 Fuse Console RBAC ConfigMap 文件，请运行以下命令：

```
oc process -f https://raw.githubusercontent.com/jboss-fuse/application-templates/2.1.x.sb2.redhat-7-8-x/fuse-console-operator-rbac.yml -p APP_NAME=fuse-console | oc create -f -
```

2. 运行以下命令，在编辑器中打开 ConfigMap：

```
oc edit cm $APP_NAME-rbac
```

例如：

```
oc edit cm fuse-console-rbac
```

3. 编辑该文件。
4. 保存文件以使改变生效。OpenShift 会自动重启 Fuse Console pod。

### 其他资源

- [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)

- [确定 OpenShift 4.x 上 Fuse 控制台的访问角色](#)
- [在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制](#)

### 1.2.3.3. 在 OpenShift 4.x 上为 Fuse 控制台禁用基于角色的访问控制

Fuse Console 的 `HAWTIO_ONLINE_RBAC_ACL` 环境变量传递基于角色的访问控制(RBAC) ConfigMap 配置文件路径到 OpenShift 服务器。如果没有指定 `HAWTIO_ONLINE_RBAC_ACL` 环境变量，则只有为 pod 资源(OpenShift)授予 `update` 动词的用户才会被授权在 Fuse 控制台中对 pod 调用 MBeans 操作。

请注意，当使用 Operator 在 OpenShift 上安装 Fuse 控制台时，默认禁用基于角色的访问。

#### 前提条件

已安装 Fuse 控制台：

- 通过使用 OperatorHub，并在 Fuse 控制台配置中启用 RBAC
- 使用命令行，并且您指定了其中一个 Fuse 控制台 RBAC 模板(`deployment-cluster-rbac.yml` 或 `deployment-namespace-rbac.yml`)。

#### 流程

为 Fuse 控制台禁用基于角色的访问控制：

1. 在 OpenShift 中，编辑 Fuse 控制台的部署配置资源。
2. 删除整个 `HAWTIO_ONLINE_RBAC_ACL` 环境变量定义。  
  
(请注意，仅清除其值还不够)。
3. 保存文件以使改变生效。OpenShift 会自动重启 Fuse Console pod。

## 其他资源

- [OpenShift 4.x 上 Fuse 控制台的基于角色的访问控制](#)
- [确定 OpenShift 4.x 上 Fuse 控制台的访问角色](#)
- [在 OpenShift 4.x 上自定义对 Fuse 控制台进行基于角色的访问](#)

### 1.2.4. 在 OpenShift 4.x 上升级 Fuse 控制台

Red Hat OpenShift 4.x 处理 operator 的更新，包括 Red Hat Fuse operator。如需更多信息，请参阅 [Operators OpenShift 文档](#)。

然后，Operator 更新可以根据应用程序的配置方式触发应用程序升级。

对于 Fuse 控制台应用程序，您还可以通过编辑应用程序自定义资源定义的 `.spec.version` 字段来触发到应用程序的升级。

## 前提条件

- 有 OpenShift 集群 admin 权限。

## 流程

升级 Fuse 控制台应用程序：

1. 在终端窗口中，使用以下命令更改应用程序自定义资源定义的 `.spec.version` 字段：

```
oc patch <project-name> <custom-resource-name> --type='merge' -p '{"spec": {"version": "1.7.1"}}'
```

例如：

```
oc patch myproject example-fuseconsole --type='merge' -p '{"spec":{"version": "1.7.1"}}'
```

2.

检查应用程序的状态是否已更新：

```
oc get myproject
```

响应显示有关应用程序的信息，包括版本号：

NAME	AGE	URL	IMAGE
example-fuseconsole	1m	https://fuseconsole.192.168.64.38.nip.io	
		docker.io/fuseconsole/online:1.7.1	

当您更改 `.spec.version` 字段的值时，OpenShift 会自动重新部署应用。

3.

检查版本更改触发的重新部署状态：

```
oc rollout status deployment.v1.apps/example-fuseconsole
```

成功部署显示这个响应：

```
deployment "example-fuseconsole" successfully rolled out
```

### 1.3. 在 OPENSIFT 3.11 中设置 FUSE 控制台

在 OpenShift 3.11 中，您可以使用两种方式设置 Fuse 控制台：

- 通过将 **集中式 Fuse Console** 目录项添加到项目中，以便您可以监控项目中的所有运行 Fuse 容器。
- 从特定的 pod 中，以便您可以监控单个正在运行的 Fuse 容器。

您可以从 OpenShift 控制台或命令行部署 Fuse 控制台。



注意

要在基于 Minishift 或 CDK 的基础上安装 Fuse 控制台，请遵循以下 KCS 文章中所述的步骤。

- 要在基于 Minishift 或 CDK 上安装 Fuse 控制台，请参阅 [KCS 4998441](#)。
- 如果需要禁用 Jolokia 身份验证，请参阅 [KCS 3988671](#) 中描述的临时解决方案。

## 先决条件

- 在 OpenShift 镜像流上安装 Fuse，以及 Fuse 控制台的模板，如 [OpenShift 上的 Fuse 指南](#) 中所述。
- 对于 OpenShift 3.11 上的集群模式，您需要集群管理员角色和集群模式模板。运行以下命令：

```
oc adm policy add-cluster-role-to-user cluster-admin system:serviceaccount:openshift-infra:template-instance-controller
```

注意

- 默认情况下，集群模式模板仅在最新版本的 OpenShift Container Platform 上可用。它不是由 OpenShift Online 默认目录提供的。
- Fuse 控制台模板默认配置端到端加密，以便 Fuse 控制台请求是安全的端到端，从浏览器到集群内服务。
- Fuse 控制台的用户管理由 OpenShift 处理。
- 基于角色的访问控制（用于部署 Fuse 控制台的用户）尚不可用于 OpenShift 上的 Fuse。

## 第 1.3.1 节 “从 OpenShift 3.11 控制台部署 Fuse 控制台”

## 第 1.3.2 节 “从 OpenShift 3.11 上的 Fuse 控制台监控单个 Fuse pod”

## 第 1.3.3 节 “从命令行部署 Fuse 控制台”

### 1.3.1. 从 OpenShift 3.11 控制台部署 Fuse 控制台

要从 OpenShift 3.11 控制台在 OpenShift 集群上部署 Fuse 控制台，请按照以下步骤操作。

#### 流程

1. 在 OpenShift 控制台中，打开现有项目或创建新项目。

2. 将 Fuse 控制台添加到 OpenShift 项目中：

- a. 选择 **Add to Project** → **Browse Catalog**。

此时会打开 **Select an item to to the current project** 页。

- b. 在 **Search** 字段中，键入 **Fuse Console**。

**Red Hat Fuse 7.x Console** 和 **Red Hat Fuse 7.x Console (集群)** 项应该显示为搜索结果。



#### 注意

如果 **Red Hat Fuse Console** 项没有作为搜索结果显示，或者显示的项目不是最新版本，您可以手动安装 **Fuse Console** 模板，如 **OpenShift Guide** 中的 **Fuse "Prepare the OpenShift server"** 部分所述。

- a. 点 **Red Hat Fuse Console** 项之一：

- **Red Hat Fuse 7.x Console** - 此版本的 **Fuse** 控制台发现并连接到当前 **OpenShift** 项目中部署的 **Fuse** 应用程序。

- **Red Hat Fuse 7.x Console (集群) - 此版本的 Fuse 控制台可以发现并连接到 OpenShift 集群上多个项目中部署的 Fuse 应用程序。**
- b. **在 Red Hat Fuse Console 向导中，单击 Next。此时会打开向导的 Configuration 页面。**

另外，您可以更改配置参数的默认值。

  1. **点 Create。**

向导的 **Results** 页面表示已创建了 Red Hat Fuse 控制台。
  2. **单击 Continue to the project overview 链接，以验证 Fuse Console 应用是否已添加到项目中。**
  3. **要打开 Fuse Console，请单击提供的 URL 链接，然后登录。**

在列出所需权限的浏览器中打开 **Authorize Access** 页面。
  4. **点击 Allow selected permissions。**

**Fuse Console** 在浏览器中打开，并显示项目中运行的 **Fuse pod**。
  5. **点您要查看的应用程序的 Connect。**

这时将打开一个新的浏览器窗口，显示 **Fuse** 控制台中的应用程序。

### 1.3.2. 从 OpenShift 3.11 上的 Fuse 控制台监控单个 Fuse pod

您可以为在 OpenShift 3.11 上运行的 Fuse pod 打开 Fuse 控制台：

1. **在 OpenShift 项目中的 Applications → Pods 视图中，点 pod 名称查看正在运行的 Fuse pod 的详情。在这个页面的右侧，您会看到容器模板的概述信息：**

## Template

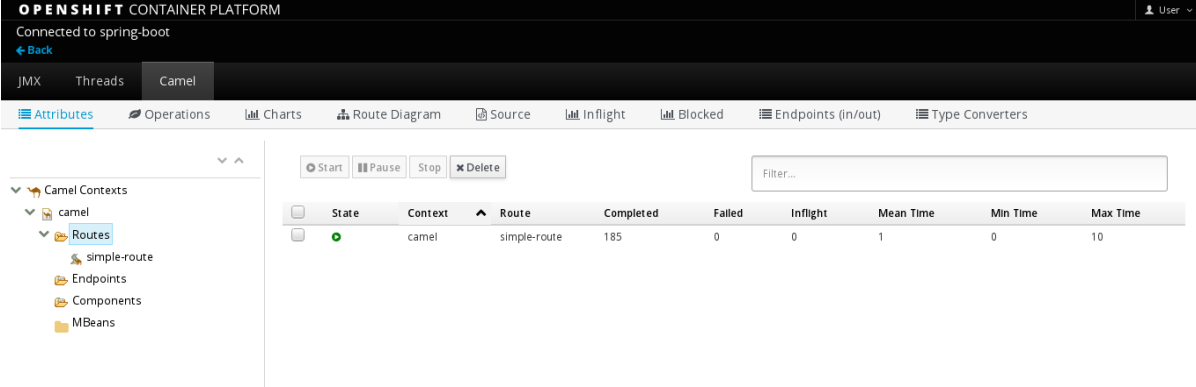
### Containers

CONTAINER: SPRING-BOOT

-  **Image:** [test/fuse70-spring-boot](#) eda527f 193.1 MiB
-  **Build:** [fuse70-spring-boot-s2i, #2](#)
-  **Source:** Binary
-  **Ports:** 8080/TCP (http), 8778/TCP (jolokia), 9779/TCP (prometheus)
-  **Mount:** default-token-p4zsn → /var/run/secrets/kubernetes.io/serviceaccount  
read-only
-  **CPU:** 200 millicores to 1 core
-  **Readiness Probe:** GET /health on port 8081 (HTTP) 10s delay, 1s timeout
-  **Liveness Probe:** GET /health on port 8081 (HTTP) 180s delay, 1s timeout
-  [Open Java Console](#)

2.

从此视图中，单击 **Open Java Console** 链接以打开 **Fuse 控制台**。




### 注意

要将 OpenShift 配置为在 pod 视图中显示到 **Fuse Console** 的链接，在 OpenShift 镜像上运行 **Fuse** 的 pod 必须声明设置为 **jolokia** 的 name 属性中的 TCP 端口：

```
{
  "kind": "Pod",
```

```
[...]
"spec": {
  "containers": [
    {
      [...]
      "ports": [
        {
          "name": "jolokia",
          "containerPort": 8778,
          "protocol": "TCP"
        }
      ]
    }
  ]
}
```

### 1.3.3. 从命令行部署 Fuse 控制台

**表 1.1 “Fuse 控制台模板”** 描述可用于从命令行部署 Fuse 控制台的 OpenShift 3.1 模板，具体取决于 Fuse 应用部署的类型。

表 1.1. Fuse 控制台模板

类型	描述
<b>fis-console-cluster-template.json</b>	Fuse 控制台可以发现并连接到在多个命名空间或项目中部署的 Fuse 应用程序。若要部署此模板，您必须具有 OpenShift cluster-admin 角色。
<b>fis-console-namespace-template.json</b>	此模板限制 Fuse 控制台对当前 OpenShift 项目（命名空间）的访问，因此充当单个租户部署。若要部署此模板，您必须具有当前 OpenShift 项目的 admin 角色。

另外，您可以通过运行这个命令来查看所有模板的参数列表：

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fis-console-namespace-template.json
```

### 流程

从命令行部署 Fuse 控制台：

1. 运行以下命令之一（其中 **myproject** 是项目的名称），基于 Fuse 控制台模板创建新应用：
  - 对于 **Fuse Console 集群模板**，其中 **myhost** 是访问 Fuse 控制台的主机名：

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fis-console-cluster-template.json -p ROUTE_HOSTNAME=myhost
```

- 对于 **Fuse Console 命名空间 模板** :

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-780019-redhat-00005/fis-console-namespace-template.json
```



#### 注意

您可以省略 命名空间 模板的 `route_hostname` 参数，因为 **OpenShift** 会自动生成一个。

2. 运行以下命令，获取 **Fuse Console** 部署的状态和 URL :

```
oc status
```

3. 要从浏览器访问 **Fuse 控制台**，请使用提供的 URL（例如 <https://fuse-console.192.168.64.12.nip.io>）。

## 1.4. 查看容器和应用程序

当您登录到 **OpenShift 的 Fuse 控制台**时，**Fuse 控制台**主页会显示可用的容器。

### 流程

- 若要管理（创建、编辑或删除）容器，请使用 **OpenShift 控制台**。
- 要查看 **OpenShift 集群上的 Fuse 应用程序和 AMQ Broker**（如果适用），请点 **Online** 选项卡。

## 1.5. 查看并管理 APACHE CAMEL 应用程序

在 **Fuse Console 的 Camel 选项卡**中，您可以查看和管理 **Apache Camel** 上下文、路由和依赖项。

您可以查看以下详情：

- 所有正在运行的 Camel 上下文的列表
- 每个 Camel 上下文的详细信息，如 Camel 版本号和运行时静态
- 每个 Camel 应用程序及其运行时统计中的所有路由列表
- 正在运行的路由的图形表示以及实时指标

您还可以通过以下方法与 Camel 应用程序交互：

- 启动和停止上下文
- 管理所有 Camel 应用程序及其路由的生命周期，以便您可以重新启动、停止、暂停、恢复等。
- 正在运行的路由的实时追踪和调试
- 浏览并发送消息到 Camel 端点

## 前提条件

只有在连接到使用一个或多个 Camel 路由的容器时，Camel 选项卡才可用。

### 1.5.1. 启动、暂停或删除上下文

1. 在 Camel 选项卡的树视图中，单击 **Camel Contexts**。
2. 选中列表中一个或多个上下文旁边的框。

3. 单击 **Start** 或 **Suspend**.
4. 删除上下文：
  - a. 停止上下文。
  - b. 单击省略图标，然后从下拉菜单中选择 **删除**。



#### 注意

删除上下文时，您可以将其从部署的应用中删除。

### 1.5.2. 查看 Camel 应用程序详情

1. 在 **Camel** 选项卡的树视图中，点 **Camel 应用程序**。
2. 要查看应用属性和值列表，请单击 **Attributes**。
3. 要查看应用程序属性的图形表示，请点 **Chart**，然后点 **Edit** 以选择您要在图表中看到的属性。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看应用端点，请单击 **Endpoints**。您可以根据 **URL**、**路由ID** 和 **方向** 过滤列表。
6. 要查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，该机制用于将消息正文和消息标头转换为不同类型的消息，请单击 **Type Converters**。
7. 要查看和执行 **JMX** 操作，如从 **XML** 添加或更新路由，或查找 **classpath** 中提供的所有 **Camel** 组件，请单击 **Operations**。



### 1.5.3. 查看 Camel 路由列表并与其交互

1.

查看路由列表：

a.

点 **Camel** 标签页。

b.

在 **tree** 视图中，点应用程序的 **routes** 文件夹：

## Routes

<input type="checkbox"/>	Name ^	State
<input type="checkbox"/>	_route1	Started
<input type="checkbox"/>	_route2	Started

2.

启动、停止或删除一个或多个路由：

a.

选中列表中一个或多个路由旁边的框。

b.

点 **Start** 或 **Stop**。

c.

要删除路由，您必须首先停止它。然后单击省略图标，然后从下拉菜单中选择 **删除**。

## Routes

<input checked="" type="checkbox"/>	Name ^	State
<input checked="" type="checkbox"/>	_route1	Started
<input checked="" type="checkbox"/>	_route2	Started

Start Stop ⋮

Delete



### 注意

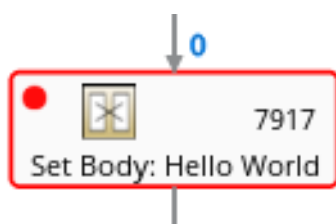
- 删除路由时，您可以将其从部署的应用程序中删除。
- 您也可以在树形视图中选择一个特定的路由，然后单击右上角的菜单来启动、停止或删除。

3. 要查看路由的图形图表，请点 **Route** 图表。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看端点，请单击 **Endpoints**。您可以根据 **URL**、**Route ID** 和 **direction** 来过滤列表。
6. 点 **Type Converters** 查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，用于将消息正文和消息标头转换为不同的类型。
7. 与特定路由交互：
  - a. 在 **Camel** 选项卡的树视图中，选择一个路由。
  - b. 要查看路由属性和值列表，请单击 **Attributes**。
  - c. 要查看路由属性的图形表示，请点 **Chart**。您可以点 **Edit** 来选择要在图表中看到的属性。
  - d. 要查看动态和阻止的交换，请点 **Exchanges**。
  - e. 单击 **Operations** 以在路由上查看并执行 **JMX** 操作，如以 **XML** 转储路由或获取路由的 **Camel ID** 值。
8. 通过路由跟踪信息：

- a. 在 Camel 选项卡的树视图中，选择一个路由。
  - b. 选择 Trace，然后单击 Start tracing。
9. 发送消息到路由：
- a. 在 Camel 选项卡的树视图中，打开上下文的端点文件夹，然后选择端点。
  - b. 点 Send 子选项卡。
  - c. 以 JSON 或 XML 格式配置消息。
  - d. 单击 Send。
  - e. 返回到路由的 Trace 选项卡，以查看通过路由的消息流。

#### 1.5.4. 调试路由

1. 在 Camel 选项卡的树视图中，选择一个路由。
2. 选择 Debug，然后单击 Start debugging。
3. 要添加断点，请在图表中选择节点，然后单击 Add breakpoint。节点上会出现一个红色点：



该节点被添加到断点列表中：

Breakpoints	
setBody1	✕
log1	✕

4. 单击向下箭头，以转至下一个节点或 **Play** 按钮，以恢复运行路由。
5. 单击 **暂停** 按钮，以暂停路由的所有线程。
6. 完成后，点 **Stop debug**。所有断点都会被清除。

## 1.6. 查看 AMQ BROKER

您可以配置 Fuse 控制台，以查看 OpenShift 集群中部署的所有 AMQ 代理。



### 重要

从 Fuse 控制台查看 AMQ 代理只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的更多信息，请参阅

<https://access.redhat.com/support/offerings/techpreview>。

### 先决条件

每个 AMQ 代理镜像（您要在 Fuse 控制台中查看）都必须：

- 安装在 Fuse 控制台所在的同一 OpenShift 集群上。
- 配置 以便 Fuse 控制台可以识别并连接它，如 AMQ Broker 文档中的 [Fuse 控制台中启用 Artemis 插件](#) 部分所述。

## 流程

- 单击 **Artemis** 以查看 AMQ 管理控制台，并监控 AMQ Broker 的状态。(AMQ Broker 基于 [Apache ActiveMQ Artemis](#)。)

有关使用 AMQ 管理控制台的详情，请参考 [管理 AMQ Broker 指南中的第 2 章“使用 AMQ 管理控制台”](#)。

## 1.7. 查看并管理 JMX 域和 MBEANS

Java 管理扩展(JMX)是一种 Java 技术，可让您在运行时动态管理资源（服务、设备和应用程序）。资源由名为 MBeans（受管 Bean）的对象表示。您可以在创建、实施或安装后立即管理和监控资源。

在 Fuse 控制台中使用 JMX 插件，您可以查看和管理 JMX 域和 MBeans。您可以查看 MBean 属性、运行命令并创建显示 MBeans 统计信息的图表。

JMX 选项卡提供活动 JMX 域和在文件夹中组织的 MBeans 的树视图。您可以查看 MBeans 的详情并执行命令。

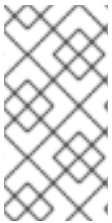
## 流程

1. 查看并编辑 MBean 属性：
  - a. 在 tree 视图中，选择一个 MBean。
  - b. 单击 **Attributes** 选项卡。
  - c. 点属性查看其详情。
2. 执行操作：
  - a. 在 tree 视图中，选择一个 MBean。

- b. 单击 **Operations** 选项卡，展开列出的操作之一。
  - c. 点 **Execute** 来运行操作。
3. 查看 chart :
- a. 在树形视图中，选择一个项目。
  - b. 单击 **Chart** 选项卡。

## 1.8. 查看诊断

使用 **diagnostics s** 选项卡，通过 **JVM diagnosticsCommand** 和 **HotspotDiangostic** 接口查看有关 **JVM** 的诊断信息。



### 注意

该功能与 **Java Mission Control (jmc)**或命令行工具 **jcmm** 中的 **diagnose Commands** 视图类似。在某些情况下，插件将提供对应的 **jcmm** 命令。

## 流程

1. 要检索载入类的实例数量以及它们所消耗的字节数，请单击 **Class Histogram**。如果操作重复，标签会显示自上次运行以来的区别。
2. 要查看 **JVM** 诊断标志设置，请单击 **JVM** 标志。
3. 对于正在运行的 **JVM**，您还可以修改标志设置。

## 其他资源

支持的 **JVM** 依赖于平台，如需更多信息，请访问以下源之一：

- <http://www.oracle.com/technetwork/java/vmoptions-jsp-140102.html>
- <http://openjdk.java.net/groups/hotspot/docs/RuntimeOverview.html>

## 1.9. 查看线程

您可以查看和监控线程的状态。

### 流程

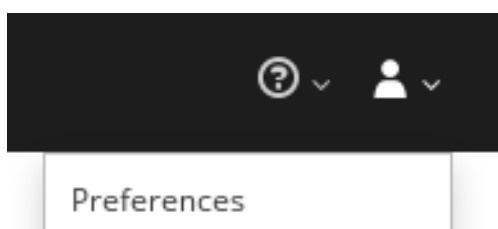
1. 单击 **Runtime** 选项卡，然后单击 **线程** 子选项卡。**Threads** 页面列出了每个线程的活跃线程和堆栈追踪详情。默认情况下，线程列表以降序显示所有线程。
2. 要通过增加 ID 来对列表进行排序，请单击 ID 列标签。
3. (可选) 根据线程状态 (如 **Blocked**) 或线程名称过滤列表。
4. 要深入查看特定线程的详细信息，如锁定类名称和该线程的完整堆栈追踪，请在 **Actions** 列中点 **More**。

## 1.10. 确保数据在 FUSE 控制台中正确显示

如果在 **Fuse** 控制台中显示队列和连接缺少队列、缺少连接或显示不一致图标，请调整 **Jolokia** 集合大小参数，以指定 **Jolokia marshals** 中的最大元素数。

### 流程

1. 在 **Fuse** 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 增加 **Maximum collection size** 选项的值（默认为 **50,000**）。
3. 单击 **Close**。



## 第 2 章 在 SPRING BOOT 独立上监控和管理 RED HAT FUSE 应用程序

### 2.1. 关于 FUSE 控制台

红帽 Fuse 控制台是一个基于 Hawtio 开源软件的 Web 控制台。如需支持的浏览器列表，请访问 [支持的配置](#)。

Fuse 控制台提供了一个中央接口，用于检查和管理一个或多个部署 Fuse 容器的详细信息。您还可以监控红帽 Fuse 和系统资源，执行更新，以及启动或停止服务。

安装 Red Hat Fuse 独立或使用 OpenShift 中的 Fuse 时，可以使用 Fuse 控制台。您可以在 Fuse 控制台中查看和管理的集成取决于正在运行的插件。可能的插件包括：

- Camel
- JMX
- OSGI
- Runtime
- Logs

### 2.2. 访问 SPRING BOOT 2.X 的 FUSE 控制台

您可以访问独立 Fuse Spring Boot 2.x 发行版本的 Fuse 控制台。

#### 流程

1. 将以下依赖项添加到 Fuse 应用程序的 pom.xml 文件中：

```
<dependency>  
  <groupId>io.hawt</groupId>  
  <artifactId>hawtio-springboot</artifactId>
```

```
</dependency>
```

请注意，您不需要指定确切的版本，因为它由 Maven BOM 提供。

2.

编辑 `src/main/resources/application.properties` 文件：

a.

设置以下属性：

- `management.endpoints.web.exposure.include=hawtio,jolokia`
- `hawtio.authenticationEnabled=false`
- `management.endpoint.hawtio.enabled=true`
- `management.endpoint.jolokia.enabled=true`

b.

另外，还可设置 `management.endpoints.web.base-path` 属性。

默认情况下，对于 Spring Boot 2.x，Fuse 控制台的 URL 包括管理端点的上下文路径 (`/actuator`)。例如：

<http://localhost:10001/actuator/hawtio/index.html>

要更改此默认 URL，例如指定 <http://localhost:10001/hawtio>，设置 `management.endpoints.web.base-path` 属性，如下所示：

```
management.endpoints.web.base-path=/
```

`application.properties` 设置应类似以下示例：

```
# ports
```

```

server.port=8080

management.server.port=10001

# enable management endpoints for healthchecks and hawtio

management.endpoints.enabled-by-default = false

management.endpoint.hawtio.enabled = true

management.endpoint.jolokia.enabled = true

management.endpoints.health.enabled = true

management.health.defaults.enabled=false

camel.health.enabled=false

camel.health.indicator.enabled=true

management.endpoints.web.exposure.include=hawtio,jolokia

hawtio.authenticationEnabled=false

# change the URL so that it does not include the actuator folder

management.endpoints.web.base-path=/

```



### 注意

默认情况下，在 Spring Boot 上 Fuse Console 的身份验证被禁用。另外，您可以通过编写特定于 Fuse 控制台的代码来启用身份验证。以下是可用于指导的示例：

<https://github.com/hawtio/hawtio/tree/master/examples/springboot-authentication>

3. 运行 Fuse 应用程序：

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

4. 要确定 Fuse Console URL 的端口号，请通过查看 `src/main/resources/application.properties` 文件中设置的值来获取 `management.server.port` 值。例如：

```
management.server.port = 10001
```

5.

要在浏览器中打开 Fuse 控制台，请使用以下 URL 语法，其中 `nnnnn` 是 `management.server.port` 属性的值：

<http://localhost:nnnnn/actuator/hawtio>

例如，如果 `management.server.port` 属性值为 `10001`，且您尚未设置 `management.endpoints.web.base-path` 属性，则 URL 为：

<http://localhost:10001/actuator/hawtio/index.html>

### 2.3. 自定义 FUSE 控制台品牌

您可以通过在 Spring Boot 独立应用程序的 Fuse 中添加 `hawtconfig.json` 文件来自定义 Fuse 控制台品牌信息，如标题、徽标和登录页面信息。

#### 流程

1.

在 Spring Boot standalone application 的 `src/main/webapp` 目录中的本地 Fuse 中创建一个名为 `hawtconfig.json` 的 JSON 文件。

2.

在您选择的编辑器中打开 `src/main/webapp/hawtconfig.json`，然后添加以下内容：

```
{
  "branding": {
    "appName": "Red Hat Fuse Console",
    "appLogoUrl": "img/Logo-Red_Hat-Fuse-A-Reverse-RGB.png",
    "companyLogoUrl": "img/Logo-RedHat-A-Reverse-RGB.png"
  },
  "login": {
    "description": "",
    "links": []
  },
  "about": {
    "title": "Red Hat Fuse Console",
    "productInfo": [],
    "additionalInfo": "",
    "copyright": "",
    "imgSrc": "img/Logo-RedHat-A-Reverse-RGB.png"
  },
  "disabledRoutes": [
```

```

"/camel/source",
"/diagnostics",
"/jvm/discover",
"/jvm/local"
]
}

```

3. 更改表 A.1 “Fuse 控制台配置属性”中列出的配置属性的值。

4. 保存您的更改。

5. 使用以下命令在 Spring Boot 上运行 Fuse :

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

6. 在 Web 浏览器中，使用此 URL 打开 Fuse 控制台：  
<http://localhost:10001/actuator/hawtio/index.html>



#### 注意

如果您已在网页浏览器中运行 Fuse 控制台，品牌会存储在浏览器的本地存储中。要使用新的品牌设置，您必须清除浏览器的本地存储。

## 2.4. 保护 FUSE 控制台

在 Spring Boot 上保护 Fuse 控制台：

- 在部署到 AWS 时禁用 Fuse 控制台的代理 servlet

如果要独立部署 Fuse 应用程序到 Amazon Web Services (AWS)，您应该通过将 `hawtio.disableProxy` 系统属性设置为 `true` 来禁用 Fuse 控制台的代理 servlet。



#### 注意

当您禁用 Fuse 控制台代理 servlet 时，Fuse Console 的 Connect 选项卡被禁用，您无法从 Fuse 控制台连接到其他 JVM。如果要在 AWS 上部署多个 Fuse 应用程序，您必须为每个应用程序部署 Fuse 控制台。

- 将 HTTPS 设置为所需的协议

您可以使用 `hawtio.http.strictTransportSecurity` 属性来要求 Web 浏览器使用安全 HTTPS 协议来访问 Fuse 控制台。此属性指定尝试使用 HTTP 访问 Fuse 控制台的 Web 浏览器必须自动转换请求以使用 HTTPS。

- 使用公钥来保护响应

您可以使用 `hawtio.http.publicKeyPins` 属性来保护 HTTPS 协议，方法是告诉 Web 浏览器将特定的加密密钥与 Fuse 控制台相关联，以降低“man-in-the-middle”攻击的风险。

## 流程

1. 设置 `hawtio.http.strictTransportSecurity` 和 `hawtio.http.publicKeyPins` 属性，如下例所示：

```
public static void main(String[] args) {
    System.setProperty("hawtio.http.strictTransportSecurity", "max-age=31536000;
includeSubDomains; preload");
    System.setProperty("hawtio.http.publicKeyPins", "pin-
sha256=cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs"; max-age=5184000;
includeSubDomains");
    SpringApplication.run(YourSpringBootApplication.class, args);
}
```

2. （用于仅在 AWS 上部署）要禁用 Fuse 控制台的代理 servlet，请设置 `hawtio.disableProxy` 属性，如下例所示：

```
public static void main(String[] args) {
    System.setProperty("hawtio.disableProxy", "true");
}
```

## 其他资源

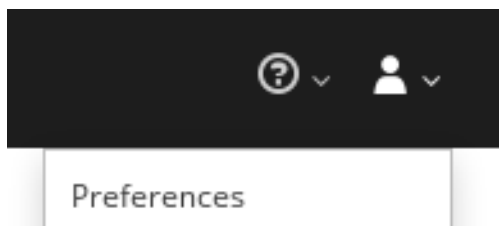
- 有关 `hawtio.http.strictTransportSecurity` 属性的语法的描述，请参阅 [HTTP Strict Transport Security \(HSTS\) 响应标头的描述页面](#)。
- 有关 `hawtio.http.publicKeyPins` 属性的语法的描述，包括如何提取 Base64 编码的公钥的说明，请参阅 [HTTP Public Key Pinning 响应标头的描述页面](#)。

## 2.5. 确保数据在 FUSE 控制台中正确显示

如果在 Fuse 控制台中显示队列和连接缺少队列、缺少连接或显示不一致图标，请调整 Jolokia 集合大小参数，以指定 Jolokia marshals 中的最大元素数。

### 流程

1. 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 增加 **Maximum collection size** 选项的值（默认为 50,000）。
3. 单击 **Close**。

## 2.6. 连接到远程 FUSE 应用程序

Fuse 控制台使用 Jolokia，这是一种基于代理的 Java 管理扩展(JMX)方法，需要在客户端上安装额外的软件（代理）。默认情况下，Red Hat Fuse 包含一个 jolokia 代理。

使用独立 Fuse 控制台发布，您可以连接到已在它们中运行的 jolokia 代理(<https://jolokia.org/>)的远程集成。如果要连接的进程没有 jolokia 代理，请参阅 jolokia 文档(<http://jolokia.org/agent.html>)。

### 流程

Fuse 控制台的代理 servlet 使用白名单主机保护，默认情况下 Fuse 控制台只能连接到 localhost。如果要将 Fuse 控制台连接到其他远程 Fuse 实例，您需要在 Spring Boot 应用程序的 main () 方法中配置 hawtio.proxyWhitelist 系统属性：

```
System.setProperty("hawtio.proxyWhitelist", "localhost, 127.0.0.1, myhost1, myhost2, myhost3");
```

### 2.6.1. 连接到远程 Jolokia 代理

开始之前，您需要知道远程 Jolokia 代理的连接详情（主机名、端口和路径）。

Spring Boot 上的 Jolokia 代理的默认连接 URL 是 <http://<host>:8080/jolokia>

作为系统管理员，您可以更改此默认值。

通常，远程连接到 Jolokia 代理的 URL 是打开 Fuse 控制台加上 /jolokia 的 URL。例如，如果打开 Fuse 控制台的 URL 是 <http://<host>:1234/hawtio>，则远程连接的 URL 可能是 <http://<host>:1234/hawtio/jolokia>。

要连接到远程 Jolokia 实例，以便您可以检查其 JVM：

1. 点 **Connect** 标签页。
2. 点 **Remote** 选项卡，然后点 **Add connection**。
3. 键入 **Name**、**Scheme** (HTTP 或 HTTPS) 以及 **主机名**。
4. 单击 **Test Connection**。
5. 单击 **Add**。



#### 注意

Fuse 控制台自动探测 localhost 和 127.0.0.1 以外的本地网络接口，并将它们添加到白名单中。因此，您不需要手动将本地计算机的地址注册到白名单中。

### 2.6.2. 设置数据移动首选项

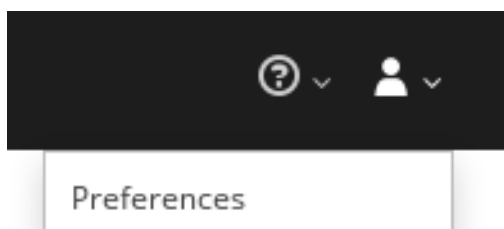
您可以更改以下 Jolokia 首选项，例如，如果要更频繁地刷新 Fuse 控制台中显示的数据。请注意，增加数据更新的频率会影响网络流量，并增加向服务器发出的请求数量。



- **更新速率** - 轮询到 Jolokia 以获取 JMX 数据之间的周期（默认为 5 秒）。
- **最大深度** - Jolokia 将在返回（默认为 7）前将对象聚合到服务器端的 JSON 级别数。
- **最大集合大小** - Jolokia marshals 在响应中的最大元素数（默认值为 50,000）。

更改这些设置的值：

1. 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 编辑选项，然后单击 **Close**。

### 2.6.3. 查看 JVM 运行时信息

要查看 JVM 运行时信息，如系统属性、指标和线程，请单击 **Runtime** 选项卡。

### 2.7. 查看并管理 APACHE CAMEL 应用程序

在 Fuse Console 的 Camel 选项卡中，您可以查看和管理 Apache Camel 上下文、路由和依赖项。

您可以查看以下详情：

- 所有正在运行的 Camel 上下文的列表

- 每个 Camel 上下文的详细信息，如 Camel 版本号和运行时静态
- 每个 Camel 应用程序及其运行时统计中的所有路由列表
- 正在运行的路由的图形表示以及实时指标

您还可以通过以下方法与 Camel 应用程序交互：

- 启动和停止上下文
- 管理所有 Camel 应用程序及其路由的生命周期，以便您可以重新启动、停止、暂停、恢复等。
- 正在运行的路由的实时追踪和调试
- 浏览并发送消息到 Camel 端点

### 前提条件

只有在连接到使用一个或多个 Camel 路由的容器时，Camel 选项卡才可用。

#### 2.7.1. 启动、暂停或删除上下文

1. 在 Camel 选项卡的树视图中，单击 **Camel Contexts**。
2. 选中列表中一个或多个上下文旁边的框。
3. 单击 **Start** 或 **Suspend**。
4. 删除上下文：

- a. 停止上下文。
- b. 单击省略图标，然后从下拉菜单中选择 **删除**。



#### 注意

删除上下文时，您可以将其从部署的应用中删除。

### 2.7.2. 查看 Camel 应用程序详情

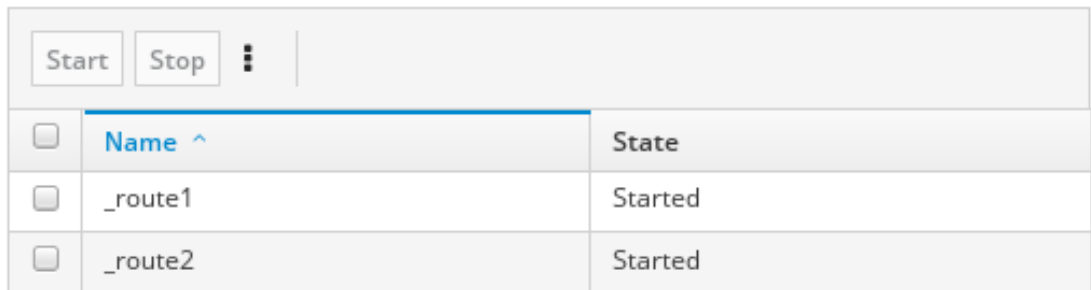
1. 在 **Camel** 选项卡的树视图中，点 **Camel 应用程序**。
2. 要查看应用属性和值列表，请单击 **Attributes**。
3. 要查看应用程序属性的图形表示，请点 **Chart**，然后点 **Edit** 以选择您要在图表中看到的属性。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看应用端点，请单击 **Endpoints**。您可以根据 **URL**、**路由ID** 和 **方向** 过滤列表。
6. 要查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，该机制用于将消息正文和消息标头转换为不同类型的消息，请单击 **Type Converters**。
7. 要查看和执行 **JMX** 操作，如从 **XML** 添加或更新路由，或查找 **classpath** 中提供的所有 **Camel** 组件，请单击 **Operations**。

### 2.7.3. 查看 Camel 路由列表并与之交互

1. 查看路由列表：

- a. 点 **Camel** 标签页。
- b. 在 **tree** 视图中，点应用程序的 **routes** 文件夹：

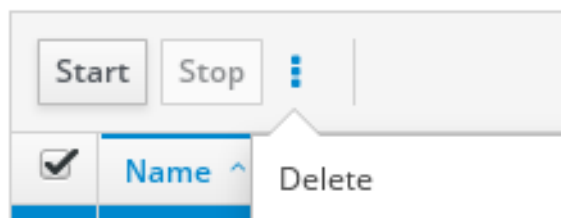
## Routes



<input type="checkbox"/>	Name ^	State
<input type="checkbox"/>	_route1	Started
<input type="checkbox"/>	_route2	Started

2. 启动、停止或删除一个或多个路由：
  - a. 选中列表中一个或多个路由旁边的框。
  - b. 点 **Start** 或 **Stop**。
  - c. 要删除路由，您必须首先停止它。然后单击省略图标，然后从下拉菜单中选择 **删除**。

## Routes





### 注意

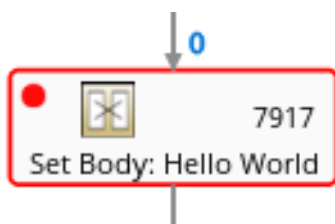
- 删除路由时，您可以将其从部署的应用程序中删除。
- 您也可以在树形视图中选择一个特定的路由，然后单击右上角的菜单来启动、停止或删除。

3. 要查看路由的图形图表，请点 **Route** 图表。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看端点，请单击 **Endpoints**。您可以根据 **URL**、**Route ID** 和 **direction** 来过滤列表。
6. 点 **Type Converters** 查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，用于将消息正文和消息标头转换为不同的类型。
7. 与特定路由交互：
  - a. 在 **Camel** 选项卡的树视图中，选择一个路由。
  - b. 要查看路由属性和值列表，请单击 **Attributes**。
  - c. 要查看路由属性的图形表示，请点 **Chart**。您可以点 **Edit** 来选择要在图表中看到的属性。
  - d. 要查看动态和阻止的交换，请点 **Exchanges**。
  - e. 单击 **Operations** 以在路由上查看并执行 **JMX** 操作，如以 **XML** 转储路由或获取路由的 **Camel ID** 值。
8. 通过路由跟踪信息：

- a. 在 **Camel** 选项卡的树视图中，选择一个路由。
  - b. 选择 **Trace**，然后单击 **Start tracing**。
9. 发送消息到路由：
- a. 在 **Camel** 选项卡的树视图中，打开上下文的端点文件夹，然后选择端点。
  - b. 点 **Send** 子选项卡。
  - c. 以 **JSON** 或 **XML** 格式配置消息。
  - d. 单击 **Send**。
  - e. 返回到路由的 **Trace** 选项卡，以查看通过路由的消息流。

#### 2.7.4. 调试路由

1. 在 **Camel** 选项卡的树视图中，选择一个路由。
2. 选择 **Debug**，然后单击 **Start debugging**。
3. 要添加断点，请在图表中选择节点，然后单击 **Add breakpoint**。节点上会出现一个红色点：



该节点被添加到断点列表中：

Breakpoints	
setBody1	×
log1	×

4. 单击向下箭头，以转至下一个节点或 **Play** 按钮，以恢复运行路由。
5. 单击 **暂停** 按钮，以暂停路由的所有线程。
6. 完成后，点 **Stop debug**。所有断点都会被清除。

## 2.8. 查看并管理 JMX 域和 MBEANS

**Java 管理扩展(JMX)**是一种 Java 技术，可让您在运行时动态管理资源（服务、设备和应用程序）。资源由名为 **MBeans**（受管 Bean）的对象表示。您可以在创建、实施或安装后立即管理和监控资源。

在 **Fuse** 控制台使用 **JMX** 插件，您可以查看和管理 **JMX** 域和 **MBeans**。您可以查看 **MBean** 属性、运行命令并创建显示 **MBeans** 统计信息的图表。

**JMX** 选项卡提供活动 **JMX** 域和在文件夹中组织的 **MBeans** 的树视图。您可以查看 **MBeans** 的详情并执行命令。

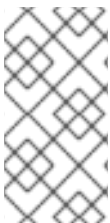
### 流程

1. 查看并编辑 **MBean** 属性：
  - a. 在 **tree** 视图中，选择一个 **MBean**。
  - b. 单击 **Attributes** 选项卡。

- c. 点属性查看其详情。
2. 执行操作：
    - a. 在 **tree** 视图中，选择一个 **MBean**。
    - b. 单击 **Operations** 选项卡，展开列出的操作之一。
    - c. 点 **Execute** 来运行操作。
  3. 查看 **chart**：
    - a. 在树形视图中，选择一个项目。
    - b. 单击 **Chart** 选项卡。

## 2.9. 查看诊断

使用 **diagnostics s** 选项卡，通过 **JVM diagnosticsCommand** 和 **HotspotDiangostic** 接口查看有关 **JVM** 的诊断信息。



### 注意

该功能与 **Java Mission Control (jmc)**或命令行工具 **jcmd** 中的 **diagnose Commands** 视图类似。在某些情况下，插件将提供对应的 **jcmd** 命令。

### 流程

1. 要检索载入类的实例数量以及它们所消耗的字节数，请单击 **Class Histogram**。如果操作重复，标签会显示自上次运行以来的区别。
2. 要查看 **JVM** 诊断标志设置，请单击 **JVM** 标志。



3. 对于正在运行的 JVM，您还可以修改标志设置。

## 其他资源

支持的 JVM 依赖于平台，如需更多信息，请访问以下源之一：

- <http://www.oracle.com/technetwork/java/vmoptions-jsp-140102.html>
- <http://openjdk.java.net/groups/hotspot/docs/RuntimeOverview.html>

## 2.10. 查看线程

您可以查看和监控线程的状态。

### 流程

1. 单击 **Runtime** 选项卡，然后单击 **线程** 子选项卡。**Threads** 页面列出了每个线程的活跃线程和堆栈追踪详情。默认情况下，线程列表以降序显示所有线程。
2. 要通过增加 ID 来对列表进行排序，请单击 ID 列标签。
3. (可选) 根据线程状态 (如 **Blocked**) 或线程名称过滤列表。
4. 要深入查看特定线程的详细信息，如锁定类名称和该线程的完整堆栈追踪，请在 **Actions** 列中点 **More**。

## 第 3 章 在 KARAF 独立上监控和管理红帽 FUSE 应用程序

### 3.1. 关于 FUSE 控制台

红帽 Fuse 控制台是一个基于 Hawtio 开源软件的 Web 控制台。如需支持的浏览器列表，请访问 [支持的配置](#)。

Fuse 控制台提供了一个中央接口，用于检查和管理一个或多个部署 Fuse 容器的详细信息。您还可以监控红帽 Fuse 和系统资源，执行更新，以及启动或停止服务。

安装 Red Hat Fuse 独立或使用 OpenShift 中的 Fuse 时，可以使用 Fuse 控制台。您可以在 Fuse 控制台中查看和管理的集成取决于正在运行的插件。可能的插件包括：

- Camel
- JMX
- OSGI
- Runtime
- Logs

### 3.2. 访问 FUSE 控制台

要访问 Apache Karaf 独立的 Fuse 控制台，请按照以下步骤操作：

#### 前提条件

在 Karaf 容器上安装 Fuse。有关逐步说明，请参阅 [在 Apache Karaf 上安装](#)。

#### 流程

1. 在命令行中，进入安装 Red Hat Fuse 的目录，并运行以下命令来启动 Fuse 独立：

```
./bin/fuse
```

Karaf 控制台启动并显示版本信息、默认 Fuse 控制台 URL 和常用命令的列表。

2. 在浏览器中，键入用于连接 Fuse 控制台的 URL。例如：<http://localhost:8181/hawtio>
3. 在登录页面中，输入您的用户名和密码，然后单击 Log In。

默认情况下，Fuse Console 显示 Home 页面。左侧导航选项卡表示正在运行的插件。

### 3.3. 保护 FUSE 控制台

在 Apache Karaf 上保护 Fuse 控制台：

- 在部署到 AWS 时禁用 Fuse 控制台的代理 servlet

如果要独立部署 Fuse 应用程序到 Amazon Web Services (AWS)，您应该通过将 `hawtio.disableProxy` 系统属性设置为 `true` 来禁用 Fuse 控制台的代理 servlet。



#### 注意

当您禁用 Fuse 控制台代理 servlet 时，Fuse Console 的 Connect 选项卡被禁用，您无法从 Fuse 控制台连接到其他 JVM。如果要在 AWS 上部署多个 Fuse 应用程序，您必须为每个应用程序部署 Fuse 控制台。

- 将 HTTPS 设置为所需的协议

您可以使用 `hawtio.http.strictTransportSecurity` 属性来要求 Web 浏览器使用安全 HTTPS 协议来访问 Fuse 控制台。此属性指定尝试使用 HTTP 访问 Fuse 控制台的 Web 浏览器必须自动转换请求以使用 HTTPS。

- 使用公钥来保护响应

您可以使用 `hawtio.http.publicKeyPins` 属性来保护 HTTPS 协议，方法是告诉 Web 浏览器将特定的加密密钥与 Fuse 控制台相关联，以降低“man-in-the-middle”攻击的风险。

- 启用 SSL/TLS 安全性

Fuse 控制台不默认启用 SSL/TLS 安全性。建议您在 Fuse 控制台中启用 SSL/TLS 安全性，以保护用户名/密码凭证不受 `snooping` 的影响。

- 实施红帽单点登录

- 控制用户访问

经过身份验证的用户可以执行的操作取决于分配给该用户的角色（或角色），如表 3.1 “在 Karaf 独立上进行基于角色的访问”中列出的。

## 流程

1. 要将 HTTPS 设置为所需的协议，请在 `$KARAF_HOME/etc/system.properties` 文件中设置 `hawtio.http.strictTransportSecurity` 属性，如下例所示：

```
hawtio.http.strictTransportSecurity = max-age=31536000; includeSubDomains; preload
```

2. 要使用公钥来保护响应，请在 `$KARAF_HOME/etc/system.properties` 文件中设置 `hawtio.http.publicKeyPins` 属性，如下例所示：

```
hawtio.http.publicKeyPins = pin-sha256="cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs"; max-age=5184000; includeSubDomains
```

3. （为了仅在 AWS 上部署）要禁用 Fuse 控制台的代理 servlet，请在 `$KARAF_HOME/etc/system.properties` 文件中将 `hawtio.disableProxy` 属性设置为 `true`，如下例所示：

```
hawtio.disableProxy = true;
```

- 4.

有关如何启用 SSL/TLS 安全性的详细信息，请参阅 [Apache Karaf 安全指南](#) 中的“启用 Undertow 的 SSL/TLS”一节。

5. 有关如何使用红帽单点登录保护 Fuse 控制台的信息，请参阅 [Red Hat Single Sign-on Securing Applications and Services Guide](#) 中的有关保护 Hawtio 管理控制台的章节。
6. 为确保用户具有必要的用户角色授权，可以执行用户需要执行的 Fuse 控制台操作，请按照以下步骤操作来设置用户角色：
  - a. 在编辑器中打开 Red Hat Fuse etc/users.properties 文件。
  - b. 为用户名、密码和角色添加一个条目。

例如，etc/users.properties 文件中的以下条目定义 admin 用户并授予 admin 角色。

```
admin = secretpass,admin
```

- c. 保存该文件。

#### 其他资源

- 有关 `hawtio.http.strictTransportSecurity` 属性的语法的描述，请参阅 [HTTP Strict Transport Security \(HSTS\) 响应标头的描述](#) 页面。
- 有关 `hawtio.http.publicKeyPins` 属性的语法的描述，包括如何提取 Base64 编码的公钥的说明，请参阅 [HTTP Public Key Pinning 响应标头的描述](#) 页面。

### 3.4. 基于角色的访问控制参考

经过身份验证的用户可以执行的操作取决于分配给该用户的角色（或角色），如 [表 3.1 “在 Karaf 独立上进行基于角色的访问”](#) 中列出的。

表 3.1. 在 Karaf 独立上进行基于角色的访问

操作	admin	Manager	viewer
登录/注销	Y	Y	Y
查看帮助主题	Y	Y	Y
设置用户首选项	Y	Y	Y
<b>connect</b>			
发现并连接到远程集成	Y	Y	Y
发现并连接到本地集成	Y	Y	Y
<b>Camel</b>			
查看所有正在运行的 Camel 应用程序	Y	Y	Y
启动、暂停、恢复和删除 Camel 上下文	Y	Y	
发送消息	Y	Y	
添加端点	Y	Y	
查看路由、路由图和运行时统计信息	Y	Y	Y
启动和停止路由	Y	Y	
删除路由	Y	Y	
<b>JMX</b>			
更改属性值	Y	Y	
在基于时间的图表中选择并查看属性	Y	Y	Y
查看操作	Y	Y	Y
<b>OSGI</b>			
查看捆绑包、功能、软件包、服务、服务器、框架和配置	Y	Y	Y

操作	admin	Manager	viewer
添加和删除捆绑包	Y	Y	
添加配置	Y	Y	
安装和卸载功能	Y		
<b>Runtime</b>			
查看系统属性、指标和线程	Y	Y	Y
<b>Logs</b>			
查看日志	Y	Y	Y

## 其他资源

有关基于角色的访问控制的更多信息，请参阅 [部署到 Apache Karaf](#) 中。

## 3.5. 自定义 FUSE 控制台品牌

您可以使用 **Fuse Console branding** 插件自定义 Fuse 控制台品牌信息，如标题、徽标和登录页面信息。

默认情况下，Fuse 控制台品牌在 `hawtconfig.json` 中定义，该文件位于 Fuse Console WAR 文件中 (`karaf-install-dir/system/io/hawt/hawtio-war/<version>/hawtio-war-<version>.war`)。当您实施 **Fuse Console branding** 插件时，您可以使用您自己的自定义品牌覆盖默认品牌。

## 流程

1. 将品牌插件示例从 <https://github.com/hawtio/hawtio/tree/master/examples/branding-plugin> 下载到您选择的本地目录。
2. 在您选择的编辑器中，打开 **Fuse Console branding** 插件的 `src/main/webapp/plugin/brandingPlugin.js` 文件，以自定义 **Fuse Console branding**。

您可以更改 [表 A.1 “Fuse 控制台配置属性”](#) 中列出的配置属性的值。

3.

保存您的更改。

4.

在您选择的编辑器中，打开 **Fuse Console branding** 插件的 `pom.xml` 文件到它的 `< parent>` 部分：

```
<parent>
  <groupId>io.hawt</groupId>
  <artifactId>project</artifactId>
  <version>2.9-SNAPSHOT</version>
  <relativePath>../..</relativePath>
</parent>
```

5.

编辑 `< parent>` 部分，如下所示：

a.

更改 `< version>` 属性的值，使其与 Karaf 安装上的 Fuse 版本匹配。例如，如果您的 **Fuse on Karaf** 安装目录名称为 `2.0.0.fuse-760015`，请将版本设为 `2.0.0.fuse-760015`。

b.

删除 `< relativePath>../..</relativePath>` 行。

例如：

```
<parent>
  <groupId>io.hawt</groupId>
  <artifactId>project</artifactId>
  <version> 2.0.0.fuse-760015</version>
</parent>
```

6.

在终端窗口中，运行以下命令来构建 **branding-plugin** 项目：

```
mvn clean install
```

7.

如果 **Fuse** 尚未运行，请运行以下命令启动它：

**Linux/Unix:** `bin/fuse`

**Windows:** `bin\fuse.bat`



- 在 Karaf CLI 提示下，键入以下命令来安装 Fuse Console branding 插件（其中 `<version>` 是 Fuse on Karaf 安装的版本）：

Linux/Unix: `install -s mvn:io.hawt/branding-plugin/<version>/war`

Windows: `install -s mvn:io.hawt\branding-plugin\<version>\war`

- 在 Web 浏览器中，使用步骤 7 返回的 `start` 命令的 URL 打开 Fuse 控制台（默认 URL 为 <http://localhost:8181/hawtio/>）。



#### 注意

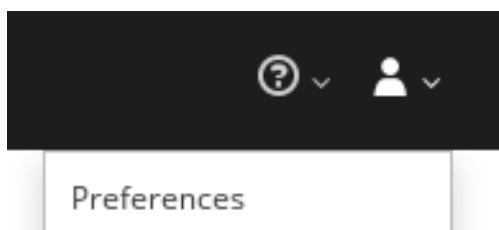
如果您已在网页浏览器中运行 Fuse 控制台，品牌会存储在浏览器的本地存储中。要使用新的品牌设置，您必须清除浏览器的本地存储。

### 3.6. 确保数据在 FUSE 控制台中正确显示

如果在 Fuse 控制台中显示队列和连接缺少队列、缺少连接或显示不一致图标，请调整 Jolokia 集合大小参数，以指定 Jolokia marshals 中的最大元素数。

#### 流程

- 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



- 增加 **Maximum collection size** 选项的值（默认为 50,000）。
- 单击 **Close**。

### 3.7. 禁用 FUSE 控制台

您可以在 Karaf 上禁用 Fuse 控制台，使其无法被所有用户访问，而不影响任何其他组件。

#### 流程

1. 要确定 hawtio-web 捆绑包 ID，请使用以下命令列出 Fuse 控制台使用的 Fuse 捆绑包：  

```
osgi:list | grep hawtio
```
2. 要停止捆绑包，请使用 `osgi:stop` 命令。例如，如果 `hawtio :: Web 控制台` 捆绑包的 ID 为 246，则键入这个命令：

```
osgi:stop 246
```

该捆绑包进入解析状态，您将无法再访问 Fuse 控制台。

#### 其他资源

有关管理捆绑包的更多信息，请参阅 [部署到 Apache Karaf](#) 中的“生命周期管理”章节。

### 3.8. 连接到远程 FUSE 应用程序

Fuse 控制台使用 Jolokia，这是一种基于代理的 Java 管理扩展(JMX)方法，需要在客户端上安装额外的软件（代理）。默认情况下，Red Hat Fuse 包含一个 jolokia 代理。

使用独立 Fuse 控制台发布，您可以连接到已在它们中运行的 jolokia 代理(<https://jolokia.org/>)的远程集成。如果要连接的进程没有 jolokia 代理，请参阅 jolokia 文档(<http://jolokia.org/agent.html>)。

#### 3.8.1. 解锁 Fuse 控制台

默认情况下，Apache Karaf 上的 Fuse 7 独立 Jolokia 被锁定，并且无法远程访问 Fuse 控制台。

要为 localhost 或 127.0.0.1 以外的主机名或 IP 地址解锁 Fuse 控制台，请按照以下步骤操作：

1. 在编辑器中打开 `$KARAF_HOME/etc/jolokia-access.xml` 文件。
2. 通过将其添加到 `< cors >` 部分，为您要访问 Fuse 集成注册主机名或 IP 地址。

例如，若要从 Fuse 控制台访问主机名 `0.0.0.3`，请添加

```
*<allow-origin>http://0.0.0.3:*</allow-origin>*
```

显示的行：

```
<!--
Cross-Origin Resource Sharing (CORS) restrictions

By default, only CORS access within localhost is allowed for maximum security.

You can add trusted hostnames in the <cors> section to unlock CORS access from them.

-->

<cors>

  <!-- Allow cross origin access only within localhost -->

  <allow-origin>http*://localhost:*</allow-origin>

  <allow-origin>http*://127.0.0.1:*</allow-origin>

  <allow-origin>http://0.0.0.3:*</allow-origin>

  <!-- Whitelist the hostname patterns as <allow-origin> -->

  <!--

  <allow-origin>http*://*.example.com</allow-origin>

  <allow-origin>http*://*.example.com:*</allow-origin>

  -->

  <!-- Check for the proper origin on the server side to protect against CSRF -->

  <strict-checking />

</cors>
```

3. 保存该文件。

### 3.8.2. 限制远程访问

另外，您可以限制对特定主机和 IP 地址的 Fuse 控制台的远程访问。

您可以根据 HTTP 客户端的 IP 地址授予整个访问权限。指定这些限制：

在 `jolokia-access.xml` 文件中，添加或编辑包含一个或多个 `< host >` 元素的 `<remote >` 部分。对于 `<host >` 元素，您可以为来自 10.0 网络的所有客户端指定一个 IP 地址、主机名或子网掩码（例如：`10.0.0.0/16`）。

以下示例允许从 `localhost` 以及其 IP 地址以 10.0 开头的所有客户端访问。对于所有其他 IP 地址，访问被拒绝。

```
<remote>
  <host>localhost</host>
  <host>10.0.0.0/16</host>
</remote>
```

详情请查看 Jolokia 安全文档(<https://jolokia.org/reference/html/security.html>)。

### 3.8.3. 允许连接到远程 Fuse 实例

Fuse 控制台的代理 `servlet` 使用白名单主机保护，默认情况下 Fuse 控制台只能连接到 `localhost`。如果要将 Fuse 控制台连接到其他远程 Fuse 实例，您需要配置白名单，如下所示：

对于 Apache Karaf，在 `etc/system.properties` 文件中进行以下配置更改：

```
hawtio.proxyWhitelist = localhost, 127.0.0.1, myhost1, myhost2, myhost3
```

### 3.8.4. 连接到远程 Jolokia 代理

开始之前，您需要知道远程 Jolokia 代理的连接详情（主机名、端口和路径）。

Apache Karaf 上 Fuse 的 Jolokia 代理的默认连接 URL 是 <http://<host>:8181/hawtio/jolokia>。

作为系统管理员，您可以更改此默认值。

通常，远程连接到 Jolokia 代理的 URL 是打开 Fuse 控制台加上 /jolokia 的 URL。例如，如果打开 Fuse 控制台的 URL 是 <http://<host>:1234/hawtio>，则远程连接的 URL 可能是 <http://<host>:1234/hawtio/jolokia>。

要连接到远程 Jolokia 实例，以便您可以检查其 JVM：

1. 点 **Connect** 标签页。
2. 点 **Remote** 选项卡，然后点 **Add connection**。
3. 键入 **Name**、**Scheme** (HTTP 或 HTTPS) 以及 **主机名**。
4. 单击 **Test Connection**。
5. 单击 **Add**。



#### 注意

Fuse 控制台自动探测 `localhost` 和 `127.0.0.1` 以外的本地网络接口，并将它们添加到白名单中。因此，您不需要手动将本地计算机的地址注册到白名单中。

### 3.8.5. 设置数据移动首选项

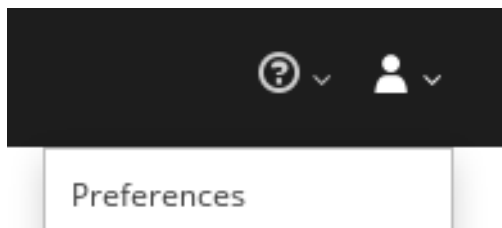
您可以更改以下 Jolokia 首选项，例如，如果要更频繁地刷新 Fuse 控制台中显示的数据。请注意，增加数据更新的频率会影响网络流量，并增加向服务器发出的请求数量。

- **更新速率** - 轮询到 Jolokia 以获取 JMX 数据之间的周期（默认为 5 秒）。

- 最大深度 - Jolokia 将在返回（默认为 7）前将对象聚合到服务器端的 JSON 级别数。
- 最大集合大小 - Jolokia marshals 在响应中的最大元素数（默认值为 50,000）。

更改这些设置的值：

1. 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 编辑选项，然后单击 **Close**。

### 3.8.6. 查看 JVM 运行时信息

要查看 JVM 运行时信息，如系统属性、指标和线程，请单击 **Runtime** 选项卡。

### 3.9. 查看并管理 APACHE CAMEL 应用程序

在 Fuse Console 的 Camel 选项卡中，您可以查看和管理 Apache Camel 上下文、路由和依赖项。

您可以查看以下详情：

- 所有正在运行的 Camel 上下文的列表
- 每个 Camel 上下文的详细信息，如 Camel 版本号和运行时静态

- 每个 Camel 应用程序及其运行时统计中的所有路由列表
- 正在运行的路由的图形表示以及实时指标

您还可以通过以下方法与 Camel 应用程序交互：

- 启动和停止上下文
- 管理所有 Camel 应用程序及其路由的生命周期，以便您可以重新启动、停止、暂停、恢复等。
- 正在运行的路由的实时追踪和调试
- 浏览并发送消息到 Camel 端点

#### 前提条件

只有在连接到使用一个或多个 Camel 路由的容器时，Camel 选项卡才可用。

#### 3.9.1. 启动、暂停或删除上下文

1. 在 Camel 选项卡的树视图中，单击 **Camel Contexts**。
2. 选中列表中一个或多个上下文旁边的框。
3. 单击 **Start** 或 **Suspend**。
4. 删除上下文：
  - a. 停止上下文。

- b. 单击省略图标，然后从下拉菜单中选择 **删除**。



#### 注意

删除上下文时，您可以将其从部署的应用中删除。

### 3.9.2. 查看 Camel 应用程序详情

1. 在 **Camel** 选项卡的树视图中，点 **Camel 应用程序**。
2. 要查看应用属性和值列表，请单击 **Attributes**。
3. 要查看应用程序属性的图形表示，请点 **Chart**，然后点 **Edit** 以选择您要在图表中看到的属性。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看应用端点，请单击 **Endpoints**。您可以根据 **URL**、**路由ID** 和 **方向** 过滤列表。
6. 要查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，该机制用于将消息正文和消息标头转换为不同类型的消息，请单击 **Type Converters**。
7. 要查看和执行 **JMX** 操作，如从 **XML** 添加或更新路由，或查找 **classpath** 中提供的所有 **Camel** 组件，请单击 **Operations**。

### 3.9.3. 查看 Camel 路由列表并与之交互

1. 查看路由列表：
  - a. 点 **Camel** 标签页。



- b. 在 tree 视图中，点应用程序的 routes 文件夹：

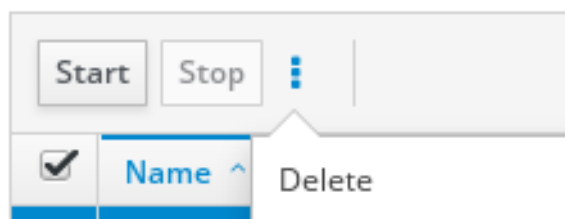
## Routes

<input type="checkbox"/>	Name ^	State
<input type="checkbox"/>	_route1	Started
<input type="checkbox"/>	_route2	Started

2. 启动、停止或删除一个或多个路由：

- a. 选中列表中一个或多个路由旁边的框。
- b. 点 Start 或 Stop。
- c. 要删除路由，您必须首先停止它。然后单击省略图标，然后从下拉菜单中选择 删除。

## Routes





### 注意

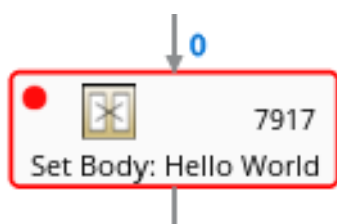
- 删除路由时，您可以将其从部署的应用程序中删除。
- 您也可以在树形视图中选择一个特定的路由，然后单击右上角的菜单来启动、停止或删除。

3. 要查看路由的图形图表，请点 **Route** 图表。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看端点，请单击 **Endpoints**。您可以根据 **URL**、**Route ID** 和 **direction** 来过滤列表。
6. 点 **Type Converters** 查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，用于将消息正文和消息标头转换为不同的类型。
7. 与特定路由交互：
  - a. 在 **Camel** 选项卡的树视图中，选择一个路由。
  - b. 要查看路由属性和值列表，请单击 **Attributes**。
  - c. 要查看路由属性的图形表示，请点 **Chart**。您可以点 **Edit** 来选择要在图表中看到的属性。
  - d. 要查看动态和阻止的交换，请点 **Exchanges**。
  - e. 单击 **Operations** 以在路由上查看并执行 **JMX** 操作，如以 **XML** 转储路由或获取路由的 **Camel ID** 值。
8. 通过路由跟踪信息：

- a. 在 Camel 选项卡的树视图中，选择一个路由。
  - b. 选择 Trace，然后单击 Start tracing。
9. 发送消息到路由：
- a. 在 Camel 选项卡的树视图中，打开上下文的端点文件夹，然后选择端点。
  - b. 点 Send 子选项卡。
  - c. 以 JSON 或 XML 格式配置消息。
  - d. 单击 Send。
  - e. 返回到路由的 Trace 选项卡，以查看通过路由的消息流。

#### 3.9.4. 调试路由

1. 在 Camel 选项卡的树视图中，选择一个路由。
2. 选择 Debug，然后单击 Start debugging。
3. 要添加断点，请在图表中选择节点，然后单击 Add breakpoint。节点上会出现一个红色点：



该节点被添加到断点列表中：

Breakpoints	
setBody1	✕
log1	✕

4. 单击向下箭头，以转至下一个节点或 **Play** 按钮，以恢复运行路由。
5. 单击 **暂停** 按钮，以暂停路由的所有线程。
6. 完成后，点 **Stop debug**。所有断点都会被清除。

### 3.10. 查看并管理 JMX 域和 MBEANS

**Java 管理扩展(JMX)**是一种 Java 技术，可让您在运行时动态管理资源（服务、设备和应用程序）。资源由名为 **MBeans**（受管 Bean）的对象表示。您可以在创建、实施或安装后立即管理和监控资源。

在 **Fuse** 控制台中使用 **JMX** 插件，您可以查看和管理 **JMX** 域和 **MBeans**。您可以查看 **MBean** 属性、运行命令并创建显示 **MBeans** 统计信息的图表。

**JMX** 选项卡提供活动 **JMX** 域和在文件夹中组织的 **MBeans** 的树视图。您可以查看 **MBeans** 的详情并执行命令。

#### 流程

1. 查看并编辑 **MBean** 属性：
  - a. 在 **tree** 视图中，选择一个 **MBean**。
  - b. 单击 **Attributes** 选项卡。

- c. 点属性查看其详情。
2. 执行操作：
    - a. 在 **tree** 视图中，选择一个 **MBean**。
    - b. 单击 **Operations** 选项卡，展开列出的操作之一。
    - c. 点 **Execute** 来运行操作。
  3. 查看 **chart**：
    - a. 在树形视图中，选择一个项目。
    - b. 单击 **Chart** 选项卡。

### 3.11. 查看和管理您的 OSGi 环境

对于 Apache Karaf 独立发行版，您可以查看和管理红帽 Fuse OSGi 环境。您可以查看和管理容器捆绑包、功能和配置，以及 Java 软件包和 OSGi 服务。

**OSGi 选项卡**包含一系列子选项卡，其中包含每个容器组件的选项：

#### 捆绑包(Bundle)

安装捆绑包列表。您可以安装和卸载捆绑包，启动和停止捆绑包，并编辑捆绑包属性。您还可以过滤列表并在列表和网格视图间切换。

#### 功能

可用功能列表。您可以安装和卸载功能或功能存储库，并深入查看功能详情。

#### 软件包

已安装的 Java 软件包列表。您可以查看软件包版本和相关捆绑包。

## 服务

正在运行的服务列表。您可以查看服务 ID、关联的捆绑包和对象类。

## 声明服务

声明 OSGi 服务列表。您可以查看服务状态并深入查看服务详情。您还可以激活和取消激活服务。

## 服务器

有关只读模式下本地或远程主机的详细信息。

## Framework

容器 OSGi 框架的配置选项。您可以设置框架启动级别和初始捆绑包启动级别。

## 配置

配置对象列表。您可以查看每个对象的状态，并深入查看或编辑对象详情。您还可以创建新的配置对象。

### 3.12. 查看诊断

使用 `diagnostics` 选项卡，通过 `JVM diagnosticsCommand` 和 `HotspotDiagnostic` 接口查看有关 JVM 的诊断信息。



#### 注意

该功能与 Java Mission Control (jmc) 或命令行工具 `jcmt` 中的 `diagnose Commands` 视图类似。在某些情况下，插件将提供对应的 `jcmt` 命令。

## 流程

1. 要检索载入类的实例数量以及它们所消耗的字节数，请单击 **Class Histogram**。如果操作重复，标签会显示自上次运行以来的区别。
2. 要查看 JVM 诊断标志设置，请单击 **JVM** 标志。

3. 对于正在运行的 JVM，您还可以修改标志设置。

## 其他资源

支持的 JVM 依赖于平台，如需更多信息，请访问以下源之一：

- <http://www.oracle.com/technetwork/java/vmoptions-jsp-140102.html>
- <http://openjdk.java.net/groups/hotspot/docs/RuntimeOverview.html>

### 3.13. 查看线程

您可以查看和监控线程的状态。

## 流程

1. 单击 **Runtime** 选项卡，然后单击 **线程** 子选项卡。**Threads** 页面列出了每个线程的活跃线程和堆栈追踪详情。默认情况下，线程列表以降序显示所有线程。
2. 要通过增加 ID 来对列表进行排序，请单击 ID 列标签。
3. (可选) 根据线程状态 (如 **Blocked**) 或线程名称过滤列表。
4. 要深入查看特定线程的详细信息，如锁定类名称和该线程的完整堆栈追踪，请在 **Actions** 列中点 **More**。

### 3.14. 查看日志条目

您可以在 **Logs** 选项卡中查看 Red Hat Fuse 的日志条目。

## 前提条件

当 Java 应用程序包含 Log MBean 时，可以使用 **Logs** 选项卡。

## 流程

1. 要查看日志条目列表，请点 **Log Entries** 选项卡。

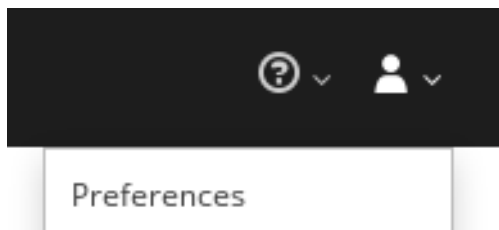
默认情况下，列表以升序显示日志条目。

您可以深入到每个日志条目中，以查看日志条目的详细信息。

2. 要过滤日志列表以显示特定的日志类型，请单击 **Action Bar**。您可以根据文本字符串或日志记录级别过滤日志条目部分。

3. 要更改 **Fuse Console** 默认设置：

- a. 在 **Fuse** 控制台右上角，单击用户图标，然后单击下拉菜单中的 **Preferences**。



- b. 要更改默认的排序顺序，请选择 **Server Logs**，然后单击日志条目链接以深入到日志条目的详细信息，如捆绑包名称、线程和完整消息文本。

- c. 另外，您可以自定义这些设置来存储日志信息：

- 在 **Fuse** 控制台中保留的日志语句数（默认为 100）。
- 全局日志级别：**INFO**（默认）、**OFF**、**ERROR**、**WARN** 和 **DEBUG**。
- 要包含的子级消息，如 **hawtio-oauth** 和 **hawtio-core-utils**。



d.

要将 Fuse Console Logs 设置重置为默认值，请点击 **Reset** → **Reset settings**。

### 3.15. 启用 PROMETHEUS 指标

Prometheus 是一个开源系统和服务监控工具包，可用于监控独立 Apache Karaf 容器中部署的服务。Prometheus 以给定间隔收集并存储已配置的服务的指标，评估规则表达式，显示结果，并在指定条件变为 true 时触发警报。



#### 重要

红帽对 Prometheus 的支持仅限于红帽产品文档中提供的设置和配置建议。

Prometheus 使用在客户端上安装和配置的 "exporters" 来公开端到 Prometheus 格式。此端点是一个 HTTP 接口，提供指标列表和指标的当前值。Prometheus 定期提取每个目标定义的端点，并将收集的数据写入其数据库。Prometheus 在延长时间内收集数据，而不仅仅是针对当前运行的会话。Prometheus 存储数据，以便您可以以图形方式可视化并在数据上运行查询。

#### 3.15.1. 启用来自独立 Apache Karaf 容器的指标导出

Prometheus 使用配置文件(<https://raw.githubusercontent.com/jboss-fuse/application-templates/master/prometheus/prometheus-config.yml>)，其中包含由 Camel 公开的指标。



#### 注意

您可以识别的指标仅限于 JMX 中提供的指标。

您必须部署一个 Fuse 应用程序，以便生成 Apache Camel 指标。

#### 流程

使用命令行从独立 Apache Karaf 容器启用 Prometheus 指标导出：

1.

打开命令提示符，并确保您处于 Apache Karaf 安装的 etc/ 目录中。

2.

键入以下命令，从 `etc/` 目录中的示例文件创建 Prometheus 配置文件：

```
cp prometheus-config.yml-example prometheus-config.yml
```

3.

导出器只有在您使用 `fuse` 或 `fuse.bat` 命令启动 Fuse 时可用：在 Linux/Unix 或 `bin/fuse.bat` 上运行 `bin/fuse`。

4.

当 Fuse 重启时，您可以通过打开 Web 浏览器来查看公开的指标：<http://localhost:9779>



注意

(可选) 您可以更改命令行中 `KARAF_PROMETHEUS_PORT` 和 `KARAF_PROMETHEUS_CONFIG` 配置变量的默认值。

### 3.15.2. 配置 Prometheus 服务器以从 Apache Karaf 容器中提取公开指标

要让 Prometheus 服务器从 Apache Karaf 容器提取指标，必须将公开指标的端点添加到 Prometheus 配置文件的 `targets` 属性中。

#### 流程

1.

进入 Prometheus 安装目录中的 `/prometheus.yml` 配置文件。

2.

添加 Apache Karaf 端点以提取：

```
scrape_configs:
- job_name: 'prometheus'

  # metrics_path defaults to '/metrics'
  # scheme defaults to 'http'.
  static_configs:
  - targets: ['localhost:9779']
```

## 第 4 章 在 EAP 独立上监控和管理红帽 FUSE 应用程序

### 4.1. 关于 FUSE 控制台

红帽 Fuse 控制台是一个基于 Hawtio 开源软件的 Web 控制台。如需支持的浏览器列表，请访问 [支持的配置](#)。

Fuse 控制台提供了一个中央接口，用于检查和管理一个或多个部署 Fuse 容器的详细信息。您还可以监控红帽 Fuse 和系统资源，执行更新，以及启动或停止服务。

安装 Red Hat Fuse 独立或使用 OpenShift 中的 Fuse 时，可以使用 Fuse 控制台。您可以在 Fuse 控制台中查看和管理的集成取决于正在运行的插件。可能的插件包括：

- Camel
- JMX
- OSGI
- Runtime
- Logs

### 4.2. 访问 FUSE 控制台

按照以下步骤访问 Red Hat JBoss Enterprise Application Platform 的 Fuse 控制台。

#### 前提条件

您必须在 JBoss EAP 容器上安装 Fuse。有关逐步说明，请参阅 [在 JBoss EAP 上安装](#)。

#### 流程

要访问独立 JBoss EAP 发行版本的 Fuse 控制台：

1. 使用以下命令启动 Red Hat Fuse standalone:  
  
在 Linux/Mac OS 上：`./bin/standalone.sh`  
  
在 Windows 上：`./bin/standalone.bat`
2. 在 Web 浏览器中，输入用于连接 Fuse 控制台的 URL。例如：  
<http://localhost:8080/hawtio>
3. 在登录页面中，输入您的用户名和密码，然后单击 **Log In**。

默认情况下，Fuse Console 显示 Home 页面。左侧导航选项卡表示正在运行的插件。



#### 注意

如果主 Fuse Console 页面需要很长时间才能在浏览器中显示，您可能需要减少日志文件的数量和大小。当文件达到最大大小(`rotate-size`)并维护多个文件(`max-backup-index`)时，您可以使用 `periodic-size-rotating-file-handler` 轮转文件。有关如何使用此处理程序的详情，请查看 Red Hat JBoss Enterprise Application Platform 产品文档。

### 4.3. 自定义 FUSE 控制台品牌

您可以使用 Fuse Console branding 插件自定义 Fuse 控制台品牌信息，如标题、徽标和登录页面信息。

默认情况下，Fuse 控制台品牌在 `hawtconfig.json` 中定义，该文件位于 Fuse Console WAR 文件中 (`eap-install-dir/standalone/deployments/hawtio-wildfly-<version>.war`)。当您实施 Fuse Console branding 插件时，您可以使用您自己的自定义品牌覆盖默认品牌。

#### 流程

1. 将品牌插件示例从 <https://github.com/hawtio/hawtio/tree/master/examples/branding-plugin> 下载到您选择的本地目录。

2. 在您选择的编辑器中，打开 Fuse Console branding 插件的 `src/main/webapp/plugin/brandingPlugin.js` 文件，以自定义 Fuse Console branding。

您可以更改 [表 A.1 “Fuse 控制台配置属性”](#) 中列出的配置属性的值。

3. 保存您的更改。
4. 在您选择的编辑器中，打开 Fuse Console branding 插件的 `pom.xml` 文件到它的 `<parent>` 部分：

```
<parent>
  <groupId>io.hawt</groupId>
  <artifactId>project</artifactId>
  <version>2.9-SNAPSHOT</version>
  <relativePath>../..</relativePath>
</parent>
```

5. 编辑 `<parent>` 部分，如下所示：
  - a. 更改 `<version>` 属性的值，使其与 EAP 安装中的 Fuse 版本匹配。例如，如果您的 EAP 安装目录名称为 `2.0.0.fuse-760015` 的 Fuse 是 `2.0.0.fuse-760015`，请将版本设为 `2.0.0.fuse-760015`。
  - b. 删除 `<relativePath>../..</relativePath>` 行。

例如：

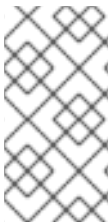
```
<parent>
  <groupId>io.hawt</groupId>
  <artifactId>project</artifactId>
  <version> 2.0.0.fuse-760015</version>
</parent>
```

6. 在终端窗口中，运行以下命令来构建 `branding-plugin` 项目：

```
mvn clean install
```

此命令在项目的 `/target` 文件夹中创建一个 `branding-plugin.war` 文件。

7. 将 `branding-plugin.war` 文件复制到 EAP 安装的 `standalone/deployments` 目录中。
8. 如果 Fuse 尚未运行，请运行以下命令启动它：  
  
在 Linux/Mac OS 上：`./bin/standalone.sh`  
  
在 Windows 上：`./bin/standalone.bat`
9. 在 Web 浏览器中，使用上一步中返回的 `start` 命令的 URL（默认 URL 为 <http://localhost:8080/hawtio>），打开 Fuse 控制台。



#### 注意

如果您已在网页浏览器中运行 Fuse 控制台，品牌会存储在浏览器的本地存储中。要使用新的品牌设置，您必须清除浏览器的本地存储。

## 4.4. 保护 FUSE 控制台

在 EAP 上保护 Fuse 控制台：

- 在部署到 AWS 时禁用 Fuse 控制台的代理 servlet

如果要独立部署 Fuse 应用程序到 Amazon Web Services (AWS)，您应该通过将 `hawtio.disableProxy` 系统属性设置为 `true` 来禁用 Fuse 控制台的代理 servlet。



#### 注意

当您禁用 Fuse 控制台代理 servlet 时，Fuse Console 的 Connect 选项卡被禁用，您无法从 Fuse 控制台连接到其他 JVM。如果要在 AWS 上部署多个 Fuse 应用程序，您必须为每个应用程序部署 Fuse 控制台。

- 将 HTTPS 设置为所需的协议

您可以使用 `hawtio.http.strictTransportSecurity` 属性来要求 Web 浏览器使用安全 HTTPS 协议来访问 Fuse 控制台。此属性指定尝试使用 HTTP 访问 Fuse 控制台的 Web 浏览器必须自动转换请求以使用 HTTPS。

- 使用公钥来保护响应

您可以使用 `hawtio.http.publicKeyPins` 属性来保护 HTTPS 协议，方法是告诉 Web 浏览器将特定的加密密钥与 Fuse 控制台相关联，以降低"man-in-the-middle"攻击的风险。

## 流程

1. 在 `$EAP_HOME/standalone/configuration/standalone*.xml` 文件的 `system-properties` 部分中设置 `hawtio.http.strictTransportSecurity` 和 `hawtio.publicKeyPins` 属性，如下例所示：

```
<property name="hawtio.http.strictTransportSecurity" value="max-age=31536000;
includeSubDomains; preload"/>
<property name="hawtio.http.publicKeyPins" value="pin-
sha256=cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs"; max-age=5184000;
includeSubDomains"/>
```

2. (用于仅在 AWS 上部署) 要禁用 Fuse 控制台的代理 servlet，请在 `$EAP_HOME/standalone/configuration/standalone*.xml` 文件的 `system-properties` 部分中设置 `hawtio.disableProxy` 属性，如下例所示：

```
<property name="hawtio.disableProxy" value="true"/>
```

## 其他资源

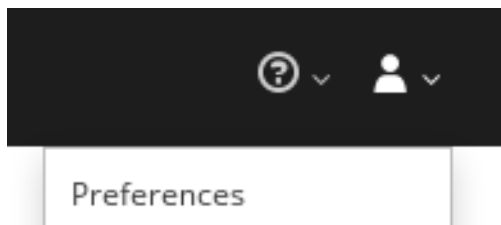
- 有关 `hawtio.http.strictTransportSecurity` 属性的语法的描述，请参阅 [HTTP Strict Transport Security \(HSTS\) 响应标头的描述](#) 页面。
- 有关 `hawtio.http.publicKeyPins` 属性的语法的描述，包括如何提取 Base64 编码的公钥的说明，请参阅 [HTTP Public Key Pinning 响应标头的描述](#) 页面。

## 4.5. 确保数据在 FUSE 控制台中正确显示

如果在 Fuse 控制台中显示队列和连接缺少队列、缺少连接或显示不一致图标，请调整 Jolokia 集合大小参数，以指定 Jolokia marshals 中的最大元素数。

## 流程

1. 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 增加 **Maximum collection size** 选项的值（默认为 50,000）。
3. 单击 **Close**。

## 4.6. 禁用 FUSE 控制台

您可以在 JBoss EAP 上禁用 Fuse 控制台，使其无法被所有用户访问，而不影响任何其他组件。

## 流程

要在 JBoss EAP 上禁用 Fuse 控制台，请执行以下操作之一：

- 删除 Fuse Console 部署文件：`$EAP_HOME/standalone/deployments/hawtio-wildfly-xxxxx.war`
- 使用 JBoss EAP 管理控制台或命令行界面取消部署 Fuse 控制台。

## 4.7. 连接到远程 FUSE 应用程序

Fuse 控制台使用 Jolokia，这是一种基于代理的 Java 管理扩展(JMX)方法，需要在客户端上安装额外的软件（代理）。默认情况下，Red Hat Fuse 包含一个 jolokia 代理。



使用独立 Fuse 控制台发布，您可以连接到已在它们中运行的 jolokia 代理(<https://jolokia.org/>)的远程集成。如果要连接的进程没有 jolokia 代理，请参阅 jolokia 文档(<http://jolokia.org/agent.html>)。

#### 4.7.1. 解锁 Fuse 控制台

默认情况下，JBoss EAP 上的 Fuse 7 独立 Jolokia 被锁定，并且无法远程访问 Fuse 控制台。

要为 localhost 或 127.0.0.1 以外的主机名或 IP 地址解锁 Fuse 控制台，请按照以下步骤操作：

1. 在编辑器中打开 `$EAP_HOME/standalone/configuration/jolokia-access.xml` 文件。
2. 通过将其添加到 `< cors >` 部分，为您要访问 Fuse 集成注册主机名或 IP 地址。

例如，若要从 Fuse 控制台访问主机名 0.0.0.3，请添加

```
*<allow-origin>http://0.0.0.3:*</allow-origin>*
```

显示的行：

```
<!--
Cross-Origin Resource Sharing (CORS) restrictions
By default, only CORS access within localhost is allowed for maximum security.
You can add trusted hostnames in the <cors> section to unlock CORS access from them.
-->
<cors>
  <!-- Allow cross origin access only within localhost -->
  <allow-origin>http*://localhost:*</allow-origin>
  <allow-origin>http*://127.0.0.1:*</allow-origin>
  <allow-origin>http://0.0.0.3:*</allow-origin>
  <!-- Whitelist the hostname patterns as <allow-origin> -->
<!--
```

```

<allow-origin>http://*.example.com</allow-origin>

<allow-origin>http://*.example.com:*</allow-origin>

-->

<!-- Check for the proper origin on the server side to protect against CSRF -->

<strict-checking />

</cors>

```

3. 保存该文件。

#### 4.7.2. 限制远程访问

另外，您可以限制对特定主机和 IP 地址的 Fuse 控制台的远程访问。

您可以根据 HTTP 客户端的 IP 地址授予整个访问权限。指定这些限制：

在 `jolokia-access.xml` 文件中，添加或编辑包含一个或多个 `< host >` 元素的 `<remote >` 部分。对于 `<host >` 元素，您可以为来自 10.0 网络的所有客户端指定一个 IP 地址、主机名或子网掩码（例如：`10.0.0.0/16`）。

以下示例允许从 `localhost` 以及其 IP 地址以 `10.0` 开头的所有客户端访问。对于所有其他 IP 地址，访问被拒绝。

```

<remote>
  <host>localhost</host>
  <host>10.0.0.0/16</host>
</remote>

```

详情请查看 Jolokia 安全文档(<https://jolokia.org/reference/html/security.html>)。

#### 4.7.3. 允许连接到远程 Fuse 实例

Fuse 控制台的代理 `Servlet` 使用白名单主机保护，默认情况下 Fuse 控制台只能连接到 `localhost`。如果要将 Fuse 控制台连接到其他远程 Fuse 实例，请在 `standalone/configuration/standalone-factory.xml` 文件中进行以下配置更改：

■

```
<property name=hawtio.proxyWhitelist" value="localhost, 127.0.0.1, myhost1, myhost2, myhost3"/>
```

#### 4.7.4. 连接到远程 Jolokia 代理

开始之前，您需要知道远程 Jolokia 代理的连接详情（主机名、端口和路径）。

Red Hat JBoss EAP 上的 Jolokia 代理的默认连接 URL 是 <http://<host>:8080/hawtio/jolokia>。

作为系统管理员，您可以更改此默认值。

通常，远程连接到 Jolokia 代理的 URL 是打开 Fuse 控制台加上 /jolokia 的 URL。例如，如果打开 Fuse 控制台的 URL 是 <http://<host>:1234/hawtio>，则远程连接的 URL 可能是 <http://<host>:1234/hawtio/jolokia>。

要连接到远程 Jolokia 实例，以便您可以检查其 JVM：

1. 点 **Connect** 标签页。
2. 点 **Remote** 选项卡，然后点 **Add connection**。
3. 键入 **Name**、**Scheme** (HTTP 或 HTTPS) 以及 **主机名**。
4. 单击 **Test Connection**。
5. 单击 **Add**。



#### 注意

Fuse 控制台自动探测 localhost 和 127.0.0.1 以外的本地网络接口，并将它们添加到白名单中。因此，您不需要手动将本地计算机的地址注册到白名单中。

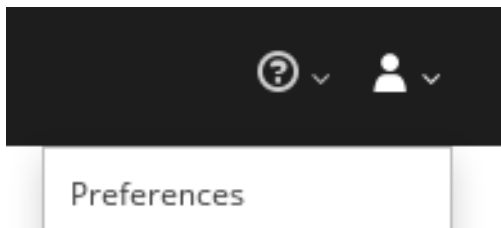
#### 4.7.5. 设置数据移动首选项

您可以更改以下 Jolokia 首选项，例如，如果要更频繁地刷新 Fuse 控制台中显示的数据。请注意，增加数据更新的频率会影响网络流量，并增加向服务器发出的请求数量。

- 更新速率 - 轮询到 Jolokia 以获取 JMX 数据之间的周期（默认为 5 秒）。
- 最大深度 - Jolokia 将在返回（默认为 7）前将对象聚合到服务器端的 JSON 级别数。
- 最大集合大小 - Jolokia marshals 在响应中的最大元素数（默认值为 50,000）。

更改这些设置的值：

1. 在 Fuse 控制台右上角，单击用户图标，然后单击 **Preferences**。



2. 编辑选项，然后单击 **Close**。

#### 4.7.6. 查看 JVM 运行时信息

要查看 JVM 运行时信息，如系统属性、指标和线程，请单击 **Runtime** 选项卡。

#### 4.8. 查看并管理 APACHE CAMEL 应用程序

在 Fuse Console 的 Camel 选项卡中，您可以查看和管理 Apache Camel 上下文、路由和依赖项。

您可以查看以下详情：

- 所有正在运行的 Camel 上下文的列表
- 每个 Camel 上下文的详细信息，如 Camel 版本号和运行时静态
- 每个 Camel 应用程序及其运行时统计中的所有路由列表
- 正在运行的路由的图形表示以及实时指标

您还可以通过以下方法与 Camel 应用程序交互：

- 启动和停止上下文
- 管理所有 Camel 应用程序及其路由的生命周期，以便您可以重新启动、停止、暂停、恢复等。
- 正在运行的路由的实时追踪和调试
- 浏览并发送消息到 Camel 端点

#### 前提条件

只有在连接到使用一个或多个 Camel 路由的容器时，Camel 选项卡才可用。

#### 4.8.1. 启动、暂停或删除上下文

1. 在 Camel 选项卡的树视图中，单击 **Camel Contexts**。
2. 选中列表中一个或多个上下文旁边的框。
3. 单击 **Start** 或 **Suspend**。

4. 删除上下文：
  - a. 停止上下文。
  - b. 单击省略图标，然后从下拉菜单中选择 **删除**。

**注意**

删除上下文时，您可以将其从部署的应用中删除。

#### 4.8.2. 查看 Camel 应用程序详情

1. 在 **Camel** 选项卡的树视图中，点 **Camel 应用程序**。
2. 要查看应用属性和值列表，请单击 **Attributes**。
3. 要查看应用程序属性的图形表示，请点 **Chart**，然后点 **Edit** 以选择您要在图表中看到的属性。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看应用端点，请单击 **Endpoints**。您可以根据 **URL**、**路由ID** 和 **方向** 过滤列表。
6. 要查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，该机制用于将消息正文和消息标头转换为不同类型的消息，请单击 **Type Converters**。
7. 要查看和执行 **JMX** 操作，如从 **XML** 添加或更新路由，或查找 **classpath** 中提供的所有 **Camel** 组件，请单击 **Operations**。

#### 4.8.3. 查看 Camel 路由列表并与之交互

1.

查看路由列表：

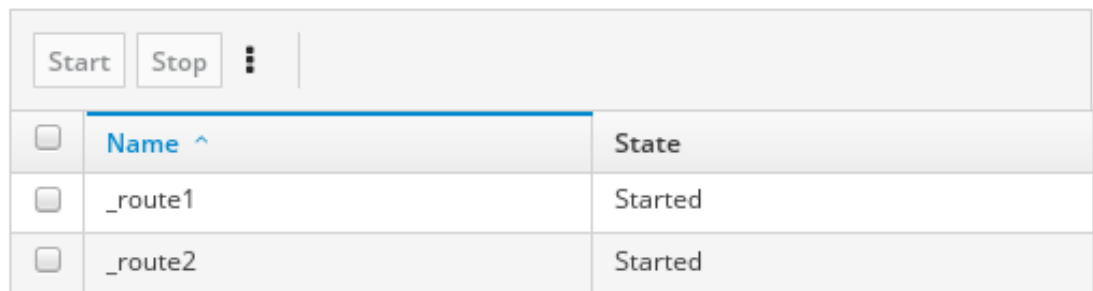
a.

点 Camel 标签页。

b.

在 tree 视图中，点应用程序的 routes 文件夹：

## Routes



<input type="checkbox"/>	Name ^	State
<input type="checkbox"/>	_route1	Started
<input type="checkbox"/>	_route2	Started

2.

启动、停止或删除一个或多个路由：

a.

选中列表中一个或多个路由旁边的框。

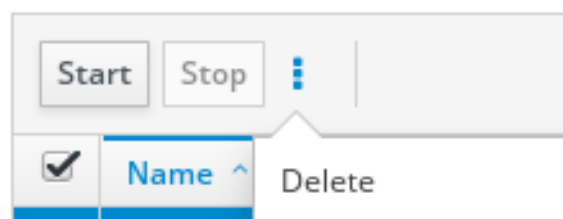
b.

点 Start 或 Stop。

c.

要删除路由，您必须首先停止它。然后单击省略图标，然后从下拉菜单中选择 删除。

## Routes





### 注意

- 删除路由时，您可以将其从部署的应用程序中删除。
- 您也可以在树形视图中选择一个特定的路由，然后单击右上角的菜单来启动、停止或删除。

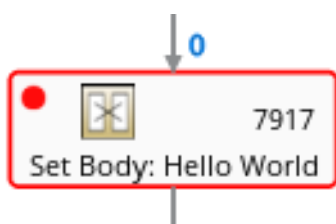
3. 要查看路由的图形图表，请点 **Route** 图表。
4. 要查看动态和阻止的交换，请点 **Exchanges**。
5. 要查看端点，请单击 **Endpoints**。您可以根据 **URL**、**Route ID** 和 **direction** 来过滤列表。
6. 点 **Type Converters** 查看、启用和禁用与 **Camel** 内置类型转换机制相关的统计信息，用于将消息正文和消息标头转换为不同的类型。
7. 与特定路由交互：
  - a. 在 **Camel** 选项卡的树视图中，选择一个路由。
  - b. 要查看路由属性和值列表，请单击 **Attributes**。
  - c. 要查看路由属性的图形表示，请点 **Chart**。您可以点 **Edit** 来选择要在图表中看到的属性。
  - d. 要查看动态和阻止的交换，请点 **Exchanges**。
  - e. 单击 **Operations** 以在路由上查看并执行 **JMX** 操作，如以 **XML** 转储路由或获取路由的 **Camel ID** 值。
8. 通过路由跟踪信息：



- a. 在 **Camel** 选项卡的树视图中，选择一个路由。
  - b. 选择 **Trace**，然后单击 **Start tracing**。
9. 发送消息到路由：
- a. 在 **Camel** 选项卡的树视图中，打开上下文的端点文件夹，然后选择端点。
  - b. 点 **Send** 子选项卡。
  - c. 以 **JSON** 或 **XML** 格式配置消息。
  - d. 单击 **Send**。
  - e. 返回到路由的 **Trace** 选项卡，以查看通过路由的消息流。

#### 4.8.4. 调试路由

1. 在 **Camel** 选项卡的树视图中，选择一个路由。
2. 选择 **Debug**，然后单击 **Start debugging**。
3. 要添加断点，请在图表中选择节点，然后单击 **Add breakpoint**。节点上会出现一个红色点：



该节点被添加到断点列表中：

Breakpoints	
setBody1	✕
log1	✕

4. 单击向下箭头，以转至下一个节点或 **Play** 按钮，以恢复运行路由。
5. 单击 **暂停** 按钮，以暂停路由的所有线程。
6. 完成后，点 **Stop debug**。所有断点都会被清除。

#### 4.9. 查看并管理 JMX 域和 MBEANS

**Java 管理扩展(JMX)**是一种 Java 技术，可让您在运行时动态管理资源（服务、设备和应用程序）。资源由名为 **MBeans**（受管 Bean）的对象表示。您可以在创建、实施或安装后立即管理和监控资源。

在 **Fuse** 控制台使用 **JMX** 插件，您可以查看和管理 **JMX** 域和 **MBeans**。您可以查看 **MBean** 属性、运行命令并创建显示 **MBeans** 统计信息的图表。

**JMX** 选项卡提供活动 **JMX** 域和在文件夹中组织的 **MBeans** 的树视图。您可以查看 **MBeans** 的详情并执行命令。

##### 流程

1. 查看并编辑 **MBean** 属性：
  - a. 在 **tree** 视图中，选择一个 **MBean**。
  - b. 单击 **Attributes** 选项卡。

- c. 点属性查看其详情。
2. 执行操作：
    - a. 在 **tree** 视图中，选择一个 **MBean**。
    - b. 单击 **Operations** 选项卡，展开列出的操作之一。
    - c. 点 **Execute** 来运行操作。
3. 查看 **chart**：
    - a. 在树形视图中，选择一个项目。
    - b. 单击 **Chart** 选项卡。

#### 4.10. 查看诊断

使用 **diagnostics s** 选项卡，通过 **JVM diagnosticsCommand** 和 **HotspotDiangostic** 接口查看有关 **JVM** 的诊断信息。



#### 注意

该功能与 **Java Mission Control (jmc)** 或命令行工具 **jcmm** 中的 **diagnose Commands** 视图类似。在某些情况下，插件将提供对应的 **jcmm** 命令。

#### 流程

1. 要检索载入类的实例数量以及它们所消耗的字节数，请单击 **Class Histogram**。如果操作重复，标签会显示自上次运行以来的区别。
2. 要查看 **JVM** 诊断标志设置，请单击 **JVM** 标志。

3. 对于正在运行的 JVM，您还可以修改标志设置。

## 其他资源

支持的 JVM 依赖于平台，如需更多信息，请访问以下源之一：

- <http://www.oracle.com/technetwork/java/vmoptions-jsp-140102.html>
- <http://openjdk.java.net/groups/hotspot/docs/RuntimeOverview.html>

## 4.11. 查看线程

您可以查看和监控线程的状态。

### 流程

1. 单击 **Runtime** 选项卡，然后单击 **线程** 子选项卡。**Threads** 页面列出了每个线程的活跃线程和堆栈追踪详情。默认情况下，线程列表以降序显示所有线程。
2. 要通过增加 ID 来对列表进行排序，请单击 ID 列标签。
3. (可选) 根据线程状态 (如 **Blocked**) 或线程名称过滤列表。
4. 要深入查看特定线程的详细信息，如锁定类名称和该线程的完整堆栈追踪，请在 **Actions** 列中点 **More**。

## 4.12. 查看日志条目

您可以在 **Logs** 选项卡中查看 Red Hat Fuse 的日志条目。

### 前提条件

当 Java 应用程序包含 Log MBean 时，可以使用 **Logs** 选项卡。

## 流程

1. 要查看日志条目列表，请点 **Log Entries** 选项卡。

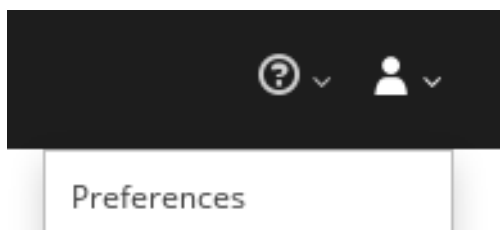
默认情况下，列表以升序显示日志条目。

您可以深入到每个日志条目中，以查看日志条目的详细信息。

2. 要过滤日志列表以显示特定的日志类型，请单击 **Action Bar**。您可以根据文本字符串或日志记录级别过滤日志条目部分。

3. 要更改 **Fuse Console** 默认设置：

- a. 在 **Fuse** 控制台右上角，单击用户图标，然后单击下拉菜单中的 **Preferences**。



- b. 要更改默认的排序顺序，请选择 **Server Logs**，然后单击日志条目链接以深入到日志条目的详细信息，如捆绑包名称、线程和完整消息文本。

- c. 另外，您可以自定义这些设置来存储日志信息：

- 在 **Fuse** 控制台中保留的日志语句数（默认为 100）。
- 全局日志级别：**INFO**（默认）、**OFF**、**ERROR**、**WARN** 和 **DEBUG**。
- 要包含的子级消息，如 **hawtio-oauth** 和 **hawtio-core-utils**。

d.

要将 **Fuse Console Logs** 设置重置为默认值，请点击 **Reset** → **Reset settings**。

## 附录 A. FUSE 控制台配置属性

默认情况下，Fuse 控制台配置在 `hawtconfig.json` 文件中定义。您可以自定义 Fuse 控制台配置信息，如标题、徽标和登录页面信息。

表 A.1 “Fuse 控制台配置属性” 提供对属性的描述，并列出了每个属性是否需要值。

表 A.1. Fuse 控制台配置属性

部分	属性名称	默认值	描述	必需？
关于	标题	Red Hat Fuse Management Console	在 Fuse 控制台的 About 页面中显示的标题。	必填
	productInfo	空值	显示在 Fuse 控制台的 About 页面中的产品信息。	选填
	additionalInfo	空值	显示在 Fuse 控制台的 About 页面中显示的任何其他信息。	选填
	版权	空值	显示在 Fuse 控制台的 About 页面上的版权信息。	选填
	imgSrc	<b>img/Logo-RedHat-A-Reverse-RGB.png</b>	Fuse Console 的 About 页面中出现的镜像。	必填
品牌	appName	Red Hat Fuse Management Console	应用程序的名称。此名称显示在 Fuse 控制台的标题栏中。	必填
	appLogoUrl	<b>img/Logo-Red_Hat-Fuse-A-Reverse-RGB.png</b>	Fuse Console <code>{navigation}</code> 栏中显示的应用程序徽标图像文件的路径。该值可以是相对于 Hawtio 状态 URL 或绝对 URL 的路径。	必填

部分	属性名称	默认值	描述	必需?
	CSS		外部 CSS 风格表的 URL，可用于风格应用程序。它可以是相对于 Hawtio 状态 URL 的路径，也可以是绝对 URL。	选填
	companyLogoUrl	<b>img/Logo-RedHat-A-Reverse- RGB.png</b>	贵公司徽标图像文件的路径。	必填
	favicon		favicon 的 URL，通常显示在 Web 浏览器标签页中。它可以是相对于 Hawtio 状态 URL 的路径，也可以是绝对 URL。	选填
login	description	<i>空值</i>	在 Fuse Console Login 页面中显示的描述性文本（例如 <a href="http://localhost:8181/hawtio">http://localhost:8181/hawtio</a> ）。	选填
	links	[ ]	指定 "url" 和 "文本" 对的数组，以向用户获取更多信息或帮助的页面提供额外的链接。	选填
disabledRoutes	<i>none</i>	[ ]	在控制台中禁用特定路径（如插件）。不要更改本节。OpenShift 以外的发行版不支持任何更改。	选填