Red Hat Fuse 7.9

Fuse の管理

Fuse Console を使用した Fuse アプリケーションの管理

Last Updated: 2023-07-26

Red Hat Fuse 7.9 Fuse の管理

Fuse Console を使用した Fuse アプリケーションの管理

法律上の通知

概要

Fuse アプリケーションをデプロイするときに Fuse Console を使用すると Red Hat Fuse インテグレーションを監視し、対話することができます。

目次

はじめに

Red Hat Fuse は、Fuse インテグレーションを表示および管理する、以下の 2 つのエンタープライズ管理ツールを提供します。

- Fuse Console は、ブラウザーからアクセスする web ベースのコンソールで、実行中の Fuse コンテナーを監視および管理します。Fuse Console は Hawtio オープンソースソフトウェア (https://hawt.io/) をベースにしています。本ガイドでは Fuse Console の使用方法を説明します。

- Prometheus は、Fuse ディストリビューションのシステムおよびインテグレーションレベルのメトリクスを保存します。Grafana などのグラフィカル分析インターフェイスを使用して、保存された履歴データを表示および分析できます。Prometheus の使用に関する詳細は、以下のドキュメントを参照してください。

  - Prometheus ドキュメント
  - Fuse on OpenShift ガイド
  - OpenShift Container Platform での Fuse Online のインストールと操作

本ガイドの対象読者は Red Hat Fuse の管理者です。本ガイドの読者は、Red Hat Fuse プラットフォーム、Apache Camel、および所属組織の処理要件をよく理解していることを前提としています。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

第1章 OPENSHIFT における RED HAT FUSE アプリケーションの監視および管理

1.1. FUSE CONSOLE

Red Hat Fuse Console は、HawtIO オープンソースソフトウェアをベースとする Web コンソールです。サポートされるブラウザーのリストは Red Hat Fuse でサポートされる設定 を参照してください。

Fuse Console は、デプロイされた 1 つ以上の Fuse コンテナーの詳細を確認および管理する中央インターフェイスを提供します。また、Red Hat Fuse およびシステムリソースの監視、更新の実行、およびサービスの開始と停止を行うこともできます。

Fuse Console は、Red Hat Fuse スタンドアロンをインストールしたり Fuse on OpenShift の使用すると利用することができます。Fuse Console で表示および管理できるインテグレーションは、実行されているプラグインによって異なります。使用できるプラグインには以下が含まれます。

- Camel
- JMX
- OSGI
- Runtime
- Logs

1.2. OPENSHIFT 4.X での FUSE CONSOLE の設定

OpenShift 4.x では、Fuse Console の設定に、インストール、およびデプロイが必要になります。Fuse Console のインストールおよびデプロイには、以下のオプションがあります。

- 「OperatorHub を使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ」
  特定の namespace で Fuse アプリケーションにアクセスするために、Fuse Console Operator を使用して Fuse Console をインストールおよびデプロイします。Operator は、Fuse Console のセキュリティー保護を処理します。

- 「コマンドラインを使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ」
  OpenShift クラスターの複数の namespace または特定の namespace にある Fuse アプリケーションにアクセスするために、コマンドラインと Fuse Console テンプレートの 1 つを使用して Fuse Console をインストールおよびデプロイします。デプロイ前にクライアント証明書を生成して、Fuse Console をセキュアにする必要があります。

任意で、「OpenShift 4.x 上の Fuse Console のロールベースアクセス制御」 の説明通りに、Fuse Console のロールベースアクセス制御 (RBAC) をカスタマイズすることができます。

1.2.1. OperatorHub を使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ

OpenShift 4.x に Fuse Console をインストールするには、OpenShift OperatorHub で提供される Fuse Console Operator を使用します。Fuse Console をデプロイするには、インストールされた Operator のインスタンスを作成します。

前提条件

- コンテナーイメージの **registry.redhat.io** を使用した認証 で説明されているように、**registry.redhat.io** との認証を設定している。
- Fuse Console のロールベースアクセス制御 (RBAC) をカスタマイズする場合は、Fuse Console Operator のインストール先と同じ OpenShift namespace に RBAC 設定マップファイルが必要になる。OpenShift 4.x の Fuse Console のロールベースアクセス制御 の説明にしたがって、デフォルトの RBAC 動作を使用する場合は、設定マップファイルを指定する必要はない。

手順

Fuse Console をインストールおよびデプロイするには、以下を行います。

1. Web ブラウザーで、**cluster admin** 権限を持つユーザーとして OpenShift コンソールにログインします。
2. **Operators** をクリックした後、**OperatorHub** をクリックします。
3. 検索フィールドウインドウに **Fuse Console** と入力し、Operator のリストを絞り込みます。
4. **Fuse Console Operator** をクリックします。
5. Fuse Console Operator インストールウィンドウで **Install** をクリックします。
   Create Operator Subscription フォームが表示されます。
   - **Update Channel** には **7.12.x** を選択します。
   - **Installation Mode** では、デフォルト (クラスターの特定の namespace) を受け入れます。operator のインストール後に Fuse Console をデプロイする場合、クラスター上のすべての namespace でアプリケーションを監視するか、Fuse Console operator がインストールされた namespace のみでアプリケーションを監視するかを選択できる点に注意してください。
   - **Installed Namespace** には、Fuse Console Operator をインストールする namespace を選択します。
   - **Approval Strategy** では、**Automatic** または **Manual** を選択し、Fuse Console の更新が OpenShift によってどのように対処されるかを設定します。
     - **Automatic** (自動) 更新を選択した場合、新しいバージョンの Fuse Console Operator が使用できるようになると、人的な介入なしで OpenShfit Operator Lifecycle Manager (OLM) によって、Fuse Console の稼働中のインスタンスが自動的にアップグレードされます。
     - **Manual** (手動) 更新を選択した場合、Operator の新しいバージョンが使用できるようになると、OLM によって更新リクエストが作成されます。クラスター管理者は、更新リクエストを手動で承認して、Fuse Console Operator を新しいバージョンに更新する必要があります。
6. **Install** をクリックします。
   OpenShift によって、Fuse Console Operator が現在の namespace にインストールされます。
7. インストールを確認するには、**Operators** をクリックした後、**Installed Operators** をクリックします。Operator のリストに Fuse Console が表示されます。
8. OpenShift Web コンソールで Fuse Console をデプロイするには、以下を行います。

a. **Installed Operators** のリストで、**Name** 列の **Fuse Console** をクリックします。

b. **Provided APIs** の **Operator Details** ページで、**Create Instance** をクリックします。
設定のデフォルト値を使用するか、任意でデフォルト値を編集します。

Fuse Console のパフォーマンスを向上する必要がある場合 (高可用性環境など)、**Replicas** で Fuse Console に割り当てられた Pod 数を増やすことができます。

Rbac (ロールベースアクセス制御) では、デフォルトの RBAC 動作をカスタマイズする場合や、Fuse Console Operator をインストールした namespace に ConfigMap ファイルがすでに存在する場合にのみ、**config Map** フィールドに値を指定します。RBAC の詳細は、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 を参照してください。

c. **Create** をクリックします。
Fuse Console Operator Details ページが開き、デプロイメントの状態が表示されます。

9. Fuse Console を開くには以下を行います。

a. **namespace** デプロイメントの場合: OpenShift Web コンソールで、Fuse Console の operator をインストールしたプロジェクトを開き、**Overview** を選択します。**Project Overview** ページで **Launcher** セクションまで下方向にスクロールし、Fuse Console の URL をクリックして開きます。
cluster デプロイメントでは、OpenShift Web コンソールのタイトルバーで、グリッドアイコン () をクリックします。ポップアップメニューの **Red Hat アプリケーション** の下にある、Fuse Console の URL リンクをクリックします。

b. Fuse Console にログインします。
ブラウザーに **Authorize Access** ページが表示され、必要なパーミッションが表示されます。

c. **Allow selected permissions** をクリックします。
Fuse Console がブラウザーで開き、アクセス権限のある Fuse アプリケーション Pod が表示されます。

10. 表示するアプリケーションの **Connect** をクリックします。
新しいブラウザーウィンドウが開かれ、Fuse Console にアプリケーションが表示されます。

1.2.2. コマンドラインを使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ

OpenShift 4.x では、次のデプロイメントオプションの 1 つを選択して、コマンドラインから Fuse Console をインストールおよびデプロイできます。

- **cluster** - Fuse Console は、OpenShift クラスターで複数の namespace (プロジェクト) 全体にデプロイされた Fuse アプリケーションを検索し、接続することができます。このテンプレートをデプロイするには、OpenShift クラスターの管理者ロールが必要です。

- **ロールベースアクセス制御のあるクラスター** - 設定可能なロールベースアクセス制御 (RBAC) のあるクラスターテンプレート。詳細は、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 を参照してください。

- **namespace** - Fuse Console は特定の OpenShift プロジェクト (namespace) にアクセスできます。このテンプレートをデプロイするには、OpenShift プロジェクトの管理者ロールが必要です。

- **ロールベースアクセス制御のある namespace** - 設定可能な RBAC のある namespace テンプレート。詳細は、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 を参照してください。

Fuse Console テンプレートのパラメーターリストを表示するには、以下の OpenShift コマンドを実行します。

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fuse-console-namespace-os4.json
```

前提条件

- Fuse Console をインストールおよびデプロイする前に、OpenShift 4.x で Fuse Console をセキュア化するための証明書の生成 の説明どおりに、サービス署名認証局で署名されたクライアント証明書を生成する必要があります。
- OpenShift クラスターの **cluster admin** ロールが必要です。
- コンテナーイメージの **registry.redhat.io** を使用した認証 で説明されているように、**registry.redhat.io** との認証を設定している。
- OpenShift 4.x サーバーでの Fuse イメージストリームおよびテンプレートのインストール の説明どおりに Fuse Console イメージストリーム (およびその他の Fuse イメージストリーム) がインストールされている必要があります。

手順

1. 以下のコマンドを実行してすべてのテンプレートのリストを取得し、Fuse Console イメージストリームがインストールされていることを確認します。

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

2. 任意で、すでにインストールされているイメージストリームを新しいリリースタグで更新する場合は、以下のコマンドを使用して **openshift** namespace に Fuse Console イメージをインポートします。

   ```
   oc import-image fuse7/fuse-console-rhel8:1.9 --from=registry.redhat.io/fuse7/fuse-console-rhel8:1.9 --confirm -n openshift
   ```

3. 以下のコマンドを実行して、Fuse Console の **APP_NAME** の値を取得します。

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

   ここで、**TEMPLATE-FILENAME** は以下のテンプレートのいずれかになります。

   - クラスターテンプレート:
     https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fuse-console-cluster-os4.json
   - 設定可能な RBAC のあるクラスターテンプレート:
     https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fuse-console-cluster-rbac.yml

- namespace テンプレート:
  https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fuse-console-namespace-os4.json

- 設定可能な RBAC のある namespace テンプレート:
  https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fuse-console-namespace-rbac.yml

  たとえば、設定可能な RBAC のあるクラスターテンプレートの場合は、以下のコマンドを実行します。

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fuse-console-cluster-rbac.yml
```

4. 以下のコマンドを使用して、OpenShift 4.x で Fuse Console をセキュア化するための証明書の生成 で生成した証明書からシークレットを作成し、Fuse Console にマウントします。コマンドの **APP_NAME** は、Fuse Console アプリケーションの名前に置き換えます。

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

5. 以下のコマンドを実行して、ローカルにコピーされた Fuse Console テンプレートをベースとした新しいアプリケーションを作成します。コマンドの **myproject** は OpenShift プロジェクトの名前、**mytemp** は Fuse Console テンプレートが含まれるローカルディレクトリーへのパス、**myhost** は Fuse Console にアクセスするホストの名前に置き換えます。

   - クラスターテンプレートの場合:

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

   - RBAC テンプレートのあるクラスターの場合:

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

   - Namespace テンプレートの場合:

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

   - RBAC テンプレートのある namespace の場合:

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

6. OpenShift Web コンソールを開くように Fuse Console を設定するには、以下のコマンドを実行して **OPENSHIFT_WEB_CONSOLE_URL** 環境変数を設定します。

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

7. 以下のコマンドを実行して、Fuse Console デプロイメントの状態と URL を取得します。

```
oc status
```

8. ブラウザーから Fuse Console にアクセスするには、手順 7. で返される URL (例: https://fuse-console.192.168.64.12.nip.io) を使用します。

1.2.2.1. OpenShift 4.x で Fuse Console をセキュア化するための証明書の生成

OpenShift 4.x で、Fuse Console プロキシーと Jolokia エージェントとの間の接続をセキュアにするには、Fuse Console をデプロイする前にクライアント証明書を生成する必要があります。サービス署名認証局の秘密鍵を使用して、クライアント証明書を署名する必要があります。

この手順は、コマンドラインを使用して Fuse Console をインストールおよびデプロイしている場合に**のみ** 実行する必要があります。Fuse Console Operator を使用している場合は、Fuse Console Operator がこのタスクを処理します。

重要

各 OpenShift クラスターに別のクライアント証明書を生成し、署名する必要があります。複数のクラスターに同じ証明書を使用しないでください。

前提条件

- OpenShift クラスターにアクセス可能な **cluster admin** 権限がある。
- 複数の OpenShift クラスターの証明書を生成し、現在のディレクトリーに別のクラスターの証明書を生成している場合は、以下のいずれかを実行して、現在のクラスターに別の証明書を生成するようにします。
  - 現在のディレクトリーから既存の証明書ファイル (**ca.crt**、**ca.key**、および **ca.srl**) を削除します。
  - 別の作業ディレクトリーに移動します。たとえば、現在の作業ディレクトリーの名前が **cluster1** の場合、新しい **cluster2** ディレクトリーを作成し、作業ディレクトリーをそのディレクトリーに変更します。
    mkdir ../cluster2

    cd ../cluster2

手順

1. cluster admin 権限を持つユーザーとして OpenShift にログインします。

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

2. 以下のコマンドを実行して、サービス署名認証局のキーを取得します。
   - 以下を実行して、証明書を取得します。

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

   - 以下を実行して、秘密鍵を取得します。

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

3. Kubernetes certificates administration の説明にしたがって、**easyrsa**、**openssl**、または **cfssl** を使用して、クライアント証明書を生成します。
以下に、openssl を使用したコマンドの例を示します。

   a. 秘密鍵を生成します。

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

   b. CSR 設定ファイルを作成します。

   ```
   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
   ```

   ここでは、**CN** パラメーターの値はアプリケーション名とアプリケーションが使用する namespace を参照します。

   c. CSR を生成します。

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

   d. 署名済みの証明書を発行します。

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

次のステップ

コマンドラインを使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ で説明されているように、Fuse Console のシークレットを作成するには、この証明書が必要です。

1.2.3. OpenShift 4.x 上の Fuse Console のロールベースアクセス制御

Fuse Console は、OpenShift によって提供されるユーザー承認に応じてアクセスを推測する、ロールベースアクセス制御 (RBAC) を提供します。Fuse Console では、RBAC はユーザーが Pod で MBean 操作を実行できるかどうかを判断します。

OpenShift の承認に関する詳細は、OpenShift ドキュメントの RBAC の使用によるパーミッションの定義および適用 を参照してください。

Operator を使用して OpenShift に Fuse Console をインストールした場合、ロールベースのアクセスはデフォルトで有効になります。

テンプレートでインストールして、Fuse Console のロールベースアクセスを実装する場合は、RBAC で設定可能なテンプレートの 1 つ (**fuse-console-cluster-rbac.yml** または **fuse-console-namespace-rbac.yml**) を使用して、Installing and deploying the Fuse Console on OpenShift 4.x by using the command line の説明どおりに Fuse Console をインストールする必要があります。

Fuse Console RBAC は、OpenShift の Pod リソースでユーザーの **verb (動詞)** アクセスを利用して、Fuse Console の Pod の MBean 操作にユーザーがアクセスできるかどうかを判断します。デフォルトでは、Fuse Console には 2 つのユーザーロールがあります。

- **admin**
  ユーザーが OpenShift で Pod を **更新** できる場合、ユーザーには Fuse Console の **admin** ロールが付与されます。ユーザーは、Fuse Console で、その Pod に対して **書き込み** の MBean 操作を実行できます。

- **viewer**
  ユーザーが OpenShift で Pod を **取得** できる場合、ユーザーには Fuse Console の **viewer** ロールが付与されます。ユーザーは、Fuse Console で、その Pod に対して **読み取り専用** の MBean 操作を実行できます。

注記

RBAC 以外のテンプレートを使用して Fuse Console をインストールした場合、Pod リソースで **update** verb (動詞) が付与された OpenShift ユーザーのみが Fuse Console の MBeans 操作の実行が許可されます。Pod リソースで **get** verb (動詞) を付与されたユーザーは、Pod を **表示** できますが、Fuse Console の操作は実行できません。

関連情報

- OpenShift 4.x での Fuse Console のアクセ出力ルの判断
- OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ
- OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

1.2.3.1. OpenShift 4.x での Fuse Console のアクセ出力ルの判断

Fuse Console のロールベースアクセス制御は、Pod に対するユーザーの OpenShift パーミッションから推測されます。特定のユーザーに付与された Fuse Console のアクセ出力ルを判断するには、Pod に対してユーザーに付与された OpenShift パーミッションを取得します。

前提条件

- ユーザーの名前を知っている必要があります。
- Pod の名前を知っている必要があります。

手順

- ユーザーが Pod に対して Fuse Console の admin ロールを持っているかどうかを確認するには、以下のコマンドを実行してユーザーが OpenShift で Pod を更新できるかどうかを確認します。

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

応答が **yes** の場合、ユーザーはその Pod に対して Fuse Console の **admin** ロールを持っています。ユーザーは、Fuse Console で、その Pod に対して **書き込み** の MBean 操作を実行できます。

- ユーザーが Pod に対して Fuse Console の **viewer** ロールを持っているかどうかを確認するには、以下のコマンドを実行してユーザーが OpenShift で Pod を**取得**できるかどうかを確認します。

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

応答が **yes** の場合、ユーザーはその Pod に対して Fuse Console の **viewer** ロールを持っています。ユーザーは、Fuse Console で、その Pod に対して **読み取り専用** の MBean 操作を実行できます。コンテキストによっては、Fuse Console でオプションを無効にしたり、ユーザーによる **書き込み** MBean 操作の試行時にこのユーザーの操作は許可されないという内容のメッセージを表示したりして、**viewer** ロールを持つユーザーによる **書き込み** MBean 操作が実行されないようにします。

応答が **no** の場合、ユーザーは Fuse Console のロールにバインドされず、ユーザーは Fuse Console で Pod を確認できません。

関連情報

- OpenShift 4.x 上の Fuse Console のロールベースアクセス制御
- OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ
- OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

1.2.3.2. OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ

OperatorHub を使用して Fuse Console をインストールした場合、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 の説明どおりに、ロールベースアクセス制御 (RBAC) はデフォルトで有効になります。Fuse Console の RBAC 動作をカスタマイズする場合、Fuse Console をデプロイする前に ConfigMap ファイル (カスタムの RBAC 動作を定義する) を指定する必要があります。カスタム ConfigMap ファイルは、Fuse Console Operator をインストールした namespace に配置する必要があります。

コマンドラインテンプレートを使用して Fuse Console をインストールする場合、**deployment-cluster-rbac.yml** および **deployment-namespace-rbac.yml** テンプレートによって、設定ファイル (**ACL.yml**) が含まれる ConfigMap が作成されます。設定ファイルでは、MBean 操作に許可されるロールが定義されます。

前提条件

- OperatorHub または Fuse Console の RBAC テンプレート (**deployment-cluster-rbac.yml** または **deployment-namespace-rbac.yml**) の 1 つを使用して、Fuse Console がインストール済みである必要があります。

手順

Fuse Console の RBAC ロールをカスタマイズする場合は、以下を行います。

1. コマンドラインを使用して Fuse Console をインストールした場合、インストールテンプレートにはデフォルトの ConfigMap ファイルが含まれるため、次のステップを省略できます。OperatorHub を使用して Fuse Console をインストールした場合、Fuse Console をデプロイする前に RBAC ConfigMap を作成します。

a. 現在の OpenShift プロジェクトが Fuse Console をインストールするプロジェクトであることを確認します。たとえば、Fuse Console を **fusetest** プロジェクトにインストールするには、以下のコマンドを実行します。

```
oc project fusetest
```

b. テンプレートから Fuse Console RBAC ConfigMap ファイルを作成するには、以下のコマンドを実行します。

```
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. 以下のコマンドを実行してエディターで ConfigMap を開きます。

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

以下に例を示します。

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

3. ファイルを編集します。

4. ファイルを保存して変更を適用します。Fuse Console の Pod は OpenShift によって自動的に再起動されます。

関連情報

- OpenShift 4.x 上の Fuse Console のロールベースアクセス制御
- OpenShift 4.x での Fuse Console のアクセ出力ルの判断
- OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

1.2.3.3. OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

コマンドラインを使用して Fuse Console をインストール し、Fuse Console の RBAC テンプレートのいずれかを指定した場合、Fuse Console の **HAWTIO_ONLINE_RBAC_ACL** 環境変数は、ロールベースアクセス制御 (RBAC) の ConfigMap 設定ファイルのパスを OpenShift サーバーに渡します。**HAWTIO_ONLINE_RBAC_ACL** 環境変数が指定されていない場合、RBAC のサポートは無効になり、Pod リソース (OpenShift の) で **update** verb (動詞) が付与されたユーザーのみが Fuse Console の Pod で MBean 操作を呼び出すことが承認されます。

OperatorHub を使用して Fuse Console をインストールする場合は、ロールベースのアクセスはデフォルトで有効になり、**HAWTIO_ONLINE_RBAC_ACL** 環境変数は適用されません。

前提条件

コマンドラインを使用して Fuse Console がインストール済みで、Fuse Console の RBAC テンプレート (**deployment-cluster-rbac.yml** または **deployment-namespace-rbac.yml**) のいずれかを指定している。

手順

Fuse Console のロールベースのアクセスを無効にするには、以下を実行します。

1. OpenShift で、Fuse Console の **Deployment Config** リソースを編集します。
2. **HAWTIO_ONLINE_RBAC_ACL** 環境変数の定義をすべて削除します。
(値を消去するだけでは不十分です)。
3. ファイルを保存して変更を適用します。Fuse Console の Pod は OpenShift によって自動的に再起動されます。

関連情報

- OpenShift 4.x 上の Fuse Console のロールベースアクセス制御
- OpenShift 4.x での Fuse Console のアクセ出力ルの判断
- OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ

1.2.4. OpenShift 4.x での Fuse Console のアップグレード

Red Hat OpenShift 4.x では、Red Hat Fuse Operator などの Operator の更新が処理されます。詳細は、OpenShift ドキュメントの Operator を参照してください。

その後、アプリケーションの設定方法によっては、Operator の更新でアプリケーションのアップグレードをトリガーできるようになります。

Fuse Console アプリケーションでは、アプリケーションカスタムリソース定義の **.spec.version** フィールドを編集して、アプリケーションのアップグレードをトリガーすることもできます。

前提条件

- OpenShift クラスターの管理者権限が必要です。

手順

Fuse Console アプリケーションをアップグレードするには、以下を行います。

1. ターミナルウィンドウで、以下のコマンドを使用してアプリケーションカスタムリソース定義の **.spec.version** フィールドを変更します。

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

以下に例を示します。

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

2. アプリケーションの状態が更新されたことを確認します。

```
 oc get myproject
```

応答には、バージョン番号などのアプリケーションに関する情報が表示されます。

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

.spec.version フィールドの値を変更すると、OpenShift によってアプリケーションが自動的に再デプロイされます。

3. バージョンの変更によってトリガーされた再デプロイの状態をチェックするには、以下を実行します。

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

正常にデプロイされた場合は以下の応答が表示されます。

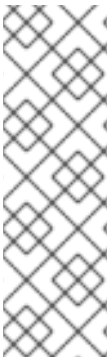
```
deployment "example-fuseconsole" successfully rolled out
```

1.3. OPENSHIFT 3.11 での FUSE CONSOLE の設定

OpenShift 3.11 では、以下のように Fuse Console にアクセスできます。

- プロジェクトで実行しているすべての Fuse コンテナーを監視できるように、Fuse Console を OpenShift プロジェクトに追加する。
- クラスター上のすべてのプロジェクトで稼働中のすべての Fuse コンテナーを監視できるように、Fuse Console を OpenShift クラスターに追加する。
- 実行している単一の Fuse コンテナーを監視できるように、特定の Fuse Pod から Fuse Console を開く。

コマンドラインから Fuse Console テンプレートをデプロイします。



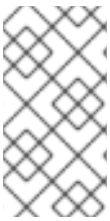
注記

Minishift または CDK ベースの環境変数に Fuse Console をインストールするには、以下の KCS の記事で説明されている手順に従います。

- Minishift または CDK ベースの環境に Fuse Console をインストールするには、KCS 4998441 を参照してください。
- Jolokia 認証を無効にする必要がある場合は、KCS 3988671 で説明されている回避策を参照してください。

前提条件

- Fuse on OpenShift ガイド の説明にしたがって、Fuse Console の Fuse on OpenShift イメージストリームおよびテンプレートをインストールする必要があります。



注記

- Fuse Console のユーザー管理は、OpenShift によって処理されます。
- ロールベースアクセス制御 (デプロイ後に Fuse Console にアクセスするユーザーの場合) は現在 Fuse on OpenShift 3.11 では使用できません。

「OpenShift 3.11 での Fuse Console のデプロイ」

「OpenShift 3.11 の Fuse Console から単一の Fuse Pod を監視」

1.3.1. OpenShift 3.11 での Fuse Console のデプロイ

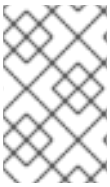
表1.1「Fuse Console テンプレート」は、Fuse アプリケーションのデプロイメントのタイプに応じて、コマンドラインから Fuse Console をデプロイするために使用する OpenShift 3.11 テンプレートについて説明しています。

表1.1 Fuse Console テンプレート

タイプ	説明
fis-console-cluster-template.json	Fuse Console は、複数の namespace またはプロジェクトにまたがってデプロイされた Fuse アプリケーションを検出し、接続することができます。このテンプレートをデプロイするには、OpenShift の cluster-admin ロールが必要です。
fis-console-namespace-template.json	このテンプレートは、Fuse Console の現在の OpenShift プロジェクト (namespace) へのアクセスを制限するため、単一のテナントデプロイメントとして動作します。このテンプレートをデプロイするには、現在の OpenShift プロジェクトの admin ロールが必要です。

任意で以下のコマンドを実行すると、すべてのテンプレートのパラメーターの一覧を表示できます。

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fis-console-namespace-template.json
```



注記

Fuse Console のテンプレートは、デフォルトでエンドツーエンド暗号化を設定するため、Fuse Console のリクエストはブラウザーからクラスター内のサービスまでエンドツーエンドでセキュア化されます。

前提条件

- OpenShift 3.11 のクラスターモードでは、cluster admin ロールとクラスターモードテンプレートが必要です。以下のコマンドを実行します。

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

手順

コマンドラインから Fuse Console をデプロイするには、以下を行います。

1. 以下のコマンドの 1 つを実行して、Fuse Console テンプレートをベースとした新しいアプリケーションを作成します。コマンドの **myproject** はプロジェクトの名前に置き換えます。
   - Fuse Console の **cluster** テンプレートの場合は、以下のようになります。**myhost** は Fuse Console にアクセスするホストの名前になります。

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fis-console-cluster-template.json -p ROUTE_HOSTNAME=myhost
```

- Fuse Console の **namespace** テンプレートの場合は以下のようになります。

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-790047-redhat-00005/fis-console-namespace-template.json
```

注記

namespace テンプレートの route_hostname パラメーターは OpenShift によって自動的に生成されるため、省略することが可能です。

2. 以下のコマンドを実行して、Fuse Console デプロイメントの状態と URL を取得します。

```
oc status
```

3. ブラウザーから Fuse Console にアクセスするには、提供される URL (例: https://fuse-console.192.168.64.12.nip.io) を使用します。

1.3.2. OpenShift 3.11 の Fuse Console から単一の Fuse Pod を監視

OpenShift 3.11 で実行している Fuse Pod の Fuse Console を開きます。

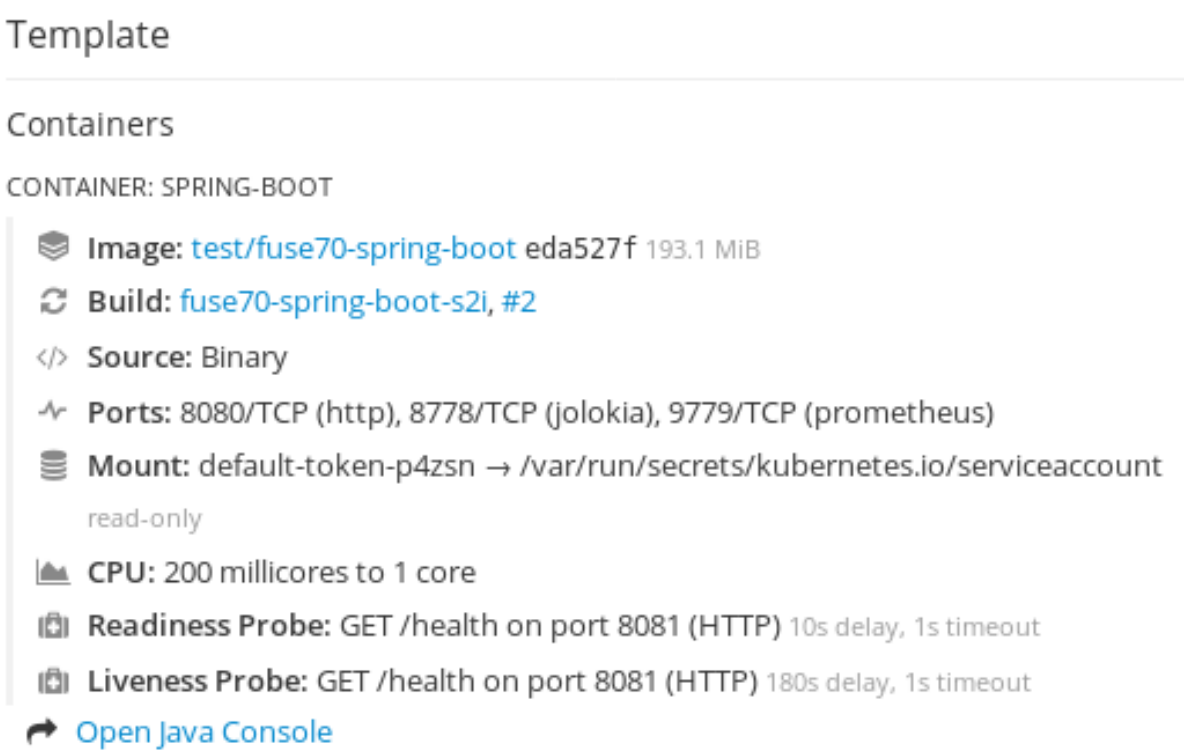
前提条件

- Pod ビューで Fuse Console へのリンクを表示するよう OpenShift を設定するには、Fuse on OpenShift イメージを実行している Pod が **jolokia** に設定された name 属性内で TCP ポートを宣言する必要があります。

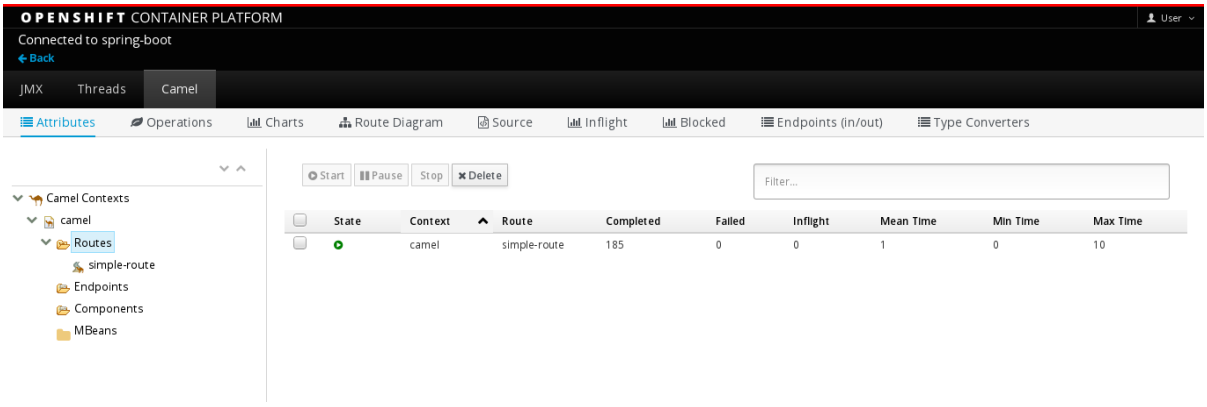
```
{
  "kind": "Pod",
  [...]
  "spec": {
    "containers": [
      {
        [...]
        "ports": [
          {
            "name": "jolokia",
            "containerPort": 8778,
            "protocol": "TCP"
          }
```

手順

1. OpenShift プロジェクトの **Applications → Pods** ビューで、Pod 名をクリックし、実行している Fuse Pod の詳細を表示します。このページの右側に、コンテナーテンプレートの概要が表示されます。



2. このビューの **Open Java Console** リンクをクリックし、Fuse Console を開きます。



1.4. コンテナーおよびアプリケーションの表示

OpenShift の Fuse Console にログインすると、Fuse Console のホームページに利用可能なコンテナーが表示されます。

手順

- コンテナーを管理 (作成、編集、または削除) するには、OpenShift コンソールを使用します。
- OpenShift クラスターで Fuse アプリケーションと AMQ ブローカー (該当する場合) を確認するには、**Online** タブをクリックします。

1.5. APACHE CAMEL アプリケーションの表示および管理

Fuse Console の **Camel** タブで Apache Camel のコンテキスト、ルート、および依存関係を表示および管理します。

次の詳細を表示できます。

- 実行中の Camel コンテキストすべてのリスト。
- Camel バージョン番号やランタイム統計など、各 Camel コンテキストの詳細情報。
- 各 Camel アプリケーションの全ルートおよびランタイム統計のリスト。
- 実行中のルートとリアルタイムのメトリクスのグラフィック表示。

また、以下を行うと Camel アプリケーションと対話もできます。

- コンテキストの開始および一時停止。
- 再起動、停止、一時停止、再開などを実行できるよう、すべての Camel アプリケーションとそれらのルートのライフサイクルを管理。
- 実行中のルートのライブトレースおよびデバッグ。
- Camel エンドポイントへのメッセージの閲覧および送信。

前提条件

Camel タブは、1 つ以上の Camel ルートを使用するコンテナーに接続する場合のみ使用できます。

1.5.1. コンテキストの開始、一時停止、または削除

1. **Camel** タブのツリービューで、**Camel Contexts** をクリックします。
2. リストのコンテキストの横にあるボックスにチェックマークを入れます。
3. **Start** または **Suspend** をクリックします。
4. コンテキストを削除するには以下を行います。
   a. コンテキストを停止します。
   b. 楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。

注記

コンテキストを削除する場合、デプロイされたアプリケーションから削除します。

1.5.2. Camel アプリケーションの詳細表示

1. **Camel** タブのツリービューで、Camel アプリケーションをクリックします。
2. アプリケーションの属性と値のリストを表示するには、**Attributes** をクリックします。
3. アプリケーション属性をグラフィックに表示するには、**Chart** をクリックした後、**Edit** をクリックし、チャートに表示する属性を選択します。
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. アプリケーションエンドポイントを表示するには、**Endpoints** をクリックします。リストは **URL**、**Route ID**、および **direction** で絞り込むことができます。

6. メッセージ本文とメッセージヘッダーを別のタイプに変換するために使用される Camel 組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化するには、**Type Converters** をクリックします。

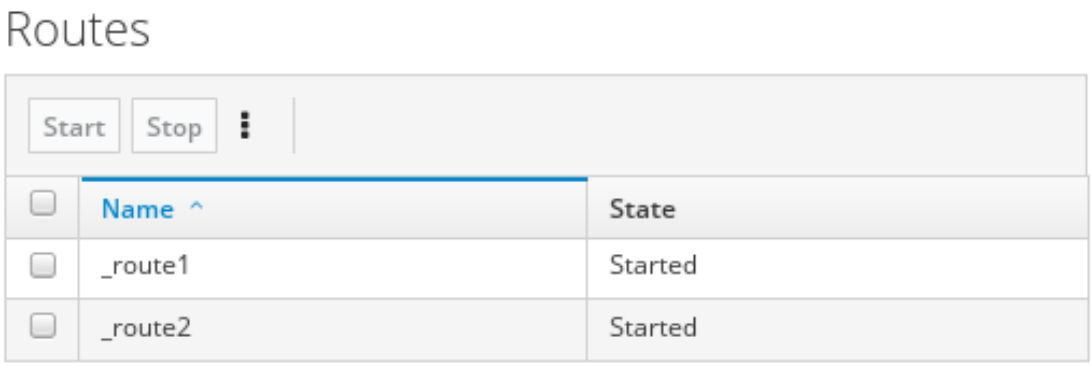
7. JMX 操作 (XML からのルートの追加または更新、クラスパスで利用できる Camel コンポーネントの検索など) を表示および実行するには、**Operations** をクリックします。

1.5.3. Camel ルートリストの表示および Camel ルートとの対話

1. ルートのリストを表示するには、以下を行います。

   a. **Camel** タブをクリックします。

   b. ツリービューでアプリケーションの routes フォルダーをクリックします。

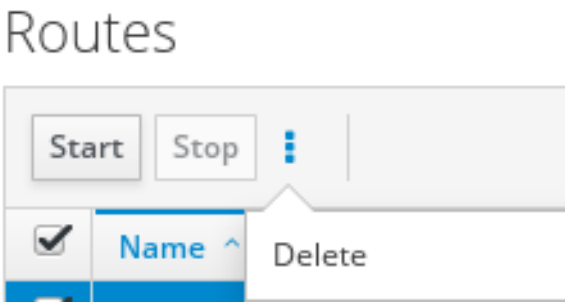


2. 1 つまたは複数のルートを開始、停止、または削除するには、以下を行います。

   a. リストのルートの横にあるボックスにチェックマークを入れます。

   b. **Start** または **Stop** をクリックします。

   c. 最初にルートを停止してから削除する必要があります。停止したら楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



注記

- ルートを削除する場合、デプロイされたアプリケーションから削除します。
- ツリービューで特定のルートを選択し、右上のメニューをクリックして開始、停止、または削除することもできます。

3. ルートのグラフィックな図を表示するには、**Route Diagram** をクリックします。

4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。

5. エンドポイントを表示するには、**Endpoints** をクリックします。URL、Route ID、および方向でリストを絞り込むことができます。

6. **Type Converters** をクリックし、Camel の組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化します。このメカニズムはメッセージ本文とメッセージヘッダーを別のタイプに変換するために使用されます。

7. 特定のルートと対話するには、以下を行います。

   a. **Camel** タブのツリービューで、ルートを選択します。

   b. ルート属性と値のリストを表示するには、**Attributes** をクリックします。

   c. ルート属性をグラフィックに表示するには、**Chart** をクリックします。**Edit** をクリックするとチャートに表示する属性を選択することができます。

   d. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。

   e. **Operations** をクリックして JMX 操作 (ルートを XML としてダンプ、ルートの Camel ID 値の取得など) を表示および実行できます。

8. ルートを介してメッセージをトレースするには、以下を実行します。

   a. **Camel** タブのツリービューで、ルートを選択します。

   b. **Trace** を選択し、**Start tracing** をクリックします。

9. メッセージをルートに送信するには、以下を行います。

   a. **Camel** タブのツリービューでコンテキストのエンドポイントフォルダーを開き、エンドポイントを選択します。

   b. **Send** サブタブをクリックします。

   c. JSON または XML 形式のメッセージを設定します。

   d. **Send** をクリックします。

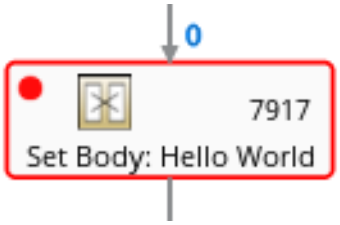
   e. ルートの **Trace** タブに戻り、ルートを介したメッセージのフローを確認します。

1.5.4. ルートのデバッグ

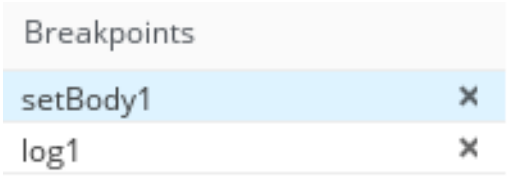
1. **Camel** タブのツリービューで、ルートを選択します。

2. **Debug** を選択し、**Start debugging** をクリックします。

3. ブレークポイントを追加するには、図のノードを選択し、**Add breakpoint** をクリックします。ノードに赤い点が表示されます。



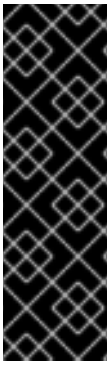
   ノードがブレークポイントのリストに追加されます。



4. 下矢印をクリックして次のノードに移動するか、**Play** ボタンをクリックしてルートの実行を再開します。
5. **Pause** ボタンをクリックして、ルートのすべてのスレッドを一時停止します。
6. 終了したら **Stop debugging** をクリックします。すべてのブレークポイントが消去されます。

1.6. AMQ ブローカーの表示

Fuse Console を設定して、OpenShift クラスターにデプロイされた AMQ ブローカーを表示できます。



重要

Fuse Console から AMQ ブローカーを表示することは、テクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat の本番環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあるため、Red Hat は本番環境での使用は推奨しません。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能では、最新の製品機能をいち早く提供します。これにより、お客様は開発段階で機能をテストし、フィードバックを提供できます。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、https://access.redhat.com/ja/support/offerings/techpreview を参照してください。

前提条件

各 AMQ ブローカーイメージ (Fuse Console で表示する) は以下である必要があります。

- Fuse Console がインストールされているのと同じ OpenShift クラスターにインストールする必要があります。
- Deploying AMQ Broker on OpenShift の Fuse Console で Artemis プラグインを有効にする 説明にあるとおり、Fuse Console が認識および接続するように 各 AMQ ブローカーイメージを設定する必要があります。

手順

- **Artemis** をクリックして AMQ 管理コンソールを表示し、AMQ ブローカーの状態を監視します。AMQ ブローカーは Apache ActiveMQ Artemis を基にしています。

AMQ 管理コンソールの使用に関する詳細は、**Managing AMQ Broker**の 2 章 Using AMQ Management Consoleを参照してください。

1.7. JMX ドメインおよび MBEAN の表示および管理

JMX (Java Management Extensions) は、実行時にリソース (サービス、デバイス、およびアプリケーション) を動的に管理できる Java 技術です。リソースは MBean (Managed Bean) と呼ばれるオブジェクトで表現されます。リソースが作成、実装、またはインストールされると即座に管理することができます。

Fuse Console で JMX プラグインを使用すると、JMX ドメインと MBean を表示および管理できます。MBean 属性の表示、コマンドの実行、および MBean の統計を示すチャートの作成を行うことができます。

JMX タブは、フォルダーに整理されたアクティブな JMX ドメインと MBean のツリービューを提供します。詳細を確認し、MBean でコマンドを実行できます。

手順

1. MBean 属性を表示および編集するには、以下を行います。
   a. ツリービューで MBean を選択します。
   b. **Attributes** タブをクリックします。
   c. 属性をクリックしてその詳細を表示します。
2. 操作を実行するには、以下を行います。
   a. ツリービューで MBean を選択します。
   b. **Operations** タブをクリックし、リストにある操作の 1 つを展開します。
   c. **Execute** をクリックし、操作を実行します。
3. チャートを表示するには、以下を行います。
   a. ツリービューで項目を選択します。
   b. **Chart** タブをクリックします。

1.8. QUARTZ スケジュールの表示および管理

Quartz(http://www.quartz-scheduler.org/) は、ほとんどの Java アプリケーション内で統合できる、機能豊富なオープンソースジョブスケジューリングライブラリーです。Quartz を使用して、ジョブを実行するための単純または複雑なスケジュールを作成できます。ジョブは、プログラムするほとんどすべてのものを実行できる標準の Java コンポーネントとして定義されます。

Camel ルートが **camel-quartz2** コンポーネントをデプロイすると、Fuse Console には **Quartz** タブが表示されます。JMX ツリービューから Quartz mbeans に交互にアクセスできることに注意してください。

手順

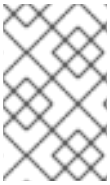
1. Fuse Console で **Quartz** タブをクリックします。
   Quartz ページには、Quartz スケジューラーのツリービューならびに **Scheduler**、**Triggers**、および **Jobs** タブが含まれます。
2. スケジューラーを一時停止または開始するには、**Scheduler** タブのボタンをクリックします。
3. **Triggers** タブをクリックして、ジョブを実行するタイミングを決定するトリガーを表示します。たとえば、トリガーは、一日の指定の時刻 (ミリ秒単位) に、指定した日数、または指定した回数もしくは特定の回数繰り返してジョブを開始するように指定できます。
   - トリガーの一覧をフィルターリングするには、ドロップダウンリストから **State**、**Group**、**Name**、または **Type** を選択します。続いて、入力フィールドに選択または入力することで、さらに一覧をフィルターできます。

- トリガーを一時停止、再開、更新、または手動で実行するには、**Action** 列のオプションをクリックします。

4. **Jobs** タブをクリックして実行中のジョブの一覧を表示します。テーブルの**Group**、**Name**、**Durable**、**Recover**、**Job ClassName**、および**Description** の列でリストをソートできます。

1.9. 診断の表示

Diagnostics タブを使用して、JVM DiagnosticCommand および HotspotDiangostic インターフェイスから JVM に関する診断情報を表示します。



注記

この機能は、Java Mission Control (jmc) の **Diagnostic Commands** ビューや、コマンドラインツールの jcmd と似ています。場合によっては、プラグインが対応する jcmd コマンドを提供します。

手順

1. ロードされたクラスのインスタンス数や、これらのインスタンスが使用するバイト数を取得するには、**Class Histogram** をクリックします。操作が繰り返し行われると、最後の操作実行との差異がタブに表示されます。

2. JVM 診断フラグ設定を表示するには、**JVM flags** をクリックします。

3. 稼働中の JVM でもフラグ設定を変更できます。

関連情報

サポートされる JVM はプラットフォームによって異なります。詳細は以下を参照してください。

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

1.10. スレッドの表示

スレッドの状態を表示および監視できます。

手順

1. **Runtime** タブをクリックし、**Threads** サブタブをクリックします。**Threads** ページには、アクティブなスレッドと各スレッドのスタックトレースの詳細が表示されます。デフォルトでは、スレッドリストにはすべてのスレッドが ID 値が大きい順に表示されます。

2. ID 値が小さい順に表示するには、**ID** 列ラベルをクリックします。

3. 任意で、スレッドの状態 (例: **Blocked**) やスレッド名でリストを絞り込むことができます。

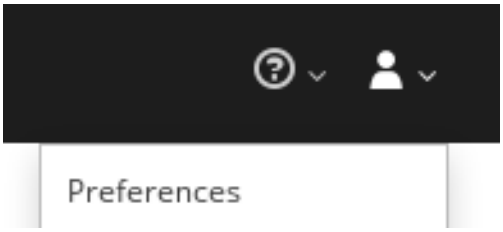
4. ロッククラス名やスレッドのフルスタックトレースなど、特定スレッドの詳細情報を表示するには、**Actions** 列で **More** をクリックします。

1.11. FUSE CONSOLE でデータが正しく表示されるよう確認

Fuse Console のキューおよびコネクションの表示で、不足しているキューやコネクションがあったり、一貫性のないアイコンが表示される場合、Jolokia が応答でマーシャルするアレイの要素の最大数を指定する、Jolokia コレクションサイズパラメーターを調節します。

手順

1. Fuse Console の右上隅にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. **Maximum collection size** オプションの値を大きくします (デフォルトは 50,000)。
3. **Close** をクリックします。

第2章 SPRING BOOT スタンドアロンでの RED HAT FUSE アプリケーションの監視および管理

2.1. FUSE CONSOLE

Red Hat Fuse Console は、HawtIO オープンソースソフトウェアをベースとする Web コンソールです。サポートされるブラウザーのリストは Red Hat Fuse でサポートされる設定 を参照してください。

Fuse Console は、デプロイされた 1 つ以上の Fuse コンテナーの詳細を確認および管理する中央インターフェイスを提供します。また、Red Hat Fuse およびシステムリソースの監視、更新の実行、およびサービスの開始と停止を行うこともできます。

Fuse Console は、Red Hat Fuse スタンドアロンをインストールしたり Fuse on OpenShift の使用すると利用することができます。Fuse Console で表示および管理できるインテグレーションは、実行されているプラグインによって異なります。使用できるプラグインには以下が含まれます。

- Camel
- JMX
- OSGI
- Runtime
- Logs

2.2. SPRING BOOT 2.X の FUSE CONSOLE へのアクセス

スタンドアロン Fuse Spring Boot 2.x ディストリビューションの Fuse Console にアクセスできます。

手順

1. 以下の依存関係を Fuse アプリケーションの **pom.xml** ファイルに追加します。

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

   バージョンは Maven BOM によって提供されるため、指定する必要はありません。

2. **src/main/resources/application.properties** ファイルを編集します。
   a. 以下のプロパティーを設定します。
      - **management.endpoints.web.exposure.include=hawtio,jolokia**
      - **hawtio.authenticationEnabled=false**
      - **management.endpoint.hawtio.enabled=true**
      - **management.endpoint.jolokia.enabled=true**
   b. 任意で、**management.endpoints.web.base-path** プロパティーを設定します。

Spring Boot 2.x のデフォルトでは、Fuse Console の URL に管理エンドポイントのコンテキストパス (/**actuator**) が含まれます。以下に例を示します。

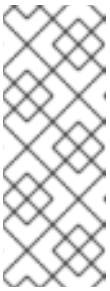
http://localhost:10001/actuator/hawtio/index.html

このデフォルト URL を変更し、**http://localhost:10001/hawtio** を指定するには、**management.endpoints.web.base-path** プロパティーを以下のように設定します。

management.endpoints.web.base-path=/

application.properties 設定は以下の例のようになるはずです。

```
# 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=/
```



注記

デフォルトでは、Spring Boot の Fuse Console の認証は無効になっています。任意で、Fuse Console ディストリビューションに固有するコードを作成すると、認証を有効にすることができます。例は次のとおりです。

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

3. Fuse アプリケーションを実行します。

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

4. Fuse Console の URL のポート番号を特定するには、**src/main/resources/application.properties** ファイルで **management.server.port** の設定値を見つけます。以下に例を示します。

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

5. ブラウザーで Fuse Console を開くには、以下の URL 構文を使用します。**nnnnn** は **management.server.port** プロパティーの値に置き換えます。
http://localhost:nnnnn/actuator/hawtio

   たとえば、**management.server.port** プロパティーの値が **10001** で、**management.endpoints.web.base-path** プロパティーが設定されていない場合、URL は以下になります。

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

2.3. FUSE CONSOLE ブランディングのカスタマイズ

hawtconfig.json ファイルを Fuse on Spring Boot スタンドアロンアプリケーションに追加すると、タイトル、ロゴ、ログインページの情報などの Fuse Console ブランディング情報をカスタマイズできます。

手順

1. ローカルの Fuse on Spring Boot スタンドアロンアプリケーションの **src/main/webapp** ディレクトリーに **hawtconfig.json** という名前の JSON ファイルを作成します。

2. 任意のエディターで **src/main/webapp/hawtconfig.json** を開き、以下の内容を追加します。

```
{
  "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. 表A.1「Fuse Console 設定プロパティー」 に記載されている設定プロパティーの値を変更します。

4. 変更を保存します。

5. 以下のコマンドを使用して Fuse on Spring Boot を実行します。

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

6. Web ブラウザーで、URL **http://localhost:10001/actuator/hawtio/index.html** を使用して、Fuse Console を開きます。

注記

Web ブラウザーで Fuse Console をすでに実行した場合、ブランディングはブラウザーのローカルストレージに保存されます。新しいブランディング設定を使用するには、ブラウザーのローカルストレージをクリアする必要があります。

2.4. FUSE CONSOLE のセキュア化

Spring Boot で Fuse Console をセキュアにするには、以下を行います。

- **AWS へのデプロイ時に Fuse Console のプロキシーサーブレットを無効化**
  スタンドアロン Fuse アプリケーションを Amazon Web Services (AWS) にデプロイする場合、**hawtio.disableProxy** システムプロパティーを **true** に設定して、Fuse Console のプロキシーサーブレットを無効にする必要があります。

注記

Fuse Console のプロキシーサーブレットを無効にすると、Fuse Console の **Connect** タブが無効になり、Fuse Console から他の JVM に接続できなくなります。AWS に複数の Fuse アプリケーションをデプロイする場合は、アプリケーションごとに Fuse Console をデプロイする必要があります。

- **必要なプロトコルとして HTTPS を設定する**
  hawtio.http.strictTransportSecurity プロパティーを使用すると、Web ブラウザーでセキュア HTTPS プロトコルを使用して Fuse Console にアクセスするよう要求できます。このプロパティーでは、HTTP を使用して Fuse Console へのアクセスを試みる Web ブラウザーは、要求を自動的に変換して HTTPS を使用する必要があることが指定されます。

- **公開鍵を使用して応答をセキュアにする**
  特定の暗号の公開鍵を Fuse Console に関連付けし、偽造された証明書を使用した中間者攻撃のリスクを軽減するよう Web ブラウザーに要求すると、**hawtio.http.publicKeyPins** プロパティーを使用して HTTPS プロトコルをセキュアにすることができます。

手順

1. **hawtio.http.strictTransportSecurity** および **hawtio.http.publicKeyPins** プロパティーを以下の例のように設定します。

```
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. (AWS のみにデプロイする場合) Fuse Console のプロキシーサーブレットを無効にするには、以下の例のように **hawtio.disableProxy** プロキシーを設定します。

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

関連情報

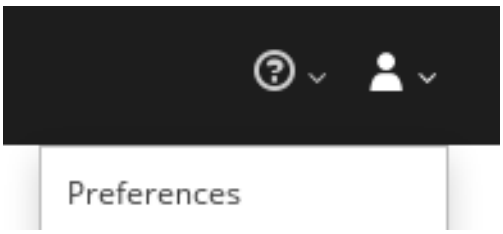
- **hawtio.http.strictTransportSecurity** プロパティーの構文の説明は、HTTP Strict Transport Security (HSTS) 応答ヘッダーの説明ページを参照してください。
- **hawtio.http.publicKeyPins** プロパティーの構文や、Base64 でエンコードされた公開鍵の抽出方法の説明は、HTTP Public Key Pinning 応答ヘッダーの説明ページを参照してください。

2.5. FUSE CONSOLE でデータが正しく表示されるよう確認

Fuse Console のキューおよびコネクションの表示で、不足しているキューやコネクションがあったり、一貫性のないアイコンが表示される場合、Jolokia が応答でマーシャルするアレイの要素の最大数を指定する、Jolokia コレクションサイズパラメーターを調節します。

手順

1. Fuse Console の右上隅にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. **Maximum collection size** オプションの値を大きくします (デフォルトは 50,000)。
3. **Close** をクリックします。

2.6. リモート FUSE アプリケーションへの接続

Fuse Console は Jolokia を使用します。Jolokia は、クライアントに追加のソフトウェア (エージェント) をインストールする必要がある Java Management Extensions (JMX) にエージェントベースで対応します。Red Hat Fuse には jolokia エージェントがデフォルトで含まれています。

スタンドアロン Fuse Console ディストリビューションでは、内部ですでに jolokia エージェント (https://jolokia.org/) が実行されているリモートインテグレーションに接続することができます。接続するプロセス内に jolokia エージェントがない場合は、jolokia のドキュメント (http://jolokia.org/agent.html) を参照してください。

手順

Fuse Console のプロキシーサーブレットはホワイトリストを使ってホストを保護し、Fuse Console はデフォルトではローカルホストのみに接続できます。Fuse Console を他のリモート Fuse インスタンスに接続する場合は、Spring Boot アプリケーションの **main()** メソッドにある **hawtio.proxyWhitelist** プロパティーを以下のように設定する必要があります。

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

2.6.1. リモート Jolokia エージェントへの接続

作業を開始する前に、リモート Jolokia エージェントの接続詳細 (ホスト名、ポート、およびパス) を知っておく必要があります。

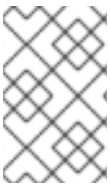
Spring Boot の Jolokia エージェントのデフォルト接続 URL は **http://<host>:8080/jolokia** です。

システム管理者はこのデフォルト設定を変更できます。

通常、Jolokia エージェントにリモートで接続する URL は、Fuse Console を開く URL に **/jolokia** を追加したものです。たとえば、Fuse Console を開く URL が **http://<host>:1234/hawtio** の場合、リモート接続の URL は **http://<host>:1234/hawtio/jolokia** になります。

JVM を確認するためにリモート Jolokia インスタンスに接続するには、以下を行います。

1. **Connect** タブをクリックします。
2. **Remote** タブをクリックし、**Add connection** をクリックします。
3. **Name**、**Scheme** (HTTP または HTTPS)、および **hostname** を入力します。
4. **Test Connection** をクリックします。
5. **Add** をクリックします。



注記

Fuse Console は自動的にローカルホストと 127.0.0.1 以外のローカルネットワークインターフェイスをプローブし、ホワイトリストに追加します。そのため、ローカルマシンのアドレスを手作業でホワイトリストに登録する必要はありません。

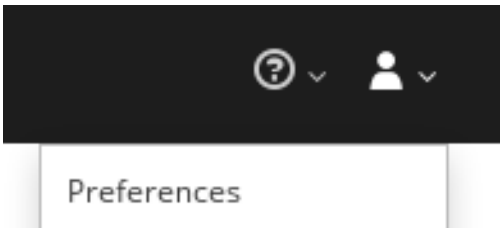
2.6.2. データ移動設定の指定

Fuse Console に表示されるデータをより頻繁にリフレッシュする場合などに、以下の Jolokia 設定を変更することができます。データの更新を頻繁に行うと、ネットワークトラフィックに影響し、サーバーに対するリクエストの数が増加するため注意してください。

- **Update rate** - JMX データを取得するため Jolokia へポーリングを行う間隔 (デフォルトは 5 秒)。
- **Maximum depth** - 戻る前にサーバー側で Jolokia がオブジェクトを JSON にマーシャルするレベル数 (デフォルトは 7)。
- **Maximum collection size** - 応答で Jolokia がマーシャルするアレイの最大要素数 (デフォルトは 50,000)。

これらの設定の値を変更するには、以下を行います。

1. Fuse Console の右上にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. オプションを編集して **Close** をクリックします。

2.6.3. JVM ランタイム情報の表示

システムプロパティー、メトリクス、スレッドなどの JVM のランタイム情報を表示するには、**Runtime** タブをクリックします。

2.7. APACHE CAMEL アプリケーションの表示および管理

Fuse Console の **Camel** タブで Apache Camel のコンテキスト、ルート、および依存関係を表示および管理します。

次の詳細を表示できます。

- 実行中の Camel コンテキストすべてのリスト。
- Camel バージョン番号やランタイム統計など、各 Camel コンテキストの詳細情報。
- 各 Camel アプリケーションの全ルートおよびランタイム統計のリスト。
- 実行中のルートとリアルタイムのメトリクスのグラフィック表示。

また、以下を行うと Camel アプリケーションと対話もできます。

- コンテキストの開始および一時停止。
- 再起動、停止、一時停止、再開などを実行できるよう、すべての Camel アプリケーションとそれらのルートのライフサイクルを管理。
- 実行中のルートのライブトレースおよびデバッグ。
- Camel エンドポイントへのメッセージの閲覧および送信。

前提条件

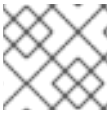
Camel タブは、1 つ以上の Camel ルートを使用するコンテナーに接続する場合のみ使用できます。

2.7.1. コンテキストの開始、一時停止、または削除

1. **Camel** タブのツリービューで、**Camel Contexts** をクリックします。
2. リストのコンテキストの横にあるボックスにチェックマークを入れます。
3. **Start** または **Suspend** をクリックします。
4. コンテキストを削除するには以下を行います。

a. コンテキストを停止します。

b. 楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



注記

コンテキストを削除する場合、デプロイされたアプリケーションから削除します。

2.7.2. Camel アプリケーションの詳細表示

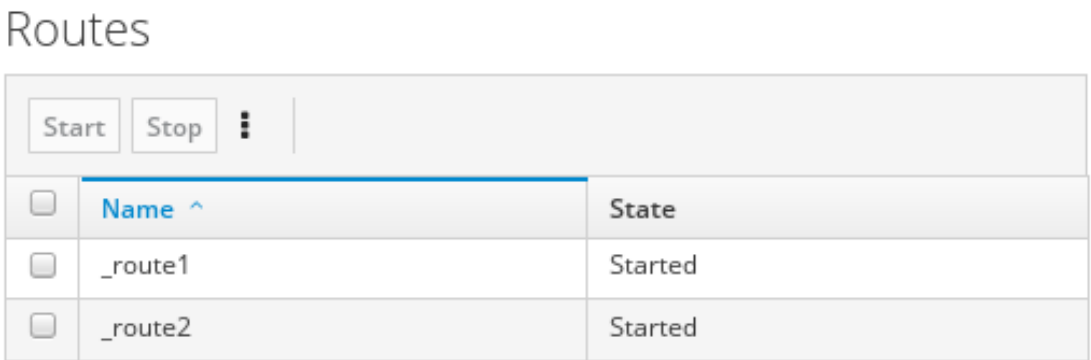
1. **Camel** タブのツリービューで、Camel アプリケーションをクリックします。
2. アプリケーションの属性と値のリストを表示するには、**Attributes** をクリックします。
3. アプリケーション属性をグラフィックに表示するには、**Chart** をクリックした後、**Edit** をクリックし、チャートに表示する属性を選択します。
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. アプリケーションエンドポイントを表示するには、**Endpoints** をクリックします。リストは **URL**、**Route ID**、および **direction** で絞り込むことができます。
6. メッセージ本文とメッセージヘッダーを別のタイプに変換するために使用される Camel 組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化するには、**Type Converters** をクリックします。
7. JMX 操作 (XML からのルートの追加または更新、クラスパスで利用できる Camel コンポーネントの検索など) を表示および実行するには、**Operations** をクリックします。

2.7.3. Camel ルートリストの表示および Camel ルートとの対話

1. ルートのリストを表示するには、以下を行います。

   a. **Camel** タブをクリックします。

   b. ツリービューでアプリケーションの routes フォルダーをクリックします。

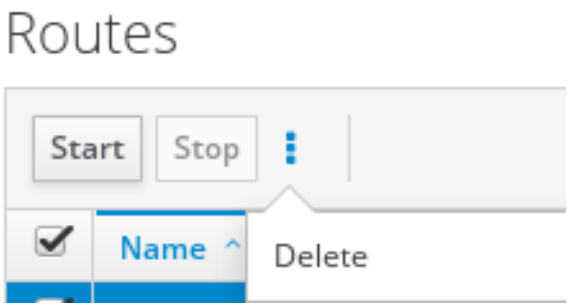


2. 1 つまたは複数のルートを開始、停止、または削除するには、以下を行います。

   a. リストのルートの横にあるボックスにチェックマークを入れます。

   b. **Start** または **Stop** をクリックします。

   c. 最初にルートを停止してから削除する必要があります。停止したら楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



注記

- ルートを削除する場合、デプロイされたアプリケーションから削除します。
- ツリービューで特定のルートを選択し、右上のメニューをクリックして開始、停止、または削除することもできます。

3. ルートのグラフィックな図を表示するには、**Route Diagram** をクリックします。
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. エンドポイントを表示するには、**Endpoints** をクリックします。URL、Route ID、および方向でリストを絞り込むことができます。
6. **Type Converters** をクリックし、Camel の組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化します。このメカニズムはメッセージ本文とメッセージヘッダーを別のタイプに変換するために使用されます。
7. 特定のルートと対話するには、以下を行います。
    a. **Camel** タブのツリービューで、ルートを選択します。
    b. ルート属性と値のリストを表示するには、**Attributes** をクリックします。
    c. ルート属性をグラフィックに表示するには、**Chart** をクリックします。**Edit** をクリックするとチャートに表示する属性を選択することができます。
    d. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
    e. **Operations** をクリックして JMX 操作 (ルートを XML としてダンプ、ルートの Camel ID 値の取得など) を表示および実行できます。
8. ルートを介してメッセージをトレースするには、以下を実行します。
    a. **Camel** タブのツリービューで、ルートを選択します。
    b. **Trace** を選択し、**Start tracing** をクリックします。
9. メッセージをルートに送信するには、以下を行います。
    a. **Camel** タブのツリービューでコンテキストのエンドポイントフォルダーを開き、エンドポイントを選択します。
    b. **Send** サブタブをクリックします。

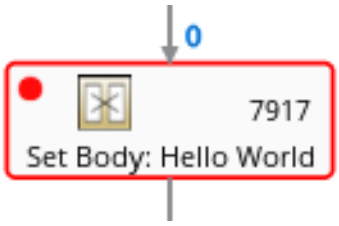
c. JSON または XML 形式のメッセージを設定します。

d. **Send** をクリックします。

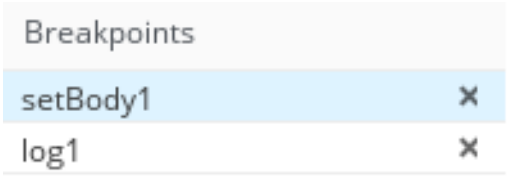
e. ルートの **Trace** タブに戻り、ルートを介したメッセージのフローを確認します。

2.7.4. ルートのデバッグ

1. **Camel** タブのツリービューで、ルートを選択します。
2. **Debug** を選択し、**Start debugging** をクリックします。
3. ブレークポイントを追加するには、図のノードを選択し、**Add breakpoint** をクリックします。ノードに赤い点が表示されます。



ノードがブレークポイントのリストに追加されます。



4. 下矢印をクリックして次のノードに移動するか、**Play** ボタンをクリックしてルートの実行を再開します。
5. **Pause** ボタンをクリックして、ルートのすべてのスレッドを一時停止します。
6. 終了したら **Stop debugging** をクリックします。すべてのブレークポイントが消去されます。

2.8. JMX ドメインおよび MBEAN の表示および管理

JMX (Java Management Extensions) は、実行時にリソース (サービス、デバイス、およびアプリケーション) を動的に管理できる Java 技術です。リソースは MBean (Managed Bean) と呼ばれるオブジェクトで表現されます。リソースが作成、実装、またはインストールされると即座に管理することができます。

Fuse Console で JMX プラグインを使用すると、JMX ドメインと MBean を表示および管理できます。MBean 属性の表示、コマンドの実行、および MBean の統計を示すチャートの作成を行うことができます。

JMX タブは、フォルダーに整理されたアクティブな JMX ドメインと MBean のツリービューを提供します。詳細を確認し、MBean でコマンドを実行できます。

手順

1. MBean 属性を表示および編集するには、以下を行います。
   a. ツリービューで MBean を選択します。

b. **Attributes** タブをクリックします。

c. 属性をクリックしてその詳細を表示します。

2. 操作を実行するには、以下を行います。

a. ツリービューで MBean を選択します。

b. **Operations** タブをクリックし、リストにある操作の 1 つを展開します。

c. **Execute** をクリックし、操作を実行します。

3. チャートを表示するには、以下を行います。

a. ツリービューで項目を選択します。

b. **Chart** タブをクリックします。

2.9. QUARTZ スケジュールの表示および管理

Quartz(http://www.quartz-scheduler.org/) は、ほとんどの Java アプリケーション内で統合できる、機能豊富なオープンソースジョブスケジューリングライブラリーです。Quartz を使用して、ジョブを実行するための単純または複雑なスケジュールを作成できます。ジョブは、プログラムするほとんどすべてのものを実行できる標準の Java コンポーネントとして定義されます。

Camel ルートが **camel-quartz2** コンポーネントをデプロイすると、Fuse Console には **Quartz** タブが表示されます。JMX ツリービューから Quartz mbeans に交互にアクセスできることに注意してください。

手順

1. Fuse Console で **Quartz** タブをクリックします。
Quartz ページには、Quartz スケジューラーのツリービューならびに**Scheduler**、**Triggers**、および **Jobs** タブが含まれます。

2. スケジューラーを一時停止または開始するには、**Scheduler** タブのボタンをクリックします。

3. **Triggers** タブをクリックして、ジョブを実行するタイミングを決定するトリガーを表示します。たとえば、トリガーは、一日の指定の時刻 (ミリ秒単位) に、指定した日数、または指定した回数もしくは特定の回数繰り返してジョブを開始するように指定できます。

- トリガーの一覧をフィルターリングするには、ドロップダウンリストから **State**、**Group**、**Name**、または **Type** を選択します。続いて、入力フィールドに選択または入力することで、さらに一覧をフィルターできます。

- トリガーを一時停止、再開、更新、または手動で実行するには、**Action** 列のオプションをクリックします。

4. **Jobs** タブをクリックして実行中のジョブの一覧を表示します。テーブルの**Group**、**Name**、**Durable**、**Recover**、**Job ClassName**、および**Description** の列でリストをソートできます。

2.10. 診断の表示

Diagnostics タブを使用して、JVM DiagnosticCommand および HotspotDiangostic インターフェイスから JVM に関する診断情報を表示します。

注記

この機能は、Java Mission Control (jmc) の **Diagnostic Commands** ビューや、コマンドラインツールの jcmd と似ています。場合によっては、プラグインが対応する jcmd コマンドを提供します。

手順

1. ロードされたクラスのインスタンス数や、これらのインスタンスが使用するバイト数を取得するには、**Class Histogram** をクリックします。操作が繰り返し行われると、最後の操作実行との差異がタブに表示されます。
2. JVM 診断フラグ設定を表示するには、**JVM flags** をクリックします。
3. 稼働中の JVM でもフラグ設定を変更できます。

関連情報

サポートされる JVM はプラットフォームによって異なります。詳細は以下を参照してください。

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

2.11. スレッドの表示

スレッドの状態を表示および監視できます。

手順

1. **Runtime** タブをクリックし、**Threads** サブタブをクリックします。**Threads** ページには、アクティブなスレッドと各スレッドのスタックトレースの詳細が表示されます。デフォルトでは、スレッドリストにはすべてのスレッドが ID 値が大きい順に表示されます。
2. ID 値が小さい順に表示するには、**ID** 列ラベルをクリックします。
3. 任意で、スレッドの状態 (例: **Blocked**) やスレッド名でリストを絞り込むことができます。
4. ロッククラス名やスレッドのフルスタックトレースなど、特定スレッドの詳細情報を表示するには、**Actions** 列で **More** をクリックします。

第3章 KARAF スタンドアロンでの RED HAT FUSE アプリケーションの監視および管理

3.1. FUSE CONSOLE

Red Hat Fuse Console は、HawtIO オープンソースソフトウェアをベースとする Web コンソールです。サポートされるブラウザーのリストは Red Hat Fuse でサポートされる設定 を参照してください。

Fuse Console は、デプロイされた 1 つ以上の Fuse コンテナーの詳細を確認および管理する中央インターフェイスを提供します。また、Red Hat Fuse およびシステムリソースの監視、更新の実行、およびサービスの開始と停止を行うこともできます。

Fuse Console は、Red Hat Fuse スタンドアロンをインストールしたり Fuse on OpenShift の使用すると利用することができます。Fuse Console で表示および管理できるインテグレーションは、実行されているプラグインによって異なります。使用できるプラグインには以下が含まれます。

- Camel
- JMX
- OSGI
- Runtime
- Logs

3.2. FUSE CONSOLE へのアクセス

Apache Karaf スタンドアロンの Fuse Console にアクセスするには、以下の手順にしたがいます。

前提条件

Karaf コンテナーに Fuse をインストールします。ステップごとの手順は Apache Karaf のインストール を参照してください。

手順

1. コマンドラインで、Red Hat Fuse をインストールしたディレクトリーに移動し、以下のコマンドを実行して Fuse スタンドアロンを起動します。

   ```
   ./bin/fuse
   ```

   Karaf コンソールが起動し、バージョン情報、デフォルトの Fuse Console URL、および一般的なコマンドのリストが表示されます。

2. ブラウザーで URL を入力し、Fuse Console に接続します。例: **http://localhost:8181/hawtio**

3. ログインページでユーザー名とパスワードを入力し、**Log In** をクリックします。

デフォルトでは、Fuse Console に Home ページが表示されます。左側のナビゲーションタブには実行中のプラグインが表示されます。

3.3. FUSE CONSOLE のセキュア化

Apache Karaf で Fuse Console をセキュアにするには、以下を行います。

- **AWS へのデプロイ時に Fuse Console のプロキシーサーブレットを無効化**
  スタンドアロン Fuse アプリケーションを Amazon Web Services (AWS) にデプロイする場合、**hawtio.disableProxy** システムプロパティーを **true** に設定して、Fuse Console のプロキシーサーブレットを無効にする必要があります。

注記

Fuse Console のプロキシーサーブレットを無効にすると、Fuse Console の **Connect** タブが無効になり、Fuse Console から他の JVM に接続できなくなります。AWS に複数の Fuse アプリケーションをデプロイする場合は、アプリケーションごとに Fuse Console をデプロイする必要があります。

- **必要なプロトコルとして HTTPS を設定する**
  hawtio.http.strictTransportSecurity プロパティーを使用すると、Web ブラウザーでセキュア HTTPS プロトコルを使用して Fuse Console にアクセスするよう要求できます。このプロパティーでは、HTTP を使用して Fuse Console へのアクセスを試みる Web ブラウザーは、要求を自動的に変換して HTTPS を使用する必要があることが指定されます。

- **公開鍵を使用して応答をセキュアにする**
  特定の暗号の公開鍵を Fuse Console に関連付けし、偽造された証明書を使用した中間者攻撃のリスクを軽減するよう Web ブラウザーに要求すると、**hawtio.http.publicKeyPins** プロパティーを使用して HTTPS プロトコルをセキュアにすることができます。

- **SSL/TLS セキュリティーを有効にする**
  SSL/TLS セキュリティーは、Fuse Console ではデフォルトで有効になっていません。Fuse Console で SSL/TLS セキュリティーを有効にして、ユーザー名およびパスワードのクレデンシャルをスヌーピングから保護することが推奨されます。

- **Red Hat Single Sign On を実装する**

- **ユーザーのアクセスを制御する**
  表3.1「Karaf スタンドアロンでのロールベースアクセス」に記載されているとおり、認証されたユーザーが実行できる操作はそのユーザーに割り当てられたロールによって異なります。

手順

1. HTTPS を必須のプロトコルとして設定するには、**$KARAF_HOME/etc/system.properties** ファイルの **hawtio.http.strictTransportSecurity** プロパティーを以下の例のように設定します。

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

2. 公開鍵を使用して応答をセキュアにするには、**$KARAF_HOME/etc/system.properties** ファイルの **hawtio.http.publicKeyPins** プロパティーを以下の例のように設定します。

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

3. (AWS のみにデプロイする場合) Fuse Console のプロキシーサーブレットを無効にするには、以下の例のように **$KARAF_HOME/etc/system.properties** ファイルで **hawtio.disableProxy** プロパティーを **true** に設定します。

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

4. SSL/TLS セキュリティーを有効にする方法の詳細は、Apache Karaf セキュリティーガイド の Enabling SSL/TLS for Undertow in an Apache Karaf container を参照してください。

5. Red Hat Single Sign-On で Fuse Console をセキュアにする方法については、Red Hat Single Sign-on Securing Applications and Services Guide の **Securing the Hawtio Administration Console** セクションを参照してください。

6. 以下の手順にしたがってユーザーロールを設定し、必要な Fuse Console 操作を実行できる権限をユーザーに付与します。

   a. エディターで Red Hat Fuse **etc/users.properties** ファイルを開きます。

   b. ユーザー名、パスワード、およびロールのエントリーを追加します。
   たとえば、**etc/users.properties** ファイルの以下のエントリーは管理ユーザーを定義し、管理ロールを割り当てます。

   ```
   admin = secretpass,admin
   ```

   c. ファイルを保存します。

関連情報

- **hawtio.http.strictTransportSecurity** プロパティーの構文の説明は、HTTP Strict Transport Security (HSTS) 応答ヘッダーの説明ページを参照してください。
- **hawtio.http.publicKeyPins** プロパティーの構文や、Base64 でエンコードされた公開鍵の抽出方法の説明は、HTTP Public Key Pinning 応答ヘッダーの説明ページを参照してください。

3.4. ロールベースのアクセス参照

表3.1「Karaf スタンドアロンでのロールベースアクセス」に記載されているとおり、認証されたユーザーが実行できる操作はそのユーザーに割り当てられたロールによって異なります。

表3.1 Karaf スタンドアロンでのロールベースアクセス

操作	admin	manager (マネージャー)	viewer
ログイン/ログアウト	可能	Y	可能
ヘルプトピックの表示	可能	Y	可能
ユーザー設定の指定	可能	Y	可能
接続			
リモートインテグレーションの検出および接続	可能	Y	可能
ローカルインテグレーションの検出および接続	可能	Y	可能

操作	admin	manager (マネージャー)	viewer
Camel			
実行中の Camel アプリケーションをすべて表示	可能	Y	可能
Camel コンテキストの開始、一時停止、再開、および削除	可能	可能	
メッセージの送信	可能	可能	
エンドポイントの追加	可能	可能	
ルート、ルート図、およびランタイム統計の表示	可能	Y	可能
ルートの起動と停止	可能	可能	
ルートの削除	可能	可能	
JMX			
属性値の変更	可能	可能	
タイムベースチャートで属性を選択および表示	可能	Y	可能
操作の表示	可能	Y	可能
OSGI			
バンドル、機能、パッケージ、サービス、サーバー、フレームワーク、および設定を表示	可能	Y	可能
バンドルの追加および削除	可能	可能	
設定の追加	可能	可能	
機能のインストールおよびアンインストール	可能		
Runtime			

操作	admin	manager (マネージャー)	viewer
システムプロパティー、メトリクス、およびスレッドの表示	可能	Y	可能
Logs			
ログの表示	可能	Y	可能

関連情報

ロールベースアクセス制御の詳細は、Apache Karaf へのデプロイ を参照してください。

3.5. FUSE CONSOLE ブランディングのカスタマイズ

Fuse Console ブランディングプラグインを使用すると、タイトル、ロゴ、ログインページの情報などの Fuse Console ブランディング情報をカスタマイズできます。

デフォルトでは、Fuse Console のブランディングは Fuse Console WAR ファイル (**karaf-install-dir/system/io/hawt/hawtio-war/<version>/hawtio-war-<version>.war**) の **hawtconfig.json** で定義されます。Fuse Console ブランディングプラグインを実装すると、デフォルトのブランディングを独自のカスタムブランディングで上書きすることができます。

手順

1. ブランディングプラグインの例を **https://github.com/hawtio/hawtio/tree/master/examples/branding-plugin** から任意のローカルディレクトリーにダウンロードします。
2. 任意のエディターで、Fuse Console ブランディングプラグインの **src/main/webapp/plugin/brandingPlugin.js** ファイルを開き、Fuse Console ブランディングをカスタマイズします。
   表A.1「Fuse Console 設定プロパティー」に記載されている設定プロパティーの値を変更できます。
3. 変更を保存します。
4. 任意のエディターで、Fuse Console ブランディングプラグインの **pom.xml** ファイルをその **<parent>** セクションで開きます。

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

5. 以下のように **<parent>** セクションを編集します。
   a. **<version>** プロパティーの値を Fuse on Karaf インストールのバージョンに一致するよう変更します。たとえば、Fuse on Karaf インストールディレクトリー名が **2.0.0.fuse-760015** の場合は、バージョンを **2.0.0.fuse-760015** に設定します。

b. **<relativePath>../..</relativePath>** 行を削除します。
以下に例を示します。

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

6. ターミナルウィンドウで、以下のコマンドを実行し、branding-plugin プロジェクトをビルドします。

```
mvn clean install
```

7. Fuse が稼働していない場合は、以下のコマンドを実行して起動します。
Linux/Unix の場合: bin/fuse

Windows の場合: bin\fuse.bat`

8. Karaf CLI プロンプトで以下のコマンドを入力し、Fuse Console ブランディングプラグインをインストールします。**<version>** は Fuse on Karaf インストールのバージョンに置き換えます。
Linux/Unix の場合: install -s mvn:io.hawt/branding-plugin/<version>/war

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

9. Web ブラウザーで、7. の起動コマンドによって返された URL を使用して、Fuse Console を開きます (デフォルトの URL は **http://localhost:8181/hawtio/** です)。

注記

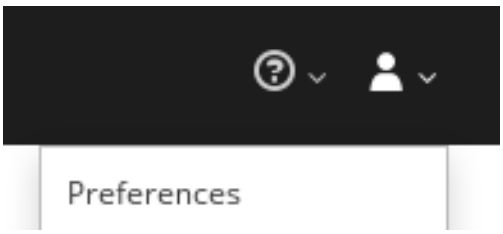
Web ブラウザーで Fuse Console をすでに実行した場合、ブランディングはブラウザーのローカルストレージに保存されます。新しいブランディング設定を使用するには、ブラウザーのローカルストレージをクリアする必要があります。

3.6. FUSE CONSOLE でデータが正しく表示されるよう確認

Fuse Console のキューおよびコネクションの表示で、不足しているキューやコネクションがあったり、一貫性のないアイコンが表示される場合、Jolokia が応答でマーシャルするアレイの要素の最大数を指定する、Jolokia コレクションサイズパラメーターを調節します。

手順

1. Fuse Console の右上隅にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. **Maximum collection size** オプションの値を大きくします (デフォルトは 50,000)。

3. **Close** をクリックします。

3.7. FUSE CONSOLE の無効化

Karaf で Fuse Console を無効にすると、他のコンポーネントに影響を与えずにすべてのユーザーをアクセス不可能にすることができます。

手順

1. hawtio-web バンドル ID を特定するには、以下のコマンドを使用して Fuse Console が使用する Fuse バンドルをリストします。
   osgi:list | grep hawtio

2. バンドルを停止するには、**osgi:stop** コマンドを使用します。たとえば、**hawtio :: Web console** バンドルの ID が 246 の場合、以下のコマンドを入力します。
   osgi:stop 246

バンドルが解決状態になり、Fuse Console にアクセスできないようになります。

関連情報

バンドルの管理に関する詳細は Apache Karaf へのデプロイ の Lifecycle Management の章を参照してください。

3.8. リモート FUSE アプリケーションへの接続

Fuse Console は Jolokia を使用します。Jolokia は、クライアントに追加のソフトウェア (エージェント) をインストールする必要がある Java Management Extensions (JMX) にエージェントベースで対応します。Red Hat Fuse には jolokia エージェントがデフォルトで含まれています。

スタンドアロン Fuse Console ディストリビューションでは、内部ですでに jolokia エージェント (https://jolokia.org/) が実行されているリモートインテグレーションに接続することができます。接続するプロセス内に jolokia エージェントがない場合は、jolokia のドキュメント (http://jolokia.org/agent.html) を参照してください。

3.8.1. Fuse Console のアンロック

デフォルトでは、Apache Karaf 上の Fuse 7 スタンドアロンの Jolokia はロックされ、Fuse Console はリモートでアクセスできません。

locahost や **127.0.0.1** 以外のホスト名や IP アドレスの Fuse Console をアンロックするには、以下の手順にしたがいます。

1. エディターで **$KARAF_HOME/etc/jolokia-access.xml** ファイルを開きます。

2. Fuse Console でアクセスする Fuse インテグレーションのホスト名または IP アドレスを登録するため、**<cors>** セクションにホスト名または IP アドレスを追加します。
   たとえば、Fuse Console からホスト名 **0.0.0.3** にアクセスするために追加する行は次のとおりです。

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

   この行を次のように追加します。

   ```
   <!--

     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. ファイルを保存します。

3.8.2. リモートアクセスの制限

任意設定として、特定のホストおよび IP アドレスの Fuse Console へのリモートアクセスを制限できます。

HTTP クライアントの IP アドレスを基にして全体的なアクセス権限を割り当てることができます。これらの制限を指定するには、以下を行います。

jolokia-access.xml ファイルで、**<host>** 要素が 1 つ以上含まれる **<remote>** セクションを追加または編集します。**<host>** 要素に対して IP アドレス、ホスト名、または CIDR 形式のネットマスク (例: 10.0 ネットワークからのすべてのクライアントは **10.0.0.0/16**) を指定できます。

以下の例は、ローカルホストと IP アドレスが **10.0** で始まるすべてのクライアントからのアクセスを許可します。他の IP アドレスの場合はアクセスが拒否されます。

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

詳細は Jolokia のセキュリティーに関するドキュメント (https://jolokia.org/reference/html/security.html) を参照してください。

3.8.3. リモート Fuse インスタンスへのコネクションの許可

Fuse Console のプロキシーサーブレットはホワイトリストを使ってホストを保護し、Fuse Console はデフォルトではローカルホストのみに接続できます。Fuse Console を他のリモート Fuse インスタンスに接続する場合は、ホワイトリストを以下のように設定する必要があります。

Apache Karaf では、**etc/system.properties** ファイルで設定を以下のように変更します。

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

3.8.4. リモート Jolokia エージェントへの接続

作業を開始する前に、リモート Jolokia エージェントの接続詳細 (ホスト名、ポート、およびパス) を知っておく必要があります。

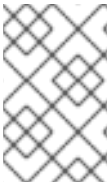
Fuse on Apache Karaf における Jolokia エージェントのデフォルトの接続 URL は **http://<host>:8181/hawtio/jolokia** になります。

システム管理者はこのデフォルト設定を変更できます。

通常、Jolokia エージェントにリモートで接続する URL は、Fuse Console を開く URL に **/jolokia** を追加したものです。たとえば、Fuse Console を開く URL が **http://<host>:1234/hawtio** の場合、リモート接続の URL は **http://<host>:1234/hawtio/jolokia** になります。

JVM を確認するためにリモート Jolokia インスタンスに接続するには、以下を行います。

1. **Connect** タブをクリックします。
2. **Remote** タブをクリックし、**Add connection** をクリックします。
3. **Name**、**Scheme** (HTTP または HTTPS)、および **hostname** を入力します。
4. **Test Connection** をクリックします。
5. **Add** をクリックします。



注記

Fuse Console は自動的にローカルホストと 127.0.0.1 以外のローカルネットワークインターフェイスをプローブし、ホワイトリストに追加します。そのため、ローカルマシンのアドレスを手作業でホワイトリストに登録する必要はありません。

3.8.5. データ移動設定の指定

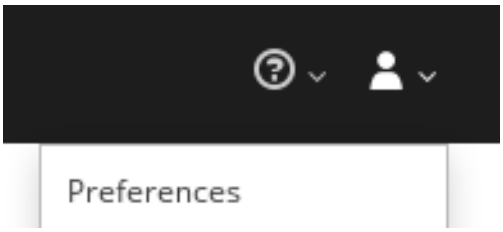
Fuse Console に表示されるデータをより頻繁にリフレッシュする場合などに、以下の Jolokia 設定を変更することができます。データの更新を頻繁に行うと、ネットワークトラフィックに影響し、サーバーに対するリクエストの数が増加するため注意してください。

- **Update rate** - JMX データを取得するため Jolokia へポーリングを行う間隔 (デフォルトは 5 秒)。

- **Maximum depth** - 戻る前にサーバー側で Jolokia がオブジェクトを JSON にマーシャルするレベル数 (デフォルトは 7)。
- **Maximum collection size** - 応答で Jolokia がマーシャルするアレイの最大要素数 (デフォルトは 50,000)。

これらの設定の値を変更するには、以下を行います。

1. Fuse Console の右上にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. オプションを編集して **Close** をクリックします。

3.8.6. JVM ランタイム情報の表示

システムプロパティー、メトリクス、スレッドなどの JVM のランタイム情報を表示するには、**Runtime** タブをクリックします。

3.9. APACHE CAMEL アプリケーションの表示および管理

Fuse Console の **Camel** タブで Apache Camel のコンテキスト、ルート、および依存関係を表示および管理します。

次の詳細を表示できます。

- 実行中の Camel コンテキストすべてのリスト。
- Camel バージョン番号やランタイム統計など、各 Camel コンテキストの詳細情報。
- 各 Camel アプリケーションの全ルートおよびランタイム統計のリスト。
- 実行中のルートとリアルタイムのメトリクスのグラフィック表示。

また、以下を行うと Camel アプリケーションと対話もできます。

- コンテキストの開始および一時停止。
- 再起動、停止、一時停止、再開などを実行できるよう、すべての Camel アプリケーションとそれらのルートのライフサイクルを管理。
- 実行中のルートのライブトレースおよびデバッグ。
- Camel エンドポイントへのメッセージの閲覧および送信。

前提条件

Camel タブは、1 つ以上の Camel ルートを使用するコンテナーに接続する場合のみ使用できます。

3.9.1. コンテキストの開始、一時停止、または削除

1. **Camel** タブのツリービューで、**Camel Contexts** をクリックします。
2. リストのコンテキストの横にあるボックスにチェックマークを入れます。
3. **Start** または **Suspend** をクリックします。
4. コンテキストを削除するには以下を行います。
   a. コンテキストを停止します。
   b. 楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。

注記

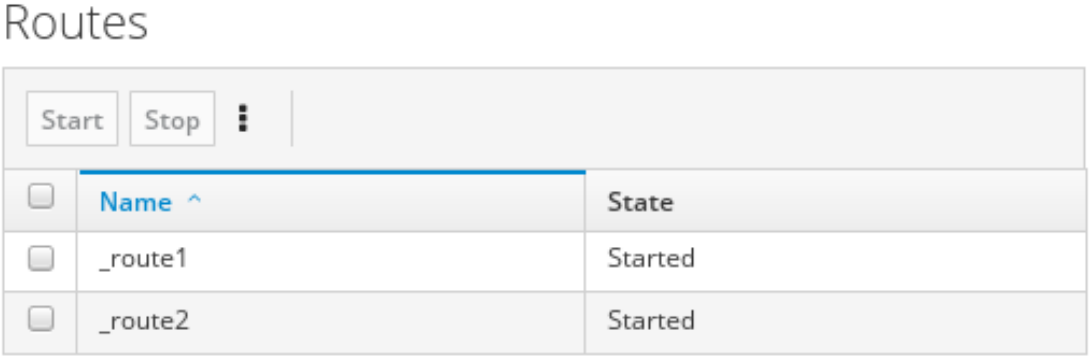
コンテキストを削除する場合、デプロイされたアプリケーションから削除します。

3.9.2. Camel アプリケーションの詳細表示

1. **Camel** タブのツリービューで、Camel アプリケーションをクリックします。
2. アプリケーションの属性と値のリストを表示するには、**Attributes** をクリックします。
3. アプリケーション属性をグラフィックに表示するには、**Chart** をクリックした後、**Edit** をクリックし、チャートに表示する属性を選択します。
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. アプリケーションエンドポイントを表示するには、**Endpoints** をクリックします。リストは **URL**、**Route ID**、および **direction** で絞り込むことができます。
6. メッセージ本文とメッセージヘッダーを別のタイプに変換するために使用される Camel 組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化するには、**Type Converters** をクリックします。
7. JMX 操作 (XML からのルートの追加または更新、クラスパスで利用できる Camel コンポーネントの検索など) を表示および実行するには、**Operations** をクリックします。

3.9.3. Camel ルートリストの表示および Camel ルートとの対話

1. ルートのリストを表示するには、以下を行います。
   a. **Camel** タブをクリックします。
   b. ツリービューでアプリケーションの routes フォルダーをクリックします。

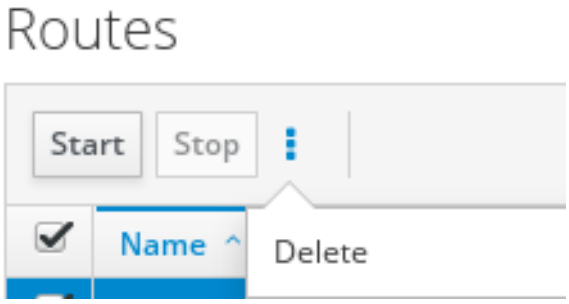


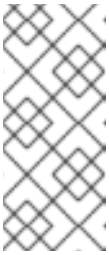
2. 1つまたは複数のルートを開始、停止、または削除するには、以下を行います。

   a. リストのルートの横にあるボックスにチェックマークを入れます。

   b. **Start** または **Stop** をクリックします。

   c. 最初にルートを停止してから削除する必要があります。停止したら楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。





注記

- ルートを削除する場合、デプロイされたアプリケーションから削除します。
- ツリービューで特定のルートを選択し、右上のメニューをクリックして開始、停止、または削除することもできます。

3. ルートのグラフィックな図を表示するには、**Route Diagram** をクリックします。

4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。

5. エンドポイントを表示するには、**Endpoints** をクリックします。URL、Route ID、および方向でリストを絞り込むことができます。

6. **Type Converters** をクリックし、Camel の組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化します。このメカニズムはメッセージ本文とメッセージヘッダーを別のタイプに変換するために使用されます。

7. 特定のルートと対話するには、以下を行います。

   a. **Camel** タブのツリービューで、ルートを選択します。

   b. ルート属性と値のリストを表示するには、**Attributes** をクリックします。

   c. ルート属性をグラフィックに表示するには、**Chart** をクリックします。**Edit** をクリックするとチャートに表示する属性を選択することができます。

   d. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。

   e. **Operations** をクリックして JMX 操作 (ルートを XML としてダンプ、ルートの Camel ID 値の取得など) を表示および実行できます。

8. ルートを介してメッセージをトレースするには、以下を実行します。

   a. **Camel** タブのツリービューで、ルートを選択します。

   b. **Trace** を選択し、**Start tracing** をクリックします。

9. メッセージをルートに送信するには、以下を行います。

   a. **Camel** タブのツリービューでコンテキストのエンドポイントフォルダーを開き、エンドポイントを選択します。

   b. **Send** サブタブをクリックします。

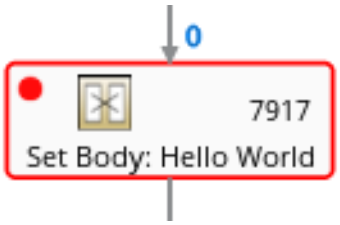
   c. JSON または XML 形式のメッセージを設定します。

   d. **Send** をクリックします。

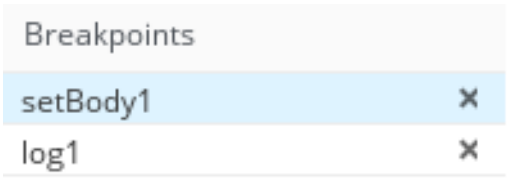
   e. ルートの **Trace** タブに戻り、ルートを介したメッセージのフローを確認します。

3.9.4. ルートのデバッグ

1. **Camel** タブのツリービューで、ルートを選択します。
2. **Debug** を選択し、**Start debugging** をクリックします。
3. ブレークポイントを追加するには、図のノードを選択し、**Add breakpoint** をクリックします。ノードに赤い点が表示されます。



ノードがブレークポイントのリストに追加されます。



4. 下矢印をクリックして次のノードに移動するか、**Play** ボタンをクリックしてルートの実行を再開します。
5. **Pause** ボタンをクリックして、ルートのすべてのスレッドを一時停止します。
6. 終了したら **Stop debugging** をクリックします。すべてのブレークポイントが消去されます。

3.10. JMX ドメインおよび MBEAN の表示および管理

JMX (Java Management Extensions) は、実行時にリソース (サービス、デバイス、およびアプリケーション) を動的に管理できる Java 技術です。リソースは MBean (Managed Bean) と呼ばれるオブジェクトで表現されます。リソースが作成、実装、またはインストールされると即座に管理することができます。

Fuse Console で JMX プラグインを使用すると、JMX ドメインと MBean を表示および管理できます。MBean 属性の表示、コマンドの実行、および MBean の統計を示すチャートの作成を行うことができます。

JMX タブは、フォルダーに整理されたアクティブな JMX ドメインと MBean のツリービューを提供します。詳細を確認し、MBean でコマンドを実行できます。

手順

1. MBean 属性を表示および編集するには、以下を行います。
   a. ツリービューで MBean を選択します。
   b. **Attributes** タブをクリックします。
   c. 属性をクリックしてその詳細を表示します。
2. 操作を実行するには、以下を行います。
   a. ツリービューで MBean を選択します。
   b. **Operations** タブをクリックし、リストにある操作の 1 つを展開します。
   c. **Execute** をクリックし、操作を実行します。
3. チャートを表示するには、以下を行います。
   a. ツリービューで項目を選択します。
   b. **Chart** タブをクリックします。

3.11. QUARTZ スケジュールの表示および管理

Quartz(http://www.quartz-scheduler.org/) は、ほとんどの Java アプリケーション内で統合できる、機能豊富なオープンソースジョブスケジューリングライブラリーです。Quartz を使用して、ジョブを実行するための単純または複雑なスケジュールを作成できます。ジョブは、プログラムするほとんどすべてのものを実行できる標準の Java コンポーネントとして定義されます。

Camel ルートが **camel-quartz2** コンポーネントをデプロイすると、Fuse Console には **Quartz** タブが表示されます。JMX ツリービューから Quartz mbeans に交互にアクセスできることに注意してください。

手順

1. Fuse Console で **Quartz** タブをクリックします。
   Quartz ページには、Quartz スケジューラーのツリービューならびに**Scheduler**、**Triggers**、および **Jobs** タブが含まれます。
2. スケジューラーを一時停止または開始するには、**Scheduler** タブのボタンをクリックします。
3. **Triggers** タブをクリックして、ジョブを実行するタイミングを決定するトリガーを表示します。たとえば、トリガーは、一日の指定の時刻 (ミリ秒単位) に、指定した日数、または指定した回数もしくは特定の回数繰り返してジョブを開始するように指定できます。
   - トリガーの一覧をフィルターリングするには、ドロップダウンリストから **State**、**Group**、**Name**、または **Type** を選択します。続いて、入力フィールドに選択または入力することで、さらに一覧をフィルターできます。
   - トリガーを一時停止、再開、更新、または手動で実行するには、**Action** 列のオプションをクリックします。

4. **Jobs** タブをクリックして実行中のジョブの一覧を表示します。テーブルの**Group**、**Name**、**Durable**、**Recover**、**Job ClassName**、および**Description** の列でリストをソートできます。

3.12. OSGI 環境の表示および管理

Apache Karaf スタンドアロンディストリビューションでは、Red Hat Fuse OSGi 環境を表示および管理できます。コンテナーバンドル、機能、および設定のほか、Java パッケージや OSGi サービスも表示および管理できます。

OSGi タブには、各コンテナーコンポーネントのオプションが含まれる複数のサブタブが含まれています。

Bundles

インストールされたバンドルのリストです。バンドルのインストールおよびアンインストール、バンドルの開始および停止、およびバンドルプロパティーの編集を行うことができます。さらに、リストの絞り込みやリストとグリッドビューの切り替えを行うこともできます。

Features

使用できる機能のリスト。機能や機能リポジトリーをインストールおよびアンインストールでき、機能の詳細を表示できます。

Packages

Java パッケージのリスト。パッケージバージョンと関連するバンドルを表示できます。

Services

実行中のサービスのリスト。サービス ID、関連するバンドル、およびオブジェクトクラスを表示できます。

Declarative Services

宣言的 OSGi サービスのリスト。サービスの状態を表示し、サービスの詳細を表示することができます。また、サービスをアクティベートおよび非アクティベートすることもできます。

Server

読み取り専用モードのローカルまたはリモートホストに関する詳細情報。

Framework

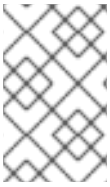
コンテナー OSGi フレームワークの設定オプション。フレームワーク開始レベルと初期バンドル開始レベルを設定できます。

Configuration

設定オブジェクトのリスト。各オブジェクトの状態を表示し、オブジェクトの詳細を表示または編集できます。また、新しい設定オブジェクトを作成することもできます。

3.13. 診断の表示

Diagnostics タブを使用して、JVM DiagnosticCommand および HotspotDiangostic インターフェイスから JVM に関する診断情報を表示します。



注記

この機能は、Java Mission Control (jmc) の **Diagnostic Commands** ビューや、コマンドラインツールの jcmd と似ています。場合によっては、プラグインが対応する jcmd コマンドを提供します。

手順

1. ロードされたクラスのインスタンス数や、これらのインスタンスが使用するバイト数を取得するには、**Class Histogram** をクリックします。操作が繰り返し行われると、最後の操作実行との差異がタブに表示されます。

2. JVM 診断フラグ設定を表示するには、**JVM flags** をクリックします。

3. 稼働中の JVM でもフラグ設定を変更できます。

関連情報

サポートされる JVM はプラットフォームによって異なります。詳細は以下を参照してください。

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

3.14. スレッドの表示

スレッドの状態を表示および監視できます。

手順

1. **Runtime** タブをクリックし、**Threads** サブタブをクリックします。**Threads** ページには、アクティブなスレッドと各スレッドのスタックトレースの詳細が表示されます。デフォルトでは、スレッドリストにはすべてのスレッドが ID 値が大きい順に表示されます。

2. ID 値が小さい順に表示するには、**ID** 列ラベルをクリックします。

3. 任意で、スレッドの状態 (例: **Blocked**) やスレッド名でリストを絞り込むことができます。

4. ロッククラス名やスレッドのフルスタックトレースなど、特定スレッドの詳細情報を表示するには、**Actions** 列で **More** をクリックします。

3.15. ログエントリーの表示

Logs タブで Red Hat Fuse のログエントリーを表示できます。

前提条件

Logs タブは、Java アプリケーションに Log MBean が含まれる場合に利用できます。

手順

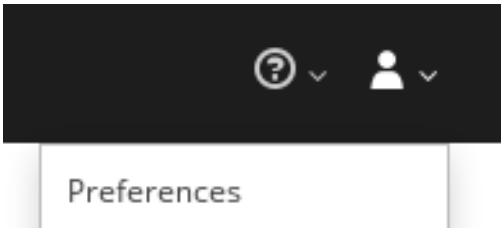
1. ログエントリーのリストを表示するには、**Log Entries** タブをクリックします。
デフォルトでは、リストのログエントリーは昇順で表示されます。

   各ログエントリーをドリルダウンすると、ログエントリーに関する詳細情報が表示されます。

2. ログのリストを絞り込み、特定のログタイプを表示するには、**Action Bar** をクリックします。テキスト文字列またはログレベルを基にしてログエントリーセクションを絞り込むことができます。

3. Fuse Console のデフォルト設定を変更するには、以下を行います。

   a. Fuse Console の右上隅にあるユーザーアイコンをクリックして、ドロップダウンメニューの **Preferences** をクリックします。



b. デフォルトのソート順序を変更するには、**Server Logs** を選択し、ログエントリーリンクをクリックして、バンドル名、スレッド、完全なメッセージテキストなどのログエントリーに関する詳細を表示します。

c. 任意で、ログメッセージを格納するための設定をカスタマイズすることができます。

- Fuse Console に保持するログステートメントの数 (デフォルトは 100)。
- グローバルログレベル: **INFO** (デフォルト)、OFF、ERROR、WARN、および DEBUG
- 子レベルのメッセージには、**hawtio-oauth** や **hawtio-core-utils** などが含まれます。

d. Fuse Console のログ設定をデフォルトの値に戻すには、**Reset → Reset settings** とクリックします。

3.16. PROMETHEUS メトリクスの有効化

Prometheus は、スタンドアロン Apache Karaf コンテナーにデプロイされたサービスの監視に使用できる、システムおよびサービスを対象としたオープンソースの監視および警告ツールキットです。Prometheus は、指定の間隔で設定されたサービスからメトリクスを収集および保存します。さらに、ルール式の評価や結果の表示を行い、指定の条件が true になるとアラートをトリガーできます。

重要

Prometheus に対する Red Hat のサポートは、Red Hat 製品ドキュメントに記載されているセットアップと推奨設定に限定されます。

Prometheus は、クライアントにインストールおよび設定された exporter を使用して、エンドポイントを Prometheus 形式で公開します。このエンドポイントは、メトリクスのリストとメトリクスの現在の値を提供する HTTP インターフェイスです。Prometheus は定期的にターゲット定義の各エンドポイントをスクレイピングし、収集したデータをそのデータベースに書き込みます。Prometheus は、現在実行中のセッションだけでなく、長期間にわたってデータを収集します。Prometheus は、データ上でクエリーをグラフィカルに可視化し、実行できるように、データを格納します。

3.16.1. スタンドアロン Apache Karaf コンテナーからメトリクスのエクスポートを有効化

Prometheus は、Camel によって公開されるメトリクスが含まれる設定ファイル (**https://raw.githubusercontent.com/jboss-fuse/application-templates/master/prometheus/prometheus-config.yml**) を使用します。

注記

識別できるメトリクスは、JMX で提供されるメトリクスに限定されます。

Apache Camel メトリクスを生成するには、Fuse アプリケーションをデプロイする必要があります。

手順

コマンドラインを使用して、スタンドアロン Apache Karaf コンテナーから Prometheus メトリクスのエクスポートを有効にするには、以下を実行します。

1. コマンドプロンプトを開き、現在の場所が Apache Karaf インストールの **etc/** ディレクトリーであることを確認します。
2. 以下のコマンドを入力して、**etc/** ディレクトリーのサンプルファイルから Prometheus 設定ファイルを作成します。

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

3. エクスポーターは、**fuse** または **fuse.bat** コマンドを使用して Fuse を起動する場合のみ使用できます。Linux または Unix の場合は **bin/fuse**、Windows の場合は **bin\fuse.bat** を実行します。
4. Fuse の再起動後、**http://localhost:9779** で Web ブラウザーを開くと、公開されたメトリクスを表示できます。

注記

必要に応じて、コマンドラインから **KARAF_PROMETHEUS_PORT** および **KARAF_PROMETHEUS_CONFIG** 設定変数のデフォルト値を変更できます。

3.16.2. Apache Karaf コンテナーから公開済みメトリクスをスクレープするための Prometheus サーバーの設定

Prometheus サーバーが Apache Karaf コンテナーからメトリクスをスクレープできるようにするには、メトリクスを公開するエンドポイントを Prometheus 設定ファイルの **targets** プロパティーに追加する必要があります。

手順

1. Prometheus インストールディレクトリーの **/prometheus.yml** 設定ファイルに移動します。
2. スクレープする Apache Karaf エンドポイントを追加します。

```
scrape_configs:
  - job_name: 'prometheus'

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

第4章 EAP スタンドアロンでの RED HAT FUSE アプリケーションの監視および管理

4.1. FUSE CONSOLE

Red Hat Fuse Console は、HawtIO オープンソースソフトウェアをベースとする Web コンソールです。サポートされるブラウザーのリストは Red Hat Fuse でサポートされる設定 を参照してください。

Fuse Console は、デプロイされた 1 つ以上の Fuse コンテナーの詳細を確認および管理する中央インターフェイスを提供します。また、Red Hat Fuse およびシステムリソースの監視、更新の実行、およびサービスの開始と停止を行うこともできます。

Fuse Console は、Red Hat Fuse スタンドアロンをインストールしたり Fuse on OpenShift の使用すると利用することができます。Fuse Console で表示および管理できるインテグレーションは、実行されているプラグインによって異なります。使用できるプラグインには以下が含まれます。

- Camel
- JMX
- OSGI
- Runtime
- Logs

4.2. FUSE CONSOLE へのアクセス

以下の手順にしたがって、Red Hat JBoss Enterprise Application Platform の Fuse Console にアクセスします。

前提条件

Fuse を JBoss EAP コンテナーにインストールする必要があります。ステップごとの手順は、JBoss EAP のインストール を参照してください。

手順

スタンドアロン JBoss EAP ディストリビューションの Fuse Console にアクセスするには、以下を実行します。

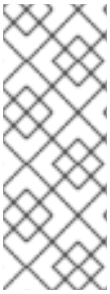
1. Red Hat Fuse スタンドアロンを以下のコマンドで起動します。
   Linux/Mac OS の場合: **./bin/standalone.sh**

   Windows の場合: **./bin/standalone.bat**

2. Web ブラウザーで URL を入力し、Fuse Console に接続します。例:
   http://localhost:8080/hawtio

3. ログインページでユーザー名とパスワードを入力し、**Log In** をクリックします。

デフォルトでは、Fuse Console に Home ページが表示されます。左側のナビゲーションタブには実行中のプラグインが表示されます。



注記

Fuse Console のメインページがブラウザーに表示されるまで時間がかかる場合、ログファイルの数とサイズを削減する必要がある場合があります。ログファイルの最大サイズ (rotate-size) に到達し、ファイル数を維持する (max-backup-index) 場合は、**periodic-size-rotating-file-handler** を使用してファイルのローテーションを行うことができます。このハンドラーの使用方法に関する詳細は、Red Hat JBoss Enterprise Application Platform の製品ドキュメントを参照してください。

4.3. FUSE CONSOLE ブランディングのカスタマイズ

Fuse Console ブランディングプラグインを使用すると、タイトル、ロゴ、ログインページの情報などの Fuse Console ブランディング情報をカスタマイズできます。

デフォルトでは、Fuse Console のブランディングは Fuse Console WAR ファイル (**eap-install-dir/standalone/deployments/hawtio-wildfly-<version>.war**) の **hawtconfig.json** で定義されます。Fuse Console ブランディングプラグインを実装すると、デフォルトのブランディングを独自のカスタムブランディングで上書きすることができます。

手順

1. ブランディングプラグインの例を **https://github.com/hawtio/hawtio/tree/master/examples/branding-plugin** から任意のローカルディレクトリーにダウンロードします。

2. 任意のエディターで、Fuse Console ブランディングプラグインの **src/main/webapp/plugin/brandingPlugin.js** ファイルを開き、Fuse Console ブランディングをカスタマイズします。
表A.1「Fuse Console 設定プロパティー」に記載されている設定プロパティーの値を変更できます。

3. 変更を保存します。

4. 任意のエディターで、Fuse Console ブランディングプラグインの **pom.xml** ファイルをその **<parent>** セクションで開きます。

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

5. 以下のように **<parent>** セクションを編集します。

   a. **<version>** プロパティーの値を Fuse on EAP インストールのバージョンに一致するよう変更します。たとえば、EAP on Karaf インストールディレクトリー名が **2.0.0.fuse-760015** の場合は、バージョンを **2.0.0.fuse-760015** に設定します。

   b. **<relativePath>../..</relativePath>** 行を削除します。
   以下に例を示します。

```
<parent>
  <groupId>io.hawt</groupId>
  <artifactId>project</artifactId>
```

```
<version> 2.0.0.fuse-760015</version>
  </parent>
```

6. ターミナルウィンドウで、以下のコマンドを実行し、branding-plugin プロジェクトをビルドします。

```
mvn clean install
```

このコマンドによって、プロジェクトの **/target** フォルダーに **branding-plugin.war** ファイルが作成されます。

7. **branding-plugin.war** ファイルを EAP インストールの **standalone/deployments** ディレクトリーにコピーします。

8. Fuse が稼働していない場合は、以下のコマンドを実行して起動します。
Linux/Mac OS の場合: **./bin/standalone.sh**

Windows の場合: **./bin/standalone.bat**

9. Web ブラウザーで、8. の起動コマンドによって返された URL を使用して、Fuse Console を開きます (デフォルトの URL は **http://localhost:8080/hawtio/** です)。

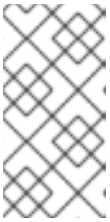
注記

Web ブラウザーで Fuse Console をすでに実行した場合、ブランディングはブラウザーのローカルストレージに保存されます。新しいブランディング設定を使用するには、ブラウザーのローカルストレージをクリアする必要があります。

4.4. FUSE CONSOLE のセキュア化

EAP で Fuse Console をセキュアにするには、以下を行います。

- **AWS へのデプロイ時に Fuse Console のプロキシーサーブレットを無効化**
スタンドアロン Fuse アプリケーションを Amazon Web Services (AWS) にデプロイする場合、**hawtio.disableProxy** システムプロパティーを **true** に設定して、Fuse Console のプロキシーサーブレットを無効にする必要があります。



注記

Fuse Console のプロキシーサーブレットを無効にすると、Fuse Console の **Connect** タブが無効になり、Fuse Console から他の JVM に接続できなくなります。AWS に複数の Fuse アプリケーションをデプロイする場合は、アプリケーションごとに Fuse Console をデプロイする必要があります。

- **必要なプロトコルとして HTTPS を設定する**
hawtio.http.strictTransportSecurity プロパティーを使用すると、Web ブラウザーでセキュア HTTPS プロトコルを使用して Fuse Console にアクセスするよう要求できます。このプロパティーでは、HTTP を使用して Fuse Console へのアクセスを試みる Web ブラウザーは、要求を自動的に変換して HTTPS を使用する必要があることが指定されます。

- **公開鍵を使用して応答をセキュアにする**
特定の暗号の公開鍵を Fuse Console に関連付けし、偽造された証明書を使用した中間者攻撃のリスクを軽減するよう Web ブラウザーに要求すると、**hawtio.http.publicKeyPins** プロパティーを使用して HTTPS プロトコルをセキュアにすることができます。

手順

1. **$EAP_HOME/standalone/configuration/standalone*.xml** ファイルの system-properties セクションにある **hawtio.http.strictTransportSecurity** および **hawtio.http.publicKeyPins** プロパティーを、以下の例のように設定します。

```
<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. (AWS のみにデプロイする場合) Fuse Console のプロキシーサーブレットを無効にするには、**$EAP_HOME/standalone/configuration/standalone*.xml** ファイルの system-properties セクションにある **hawtio.disableProxy** プロパティーを以下の例のように設定します。

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

関連情報

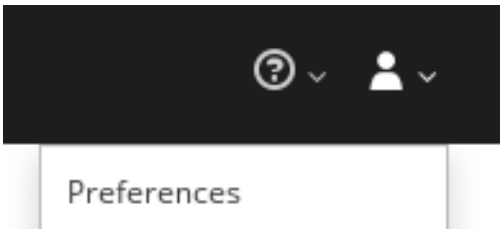
- **hawtio.http.strictTransportSecurity** プロパティーの構文の説明は、HTTP Strict Transport Security (HSTS) 応答ヘッダーの説明ページを参照してください。
- **hawtio.http.publicKeyPins** プロパティーの構文や、Base64 でエンコードされた公開鍵の抽出方法の説明は、HTTP Public Key Pinning 応答ヘッダーの説明ページを参照してください。

4.5. FUSE CONSOLE でデータが正しく表示されるよう確認

Fuse Console のキューおよびコネクションの表示で、不足しているキューやコネクションがあったり、一貫性のないアイコンが表示される場合、Jolokia が応答でマーシャルするアレイの要素の最大数を指定する、Jolokia コレクションサイズパラメーターを調節します。

手順

1. Fuse Console の右上隅にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. **Maximum collection size** オプションの値を大きくします (デフォルトは 50,000)。
3. **Close** をクリックします。

4.6. FUSE CONSOLE の無効化

JBoss EAP で Fuse Console を無効にすると、他のコンポーネントに影響を与えずにすべてのユーザーをアクセス不可能にすることができます。

手順

JBoss EAP の Fuse Console を無効にするには、以下のいずれかを実行します。

- Fuse Console デプロイメントファイル **$EAP_HOME/standalone/deployments/hawtio-wildfly-xxxxx.war** を削除します。
- JBoss EAP 管理コンソールまたはコマンドラインインターフェイスを使用して Fuse Console をアンデプロイします。

4.7. リモート FUSE アプリケーションへの接続

Fuse Console は Jolokia を使用します。Jolokia は、クライアントに追加のソフトウェア (エージェント) をインストールする必要がある Java Management Extensions (JMX) にエージェントベースで対応します。Red Hat Fuse には jolokia エージェントがデフォルトで含まれています。

スタンドアロン Fuse Console ディストリビューションでは、内部ですでに jolokia エージェント (https://jolokia.org/) が実行されているリモートインテグレーションに接続することができます。接続するプロセス内に jolokia エージェントがない場合は、jolokia のドキュメント (http://jolokia.org/agent.html) を参照してください。

4.7.1. Fuse Console のアンロック

デフォルトでは、JBoss EAP 上の Fuse 7 スタンドアロンの Jolokia はロックされ、Fuse Console はリモートでアクセスできません。

locahost や **127.0.0.1** 以外のホスト名や IP アドレスの Fuse Console をアンロックするには、以下の手順にしたがいます。

1. エディターで **$EAP_HOME/standalone/configuration/jolokia-access.xml** ファイルを開きます。
2. Fuse Console でアクセスする Fuse インテグレーションのホスト名または IP アドレスを登録するため、**<cors>** セクションにホスト名または IP アドレスを追加します。
   たとえば、Fuse Console からホスト名 **0.0.0.3** にアクセスするために追加する行は次のとおりです。

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

   この行を次のように追加します。

   ```
   <!--

     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. ファイルを保存します。

4.7.2. リモートアクセスの制限

任意設定として、特定のホストおよび IP アドレスの Fuse Console へのリモートアクセスを制限できます。

HTTP クライアントの IP アドレスを基にして全体的なアクセス権限を割り当てることができます。これらの制限を指定するには、以下を行います。

jolokia-access.xml ファイルで、**<host>** 要素が 1 つ以上含まれる **<remote>** セクションを追加または編集します。**<host>** 要素に対して IP アドレス、ホスト名、または CIDR 形式のネットマスク (例: 10.0 ネットワークからのすべてのクライアントは **10.0.0.0/16**) を指定できます。

以下の例は、ローカルホストと IP アドレスが **10.0** で始まるすべてのクライアントからのアクセスを許可します。他の IP アドレスの場合はアクセスが拒否されます。

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

詳細は Jolokia のセキュリティーに関するドキュメント (https://jolokia.org/reference/html/security.html) を参照してください。

4.7.3. リモート Fuse インスタンスへのコネクションの許可

Fuse Console のプロキシーサーブレットはホワイトリストを使ってホストを保護し、Fuse Console はデフォルトではローカルホストのみに接続できます。Fuse Console を他のリモート Fuse インスタンスに接続する場合は、**standalone/configuration/standalone-*.xml** ファイルで以下の設定変更を行います。

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

4.7.4. リモート Jolokia エージェントへの接続

作業を開始する前に、リモート Jolokia エージェントの接続詳細 (ホスト名、ポート、およびパス) を知っておく必要があります。

Red Hat JBoss EAP における Jolokia エージェントのデフォルトの接続 URL は **http://<host>:8080/hawtio/jolokia** です。

システム管理者はこのデフォルト設定を変更できます。

通常、Jolokia エージェントにリモートで接続する URL は、Fuse Console を開く URL に **/jolokia** を追加したものです。たとえば、Fuse Console を開く URL が **http://<host>:1234/hawtio** の場合、リモート接続の URL は **http://<host>:1234/hawtio/jolokia** になります。

JVM を確認するためにリモート Jolokia インスタンスに接続するには、以下を行います。

1. **Connect** タブをクリックします。
2. **Remote** タブをクリックし、**Add connection** をクリックします。
3. **Name**、**Scheme** (HTTP または HTTPS)、および **hostname** を入力します。
4. **Test Connection** をクリックします。
5. **Add** をクリックします。

注記

Fuse Console は自動的にローカルホストと 127.0.0.1 以外のローカルネットワークインターフェイスをプローブし、ホワイトリストに追加します。そのため、ローカルマシンのアドレスを手作業でホワイトリストに登録する必要はありません。

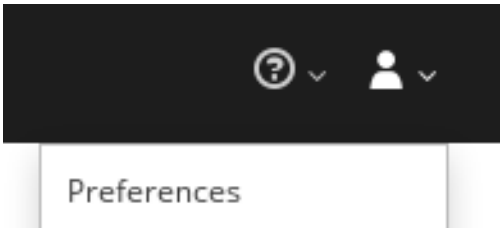
4.7.5. データ移動設定の指定

Fuse Console に表示されるデータをより頻繁にリフレッシュする場合などに、以下の Jolokia 設定を変更することができます。データの更新を頻繁に行うと、ネットワークトラフィックに影響し、サーバーに対するリクエストの数が増加するため注意してください。

- **Update rate** - JMX データを取得するため Jolokia へポーリングを行う間隔 (デフォルトは 5 秒)。
- **Maximum depth** - 戻る前にサーバー側で Jolokia がオブジェクトを JSON にマーシャルするレベル数 (デフォルトは 7)。
- **Maximum collection size** - 応答で Jolokia がマーシャルするアレイの最大要素数 (デフォルトは 50,000)。

これらの設定の値を変更するには、以下を行います。

1. Fuse Console の右上にあるユーザーアイコンをクリックして、**Preferences** をクリックします。



2. オプションを編集して **Close** をクリックします。

4.7.6. JVM ランタイム情報の表示

システムプロパティー、メトリクス、スレッドなどの JVM のランタイム情報を表示するには、**Runtime** タブをクリックします。

4.8. APACHE CAMEL アプリケーションの表示および管理

Fuse Console の **Camel** タブで Apache Camel のコンテキスト、ルート、および依存関係を表示および管理します。

次の詳細を表示できます。

- 実行中の Camel コンテキストすべてのリスト。
- Camel バージョン番号やランタイム統計など、各 Camel コンテキストの詳細情報。
- 各 Camel アプリケーションの全ルートおよびランタイム統計のリスト。
- 実行中のルートとリアルタイムのメトリクスのグラフィック表示。

また、以下を行うと Camel アプリケーションと対話もできます。

- コンテキストの開始および一時停止。
- 再起動、停止、一時停止、再開などを実行できるよう、すべての Camel アプリケーションとそれらのルートのライフサイクルを管理。
- 実行中のルートのライブトレースおよびデバッグ。
- Camel エンドポイントへのメッセージの閲覧および送信。

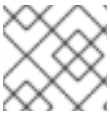
前提条件

Camel タブは、1 つ以上の Camel ルートを使用するコンテナーに接続する場合のみ使用できます。

4.8.1. コンテキストの開始、一時停止、または削除

1. **Camel** タブのツリービューで、**Camel Contexts** をクリックします。
2. リストのコンテキストの横にあるボックスにチェックマークを入れます。
3. **Start** または **Suspend** をクリックします。
4. コンテキストを削除するには以下を行います。
    a. コンテキストを停止します。

b. 楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



注記

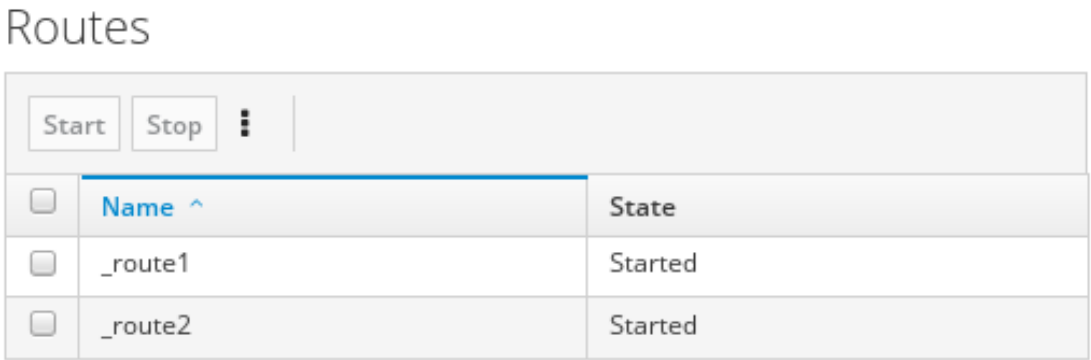
コンテキストを削除する場合、デプロイされたアプリケーションから削除します。

4.8.2. Camel アプリケーションの詳細表示

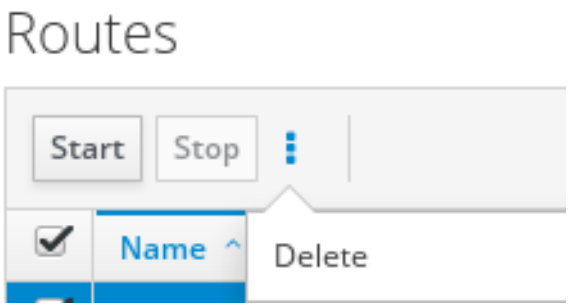
1. **Camel** タブのツリービューで、Camel アプリケーションをクリックします。
2. アプリケーションの属性と値のリストを表示するには、**Attributes** をクリックします。
3. アプリケーション属性をグラフィックに表示するには、**Chart** をクリックした後、**Edit** をクリックし、チャートに表示する属性を選択します。
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. アプリケーションエンドポイントを表示するには、**Endpoints** をクリックします。リストは **URL**、**Route ID**、および **direction** で絞り込むことができます。
6. メッセージ本文とメッセージヘッダーを別のタイプに変換するために使用される Camel 組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化するには、**Type Converters** をクリックします。
7. JMX 操作 (XML からのルートの追加または更新、クラスパスで利用できる Camel コンポーネントの検索など) を表示および実行するには、**Operations** をクリックします。

4.8.3. Camel ルートリストの表示および Camel ルートとの対話

1. ルートのリストを表示するには、以下を行います。
   a. **Camel** タブをクリックします。
   b. ツリービューでアプリケーションの routes フォルダーをクリックします。



2. 1 つまたは複数のルートを開始、停止、または削除するには、以下を行います。
   a. リストのルートの横にあるボックスにチェックマークを入れます。
   b. **Start** または **Stop** をクリックします。
   c. 最初にルートを停止してから削除する必要があります。停止したら楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



注記

- ルートを削除する場合、デプロイされたアプリケーションから削除します。
- ツリービューで特定のルートを選択し、右上のメニューをクリックして開始、停止、または削除することもできます。

3. ルートのグラフィックな図を表示するには、**Route Diagram** をクリックします。
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. エンドポイントを表示するには、**Endpoints** をクリックします。URL、Route ID、および方向でリストを絞り込むことができます。
6. **Type Converters** をクリックし、Camel の組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化します。このメカニズムはメッセージ本文とメッセージヘッダーを別のタイプに変換するために使用されます。
7. 特定のルートと対話するには、以下を行います。
   a. **Camel** タブのツリービューで、ルートを選択します。
   b. ルート属性と値のリストを表示するには、**Attributes** をクリックします。
   c. ルート属性をグラフィックに表示するには、**Chart** をクリックします。**Edit** をクリックするとチャートに表示する属性を選択することができます。
   d. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
   e. **Operations** をクリックして JMX 操作 (ルートを XML としてダンプ、ルートの Camel ID 値の取得など) を表示および実行できます。
8. ルートを介してメッセージをトレースするには、以下を実行します。
   a. **Camel** タブのツリービューで、ルートを選択します。
   b. **Trace** を選択し、**Start tracing** をクリックします。
9. メッセージをルートに送信するには、以下を行います。
   a. **Camel** タブのツリービューでコンテキストのエンドポイントフォルダーを開き、エンドポイントを選択します。
   b. **Send** サブタブをクリックします。

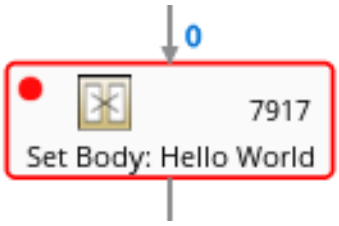
c. JSON または XML 形式のメッセージを設定します。

d. **Send** をクリックします。

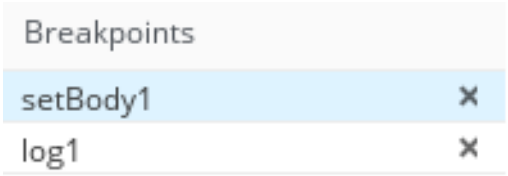
e. ルートの **Trace** タブに戻り、ルートを介したメッセージのフローを確認します。

4.8.4. ルートのデバッグ

1. **Camel** タブのツリービューで、ルートを選択します。
2. **Debug** を選択し、**Start debugging** をクリックします。
3. ブレークポイントを追加するには、図のノードを選択し、**Add breakpoint** をクリックします。ノードに赤い点が表示されます。



ノードがブレークポイントのリストに追加されます。



4. 下矢印をクリックして次のノードに移動するか、**Play** ボタンをクリックしてルートの実行を再開します。
5. **Pause** ボタンをクリックして、ルートのすべてのスレッドを一時停止します。
6. 終了したら **Stop debugging** をクリックします。すべてのブレークポイントが消去されます。

4.9. JMX ドメインおよび MBEAN の表示および管理

JMX (Java Management Extensions) は、実行時にリソース (サービス、デバイス、およびアプリケーション) を動的に管理できる Java 技術です。リソースは MBean (Managed Bean) と呼ばれるオブジェクトで表現されます。リソースが作成、実装、またはインストールされると即座に管理することができます。

Fuse Console で JMX プラグインを使用すると、JMX ドメインと MBean を表示および管理できます。MBean 属性の表示、コマンドの実行、および MBean の統計を示すチャートの作成を行うことができます。

JMX タブは、フォルダーに整理されたアクティブな JMX ドメインと MBean のツリービューを提供します。詳細を確認し、MBean でコマンドを実行できます。

手順

1. MBean 属性を表示および編集するには、以下を行います。

a. ツリービューで MBean を選択します。

b. **Attributes** タブをクリックします。

c. 属性をクリックしてその詳細を表示します。

2. 操作を実行するには、以下を行います。

a. ツリービューで MBean を選択します。

b. **Operations** タブをクリックし、リストにある操作の 1 つを展開します。

c. **Execute** をクリックし、操作を実行します。

3. チャートを表示するには、以下を行います。

a. ツリービューで項目を選択します。

b. **Chart** タブをクリックします。

4.10. QUARTZ スケジュールの表示および管理

Quartz(http://www.quartz-scheduler.org/) は、ほとんどの Java アプリケーション内で統合できる、機能豊富なオープンソースジョブスケジューリングライブラリーです。Quartz を使用して、ジョブを実行するための単純または複雑なスケジュールを作成できます。ジョブは、プログラムするほとんどすべてのものを実行できる標準の Java コンポーネントとして定義されます。

Camel ルートが **camel-quartz2** コンポーネントをデプロイすると、Fuse Console には **Quartz** タブが表示されます。JMX ツリービューから Quartz mbeans に交互にアクセスできることに注意してください。

手順

1. Fuse Console で **Quartz** タブをクリックします。
Quartz ページには、Quartz スケジューラーのツリービューならびに**Scheduler**、**Triggers**、および **Jobs** タブが含まれます。

2. スケジューラーを一時停止または開始するには、**Scheduler** タブのボタンをクリックします。

3. **Triggers** タブをクリックして、ジョブを実行するタイミングを決定するトリガーを表示します。たとえば、トリガーは、一日の指定の時刻 (ミリ秒単位) に、指定した日数、または指定した回数もしくは特定の回数繰り返してジョブを開始するように指定できます。

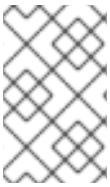
- トリガーの一覧をフィルターリングするには、ドロップダウンリストから **State**、**Group**、**Name**、または **Type** を選択します。続いて、入力フィールドに選択または入力することで、さらに一覧をフィルターできます。

- トリガーを一時停止、再開、更新、または手動で実行するには、**Action** 列のオプションをクリックします。

4. **Jobs** タブをクリックして実行中のジョブの一覧を表示します。テーブルの**Group**、**Name**、**Durable**、**Recover**、**Job ClassName**、および**Description** の列でリストをソートできます。

4.11. 診断の表示

Diagnostics タブを使用して、JVM DiagnosticCommand および HotspotDiangostic インターフェイスから JVM に関する診断情報を表示します。



注記

この機能は、Java Mission Control (jmc) の **Diagnostic Commands** ビューや、コマンドラインツールの jcmd と似ています。場合によっては、プラグインが対応する jcmd コマンドを提供します。

手順

1. ロードされたクラスのインスタンス数や、これらのインスタンスが使用するバイト数を取得するには、**Class Histogram** をクリックします。操作が繰り返し行われると、最後の操作実行との差異がタブに表示されます。
2. JVM 診断フラグ設定を表示するには、**JVM flags** をクリックします。
3. 稼働中の JVM でもフラグ設定を変更できます。

関連情報

サポートされる JVM はプラットフォームによって異なります。詳細は以下を参照してください。

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

4.12. スレッドの表示

スレッドの状態を表示および監視できます。

手順

1. **Runtime** タブをクリックし、**Threads** サブタブをクリックします。**Threads** ページには、アクティブなスレッドと各スレッドのスタックトレースの詳細が表示されます。デフォルトでは、スレッドリストにはすべてのスレッドが ID 値が大きい順に表示されます。
2. ID 値が小さい順に表示するには、**ID** 列ラベルをクリックします。
3. 任意で、スレッドの状態 (例: **Blocked**) やスレッド名でリストを絞り込むことができます。
4. ロッククラス名やスレッドのフルスタックトレースなど、特定スレッドの詳細情報を表示するには、**Actions** 列で **More** をクリックします。

4.13. ログエントリーの表示

Logs タブで Red Hat Fuse のログエントリーを表示できます。

前提条件

Logs タブは、Java アプリケーションに Log MBean が含まれる場合に利用できます。

手順

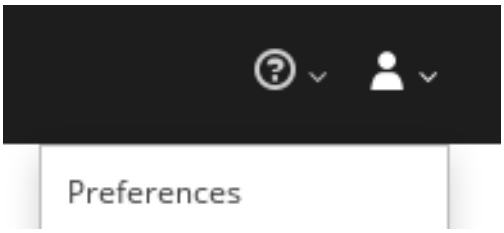
1. ログエントリーのリストを表示するには、**Log Entries** タブをクリックします。
   デフォルトでは、リストのログエントリーは昇順で表示されます。

   各ログエントリーをドリルダウンすると、ログエントリーに関する詳細情報が表示されます。

2. ログのリストを絞り込み、特定のログタイプを表示するには、**Action Bar** をクリックします。テキスト文字列またはログレベルを基にしてログエントリーセクションを絞り込むことができます。

3. Fuse Console のデフォルト設定を変更するには、以下を行います。

   a. Fuse Console の右上隅にあるユーザーアイコンをクリックして、ドロップダウンメニューの **Preferences** をクリックします。



   b. デフォルトのソート順序を変更するには、**Server Logs** を選択し、ログエントリーリンクをクリックして、バンドル名、スレッド、完全なメッセージテキストなどのログエントリーに関する詳細を表示します。

   c. 任意で、ログメッセージを格納するための設定をカスタマイズすることができます。

      - Fuse Console に保持するログステートメントの数 (デフォルトは 100)。
      - グローバルログレベル: **INFO** (デフォルト)、OFF、ERROR、WARN、および DEBUG
      - 子レベルのメッセージには、**hawtio-oauth** や **hawtio-core-utils** などが含まれます。

   d. Fuse Console のログ設定をデフォルトの値に戻すには、**Reset → Reset settings** とクリックします。

付録A FUSE CONSOLE 設定プロパティー

デフォルトでは、Fuse Console 設定は **hawtconfig.json** ファイルで定義されます。タイトル、ロゴ、ログインページの情報などの Fuse Console 設定情報をカスタマイズできます。

表A.1「Fuse Console 設定プロパティー」には、プロパティーの説明と、各プロパティーに値が必要かどうかが示されています。

表A.1 Fuse Console 設定プロパティー

セクション	プロパティー名	デフォルト値	説明	必須/任意
about	Title	Red Hat Fuse Management Console	Fuse Console の About ページに表示されるタイトル。	必要
	productInfo	**空の値**	Fuse Console の About ページに表示される製品情報。	任意
	additionalInfo	**空の値**	Fuse Console の About ページに表示される追加情報。	任意
	copyright	**空の値**	Fuse Console の About ページに表示される著作権情報。	任意
	imgSrc	**img/Logo-RedHat-A-Reverse-RGB.png**	Fuse Console の About ページに表示されるイメージ。	必要
branding	appName	Red Hat Fuse Management Console	アプリケーションの名前。この名前は Fuse Console のタイトルバーに表示されます。	必要
	appLogoUrl	**img/Logo-Red_Hat-Fuse-A-Reverse-RGB.png**	Fuse Console }navigation バーに表示されるアプリケーションロゴイメージファイルへのパス。この値は、Hawtio ステータス URL を基準とする相対パスまた絶対 URL です。	必要

セクション	プロパティー名	デフォルト値	説明	必須/任意
	Css		アプリケーションのスタイルに使用できる外部 CSS スタイルシートの URL。これは、Hawtio ステータス URL を基準とする相対パス、または絶対 URL を指定可能です。	任意
	companyLogoUrl	**img/Logo-RedHat-A-Reverse-RGB.png**	会社のロゴイメージファイルへのパス。	必要
	Favicon		通常、Web ブラウザータブに表示される favicon の URL。これは、Hawtio ステータス URL を基準とする相対パス、または絶対 URL を指定可能です。	任意
login	description	**空の値**	Fuse Console の Login ページに表示される説明テキスト (例: **http://localhost:8181/hawtio**)。	任意
	links	[ ]	**"url"** と **"text"** のペアのアレイを指定し、ユーザーが詳細またはヘルプを取得できるページへの追加リンクを提供します。	オプション

セクション	プロパティー名	デフォルト値	説明	必須/任意
disabledRoutes	**なし**	[ ]	コンソールの特定のパス (プラグインなど) を無効にします。このセクションは変更しないでください。変更は、OpenShift 以外のディストリビューションではサポートされません。	任意