



Red Hat Fuse 7.11

Springboot スタンドアロンでの Fuse の管理

Fuse Console を使用した Fuse アプリケーションの管理

Red Hat Fuse 7.11 Springboot スタンドアロンでの 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

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

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	3
多様性を受け入れるオープンソースの強化	4
第1章 FUSE CONSOLE	5
第2章 SPRING BOOT 2.X の FUSE CONSOLE へのアクセス	6
第3章 FUSE CONSOLE ブランディングのカスタマイズ	8
第4章 FUSE CONSOLE のセキュア化	9
第5章 FUSE CONSOLE でデータが正しく表示されるよう確認	11
第6章 リモート FUSE アプリケーションへの接続	12
6.1. リモート JOLOKIA エージェントへの接続	12
6.2. データ移動設定の指定	12
6.3. JVM ランタイム情報の表示	13
第7章 APACHE CAMEL アプリケーションの表示および管理	14
7.1. コンテキストの開始、一時停止、または削除	14
7.2. CAMEL アプリケーションの詳細表示	14
7.3. CAMEL ルートリストの表示および CAMEL ルートとの対話	15
7.4. ルートのデバッグ	16
第8章 JMX ドメインおよび MBEAN の表示および管理	18
第9章 QUARTZ スケジュールの表示および管理	19
第10章 診断の表示	20
第11章 スレッドの表示	21
付録A FUSE CONSOLE 設定プロパティ	22

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://hawtio.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せん。
- 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、[Prometheus のドキュメント](#) を参照してください。

本ガイドの対象者は、Red Hat Fuse on JBoss EAP 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、[CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 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

第2章 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>

第3章 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

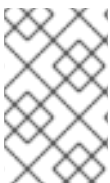
- ローカルの Fuse on Spring Boot スタンドアロンアプリケーションの **src/main/webapp** ディレクトリーに **hawtconfig.json** という名前の JSON ファイルを作成します。
- 任意のエディターで **src/main/webapp/hawtconfig.json** を開き、以下の内容を追加します。

```
{
  "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"
  ]
}
```

- 表A.1「Fuse Console 設定プロパティ」に記載されている設定プロパティの値を変更します。
- 変更を保存します。
- 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
```

- 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。

第4章 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 のプロキシサーバーレットを無効化**
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を **true** に設定して、Fuse Console のプロキシサーバーレットを無効にする必要があります。



注記

Fuse Console のプロキシサーバー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 のプロキシサーバー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。

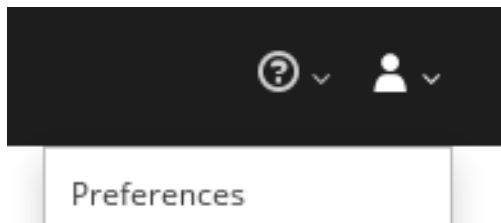
- **hawtio.http.publicKeyPins** プロパティの構文や、Base64 でエンコードされた公開鍵の抽出方法の説明は、[HTTP Public Key Pinning](#) 応答ヘッダーの説明ページを参照してください。

第5章 FUSE CONSOLE でデータが正しく表示されるよう確認

Fuse Console のキューおよび接続の表示で、不足しているキューや接続があったり、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6章 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 のプロキシサーバー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");
```

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。

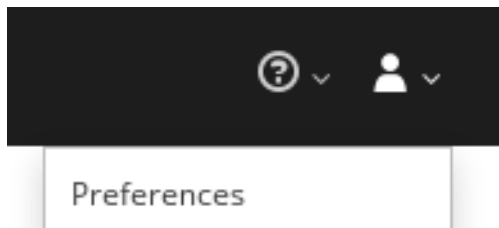
6.2. データ移動設定の指定

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。

6.3. JVM ランタイム情報の表示

システムプロパティ、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。

第7章 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に、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に接続する場合のみ使用できます。

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。

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 からのルートへの追加または更新、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。

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 フォルダをクリックします。

Routes

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

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。

Routes

Start Stop ⋮		
<input checked="" type="checkbox"/>	Name ^	Delete
<input type="checkbox"/>		



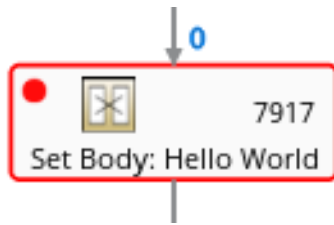
注記

- 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。

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。

Breakpoints	
setBody1	×
log1	×

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。

第8章 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。

第9章 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。

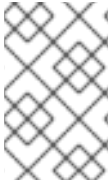
Camel ルートが **camel-quartz2** コンポーネントをデプロイすると、Fuse Console には **Quartz** タブが表示されます。JMX ツリービューから Quartz mbeans に交互にアクセスできることに注意してください。

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。

第10章 診断の表示

Diagnostics タブを使用して、JVM DiagnosticCommand および HotspotDiagnostic 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>

第11章 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。

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を示されています。

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は絶対 URL です。	必要

セクション	プロパティ名	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" のペアの配列を指定し、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。	任意

セクション	プロパティ名	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