



Red Hat Fuse 7.13

Fuse on Karaf スタンドアロンの管理

Fuse Console を使用した Fuse アプリケーションの管理

Red Hat Fuse 7.13 Fuse on Karaf スタンドアロンの管理

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 © 2024 Red Hat, Inc.

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

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

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

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

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

Linux[®] 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および操作できます。

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 FUSE CONSOLE へのアクセス	6
第3章 FUSE CONSOLE のセキュア化	7
第4章 ロールベースのアクセス参照	9
第5章 FUSE CONSOLE ブランディングのカスタマイズ	11
第6章 FUSE CONSOLE のデータ表示の修正	13
第7章 FUSE CONSOLE の無効化	14
第8章 リモート FUSE アプリケーションへの接続	15
8.1. FUSE CONSOLE のアンロック	15
8.2. リモートアクセスの制限	16
8.3. リモート FUSE インスタンスへの接続の許可	16
8.4. リモート JOLOKIA エージェントへの接続	16
8.5. データ移動設定の指定	17
8.6. JVM ランタイム情報の表示	17
第9章 APACHE CAMEL アプリケーションの表示および管理	19
9.1. コンテキストの起動、一時停止、または削除	19
9.2. CAMEL アプリケーションの詳細表示	19
9.3. CAMEL ルートリストの表示および CAMEL ルートとの対話	20
9.4. ルートのデバッグ	21
第10章 JMX ドメインおよび MBEAN の表示および管理	23
第11章 QUARTZ スケジュールの表示および管理	24
第12章 OSGI 環境の表示および管理	25
第13章 診断の表示	26
第14章 スレッドの表示	27
第15章 ログエントリの表示	28
第16章 PROMETHEUS メトリックの有効化	29
16.1. スタンドアロン APACHE KARAF コンテナからのメトリクスのエクスポートを有効にする	29
16.2. APACHE KARAF コンテナから公開済みメトリクスをスクレイピングするように PROMETHEUS サーバーを設定する	30
付録A FUSE CONSOLE 設定プロパティ	31

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/>) をベースにしています。このガイドでは 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ックを保存します。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。

このガイドの対象者は、Red Hat Fuse on Apache Karaf の管理者です。このガイドは、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に精通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 でサポートされる構成](#) 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ための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、およびサービスの起動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するか、Fuse on OpenShift を使用する場合に利用できます。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 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、以下の手順に従います。

前提条件

Karaf コンテナに Fuse をインストールします。ステップごとの手順は、[Apache Karaf へのインストール](#) 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。

第3章 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 のプロキシサーバーレットを無効にする**
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。

- **HTTPS を必須プロトコルとして設定する**
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な 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。このプロパティにより、HTTP を使用して Fuse Console へのアクセスを試みる Web ブラウザーで、HTTPS を使用するようにリクエストを自動的に変換することが必須になります。
- **公開鍵を使用して応答をセキュアにする**
hawtio.http.publicKeyPins プロパティを使用すると、特定の暗号化の公開鍵を Fuse Console に関連付けるように Web ブラウザーに指示し、偽造証明書による "中間者攻撃" のリスクを軽減することで、HTTPS プロトコルをセキュアにできます。
- **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**
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4.1「[Karaf スタンドアロンでのロールベースアクセス](#)」を参照してください。

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
```

- (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 **\$KARAF_HOME/etc/system.properties** ファイルで **hawtio.disableProxy** プロパティを **true** に設定します。

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

- 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「Apache Karaf コンテナで Undertow の SSL/TLS を有効にする」セクションを参照してください。
- 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 アプリケーションおよびサービスのセキュリティー保護ガイド](#) の「Hawtio 管理コンソールのセキュリティー保護」セクションを参照してください。
- 以下の手順に従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。
 - エディターで Red Hat Fuse **etc/users.properties** ファイルを開きます。
 - 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
```
 - ファイルを保存します。

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4章 ロールベースのアクセス参照

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4.1「Karaf スタンドアロンでのロールベースアクセス」を参照してください。

表4.1 Karaf スタンドアロンでのロールベースアクセス

操作	admin	manager	viewer
ログイン/ログアウト	可能	可能	可能
ヘルプトピックの表示	可能	可能	可能
ユーザー設定の指定	可能	可能	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	可能	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	可能	可能
Camel			
実行中の Camel アプリケーションをすべて表示	可能	可能	可能
Camel コンテキストの起動、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	可能	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	可能	可能

操作	admin	manager	viewer
操作の表示	可能	可能	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	可能	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			
システムプロパティ、メトリック、およびスレッドの表示	可能	可能	可能
Logs			
ログの表示	可能	可能	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。

第5章 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オーバーライドできます。

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」に記載されている設定プロパティ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** ファイルを開き、**<parent>** セクションに移動します。

```
<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

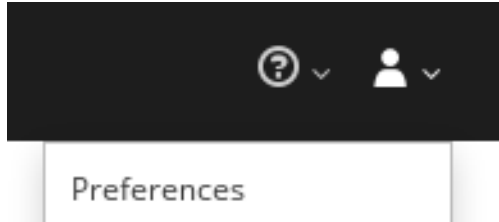
Web ブラウザーで Fuse Console をすでに実行した場合、ブランディングはブラウザーのローカルストレージに保存されます。新しいブランディング設定を使用するには、ブラウザーのローカルストレージをクリアする必要があります。

第6章 FUSE CONSOLE のデータ表示の修正

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は、Jolokia コレクションサイズパラメーターを調節します。このパラメーターは、Jolokia が応答でマーシャルするアレイの最大要素数を指定するものです。

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7章 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 にアクセスできなくなります。

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「ライフサイクル管理」の章を参照してください。

第8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 Java Management Extensions (JMX) へのエージェントベースのアプローチであり、クライアントへの追加ソフトウェア (エージェント) のインストールを必要とします。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。

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。

localhost または 127.0.0.1 以外のホスト名または IP アドレスに対して Fuse Console のロックを解除するには、次の手順に従います。

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。

8.2. リモートアクセスの制限

必要に応じて、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。

8.3. リモート FUSE インスタンスへの接続の許可

Fuse Console のプロキシーサーブレットはホワイトリストを使用してホストを保護し、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
```

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。

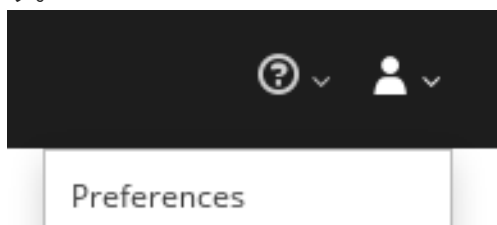
8.5. データ移動設定の指定

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** - オブジェクトを返す前に、Jolokia がサーバー側でオブジェクトを JSON にマーシャルする際のレベル数 (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。

8.6. JVM ランタイム情報の表示

システムプロパティ、メトリック、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9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カル表示。

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。

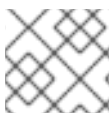
- コンテキストの起動および一時停止。
- 再起動、停止、一時停止、再開などを実行できるように、すべての Camel アプリケーションとそれらのルートのライフサイクルを管理すること。
- 実行中のルートのライブトレースおよびデバッグ。
- Camel エンドポイントへのメッセージの閲覧および送信。

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。

9.1. コンテキストの起動、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。

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カルに表示するには、**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。

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起動、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 checked="" type="checkbox"/>		



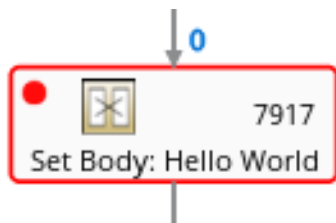
注記

- 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起動、停止、または削除することもできます。

3. ルートのグラフィカル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カルに表示するには、**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。

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。

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10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。リソースは、作成、実装、またはインストール後、すぐに管理および監視できます。

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つをデプロイメントします。
 - 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11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、標準の Java コンポーネントとして定義され、プログラムで実行するほとんどの処理を実行できます。

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と、**Scheduler**、**Triggers**、および **Jobs** タブがあります。
2. スケジューラーを一時停止または起動するには、**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、さらにリスト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リストを表示します。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12章 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あります。

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、バンドルの起動および停止、およびバンドルプロパティ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、ドリルダウンしてサービスの詳細を表示できます。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 OSGi フレームワークの設定オプション。フレームワークの起動レベルと初期バンドルの起動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。

第13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た場合、最後の操作実行との差異がタブに表示されます。
2. JVM 診断の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14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。

第15章 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。

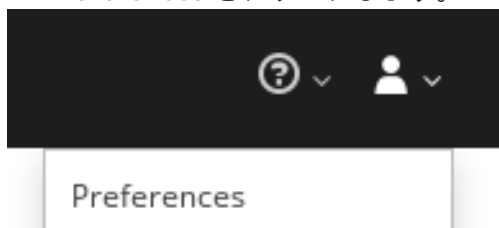
手順

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。

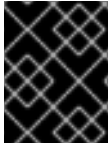
- a. Fuse Console の右上隅にあるユーザーアイコンをクリックして、ドロップダウンメニューの **Preferences** をクリックします。



- 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. 必要に応じて、ログメッセージの保存に関する次の設定をカスタマイズできます。
 - 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。
 - **hawtio-oauth** や **hawtio-core-utils** など、含める子レベルのメッセージ。
- 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。

第16章 PROMETHEUS メトリックの有効化

Prometheus は、スタンドアロン Apache Karaf コンテナ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、設定されたサービスから一定の間隔で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 "エクスポーター" 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、メトリックのリストとメトリック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、データをグラフィカルに可視化してクエリーを実行できるようにデータを格納します。

16.1. スタンドアロン APACHE KARAF コンテナからのメトリクスのエクスポートを有効にする

Prometheus は、Camel によって公開されるメトリック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

識別できるメトリックは、JMX で提供されるメトリックに限定されます。

Apache Camel メトリックを生成するには、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 コンテナから Prometheus メトリック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、公開されたメトリックを表示できます。



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。

16.2. APACHE KARAF コンテナから公開済みメトリクスをスクレイピングするように PROMETHEUS サーバーを設定する

Prometheus サーバーが Apache Karaf コンテナからメトリクスをスクレイピング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. スクレイピングする 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']
```

付録A FUSE CONSOLE 設定プロパティ

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」には、プロパティの説明と、各プロパティ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

セクション	プロパティ名	デフォルト値	説明	必須/オプション
about	Title	Red Hat Fuse Management Console	Fuse Console の About ページに表示されるタイトル。	必須
	productInfo	空の値	Fuse Console の About ページに表示される製品情報。	オプション
	additionalInfo	空の値	Fuse Console の About ページに表示される追加情報。	オプション
	copyright	空の値	Fuse Console の About ページに表示される著作権情報。	オプション
	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。	必須
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。	必須

セクション	プロパティ名	デフォルト値	説明	必須/オプション
	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。	オプション
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。	オプション