



Red Hat Fuse 7.6

Fuse 관리

Fuse 콘솔을 사용하여 Fuse 애플리케이션 관리

Red Hat Fuse 7.6 Fuse 관리

Fuse 콘솔을 사용하여 Fuse 애플리케이션 관리

법적 공지

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.

초록

Fuse 애플리케이션을 배포할 때 Fuse Console을 사용하여 Red Hat Fuse 통합을 모니터링하고 상호 작용할 수 있습니다.

차례

머리말	4
1장. OPENSIFT에서 RED HAT FUSE 애플리케이션 모니터링 및 관리	5
1.1. FUSE 콘솔 정보	5
1.2. OPENSIFT 4.X에서 FUSE 콘솔 설정	5
1.3. OPENSIFT 3.11에서 FUSE 콘솔 설정	11
1.4. 컨테이너 및 애플리케이션 보기	16
1.5. APACHE CAMEL 애플리케이션 보기 및 관리	16
1.6. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리	19
1.7. 진단 보기	20
1.8. 스택드 보기	20
1.9. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.	20
2장. SPRING BOOT에서 RED HAT FUSE 애플리케이션 모니터링 및 관리	22
2.1. FUSE 콘솔 정보	22
2.2. SPRING BOOT 2.X용 FUSE 콘솔에 액세스	22
2.3. SPRING BOOT 1.X용 FUSE 콘솔에 액세스	24
2.4. FUSE 콘솔 브랜딩 사용자 정의	25
2.5. FUSE CONSOLE 보안	26
2.6. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.	27
2.7. 원격 FUSE 애플리케이션에 연결	27
2.8. APACHE CAMEL 애플리케이션 보기 및 관리	29
2.9. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리	32
2.10. 진단 보기	33
2.11. 스택드 보기	33
3장. KARAF 독립 실행형에서 RED HAT FUSE 애플리케이션 모니터링 및 관리	34
3.1. FUSE 콘솔 정보	34
3.2. FUSE CONSOLE에 액세스	34
3.3. FUSE CONSOLE 보안	34
3.4. 역할 기반 액세스 참조	36
3.5. FUSE 콘솔 브랜딩 사용자 정의	37
3.6. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.	39
3.7. FUSE 콘솔 비활성화	39
3.8. 원격 FUSE 애플리케이션에 연결	39
3.9. APACHE CAMEL 애플리케이션 보기 및 관리	42
3.10. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리	45
3.11. OSGI 환경 보기 및 관리	46
3.12. 진단 보기	47
3.13. 스택드 보기	47
3.14. 로그 항목 보기	47
4장. EAP 독립 실행형에서 RED HAT FUSE 애플리케이션 모니터링 및 관리	49
4.1. FUSE 콘솔 정보	49
4.2. FUSE CONSOLE에 액세스	49
4.3. FUSE 콘솔 브랜딩 사용자 정의	50
4.4. FUSE CONSOLE 보안	51
4.5. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.	52
4.6. FUSE 콘솔 비활성화	52
4.7. 원격 FUSE 애플리케이션에 연결	53
4.8. APACHE CAMEL 애플리케이션 보기 및 관리	57
4.9. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리	62

4.10. 진단 보기	64
4.11. 스택드 보기	64
4.12. 로그 항목 보기	65
부록 A. FUSE 콘솔 구현 구성 속성	67

머리말

Red Hat Fuse는 Fuse 통합 보기 및 관리를 위한 두 가지 엔터프라이즈 모니터링 툴을 제공합니다.

- Fuse 콘솔은 실행 중인 Fuse 컨테이너를 모니터링하고 관리하기 위해 브라우저에서 액세스할 수 있는 웹 기반 콘솔입니다. Fuse 콘솔은 Hawtio 오픈 소스 소프트웨어(<https://hawt.io/>)를 기반으로 합니다. 이 가이드에서는 Fuse Console 사용 방법을 설명합니다.
- Prometheus(<https://prometheus.io/docs/introduction/overview/>)는 Fuse 배포에 대한 시스템 및 통합 수준 지표를 저장합니다. Grafana와 같은 그래픽 분석 인터페이스를 사용하여 저장된 기록 데이터를 보고 분석할 수 있습니다. Prometheus 사용에 대한 자세한 내용은 [Red Hat 통합 모니터링](#) 을 참조하십시오.

이 가이드의 대상은 Red Hat Fuse 관리자입니다. 이 가이드에서는 Red Hat Fuse 플랫폼, Apache Camel 및 조직의 처리 요구 사항을 잘 알고 있다고 가정합니다.

1장. OPENSIFT에서 RED HAT FUSE 애플리케이션 모니터링 및 관리

1.1. FUSE 콘솔 정보

Red Hat Fuse Console은 Hawtio 오픈 소스 소프트웨어를 기반으로 하는 웹 콘솔입니다. 지원되는 브라우저 목록은 [지원되는 구성으로 이동합니다](#).

Fuse Console은 배포된 하나 이상의 Fuse 컨테이너의 세부 정보를 검사하고 관리하는 중앙 인터페이스를 제공합니다. 또한 Red Hat Fuse 및 시스템 리소스를 모니터링하고 업데이트를 수행하며 서비스를 시작하거나 중지할 수 있습니다.

Red Hat Fuse 독립 실행형을 설치하거나 OpenShift에 Fuse를 사용하는 경우 Fuse Console을 사용할 수 있습니다. Fuse 콘솔에서 보고 관리할 수 있는 통합은 실행 중인 플러그인에 따라 다릅니다. 가능한 플러그인은 다음과 같습니다.

- Camel
- JMX
- Spring Boot
- OSGI
- 런타임
- 로그

1.2. OPENSIFT 4.X에서 FUSE 콘솔 설정

OpenShift 4.x에서 Fuse 콘솔을 설정하려면 보안, 설치 및 배포가 필요합니다.

먼저 [1.2.1절. "OpenShift 4.x에서 Fuse Console을 보호하기 위한 인증서 생성"](#)에 설명된 대로 Fuse 콘솔을 보호할 수 있도록 클라이언트 인증서를 생성해야 합니다.

클라이언트 인증서를 생성한 후 Fuse Console을 설치하고 배포하기 위한 다음과 같은 옵션이 있습니다.

- [1.2.2절. "OperatorHub를 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포"](#)
Fuse Console Operator를 사용하여 특정 네임스페이스의 Fuse 애플리케이션에 액세스할 수 있도록 Fuse Console을 설치하고 배포할 수 있습니다.
- [1.2.3절. "명령줄을 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포"](#)
명령줄과 Fuse Console 템플릿 중 하나를 사용하여 Fuse Console을 설치하고 배포하여 OpenShift 클러스터 또는 특정 네임스페이스에 있는 여러 네임스페이스의 Fuse 애플리케이션에 액세스할 수 있습니다.



참고

- Fuse Console의 사용자 관리는 OpenShift에서 처리합니다.
- 배포 후 Fuse Console에 액세스하는 사용자의 경우 OpenShift에서 Fuse를 사용할 수 없습니다.

1.2.1. OpenShift 4.x에서 Fuse Console을 보호하기 위한 인증서 생성

OpenShift 4.x에서는 Fuse Console 프록시와 Jolokia 에이전트 간에 안전하게 연결을 유지하려면 Fuse Console을 배포하기 전에 클라이언트 인증서를 생성해야 합니다. 서비스 서명 인증 기관 개인 키를 사용하여 클라이언트 인증서에 서명해야 합니다.



중요

각 OpenShift 클러스터에 대해 별도의 클라이언트 인증서를 생성하고 서명해야 합니다. 두 개 이상의 클러스터에 동일한 인증서를 사용하지 마십시오.

사전 요구 사항

- OpenShift 클러스터에 대한 **클러스터 관리자** 액세스 권한이 있어야 합니다.
- 두 개 이상의 OpenShift 클러스터에 대한 인증서를 생성하고 이전에 현재 디렉터리에 다른 클러스터에 대한 인증서를 생성한 경우 다음 중 하나를 수행하여 현재 클러스터에 대한 다른 인증서를 생성하십시오.
 - 현재 디렉터리에서 기존 인증서 파일(예: **ca.crt,ca.key, ca.srl**)을 삭제합니다.
 - 다른 작업 디렉터리로 변경합니다. 예를 들어 현재 작업 디렉터리 이름이 **cluster1** 인 경우 새 **cluster2** 디렉터리를 생성하고 작업 디렉터리를 해당 디렉터리로 변경합니다.

```
mkdir ../cluster2
```

```
cd ../cluster2
```

절차

1. 클러스터 관리자 액세스 권한이 있는 사용자로 OpenShift에 로그인합니다.

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

2. 다음 명령을 실행하여 서비스 서명 인증 기관 키를 검색합니다.

- 인증서를 검색하려면 다음을 수행합니다.

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

- 개인 키를 검색하려면 다음을 수행합니다.

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

3. **easysrsa,openssl** 또는 **cfssl** 을 사용하여 [Kubernetes 인증서 관리](#)에 설명된 대로 클라이언트 인증서를 생성합니다.

다음은 openssl을 사용하는 예제 명령입니다.

- a. 개인 키를 생성합니다.

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

- b. CSR 구성 파일을 작성합니다.

```
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
```

c. CSR을 생성합니다.

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

d. 서명된 인증서를 발급합니다.

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

다음 단계

Fuse Console 설치 방법에 따라 다음 섹션에 설명된 대로 Fuse Console의 시크릿을 생성하려면 이 인증서가 필요합니다.

- [Fuse Console Operator 사용](#)
- [명령줄 사용](#)

1.2.2. OperatorHub를 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포

OpenShift 4.x에 Fuse Console을 설치하려면 OpenShift OperatorHub에 제공된 Fuse Console Operator를 사용할 수 있습니다. Fuse 콘솔을 배포하려면 설치된 Operator의 인스턴스를 만듭니다.

사전 요구 사항

- OpenShift 클러스터에 대한 클러스터 관리자 액세스 권한이 있어야 합니다.
- OpenShift 4.x에서 Fuse 콘솔을 보호하기 위해 인증서 생성에 설명된 대로 Fuse Console의 클라이언트 인증서를 생성했습니다.

절차

Fuse 콘솔을 설치하고 배포하려면 다음을 수행합니다.

1. 웹 브라우저에서 클러스터 관리자 액세스 권한이 있는 사용자로 OpenShift 콘솔에 로그인합니다.
2. **홈** > 프로젝트를 선택한 다음 Fuse 콘솔을 추가할 프로젝트를 선택합니다.
3. **Operators** 를 클릭한 다음 **OperatorHub** 를 클릭합니다.
4. 검색 필드 창에서 **Fuse Console** 을 입력하여 **Operator** 목록을 필터링합니다.

5. Fuse Console Operator 를 클릭합니다.
6. Fuse Console Operator 설치 창에서 설치를 클릭합니다.
Create Operator Subscription양식이 열립니다.
 - 설치 모드 의 경우 Fuse Console Operator를 특정 네임스페이스(현재 OpenShift 프로젝트)에 설치합니다.
그런 다음 Operator를 설치한 후 Fuse Console을 배포하여 클러스터의 모든 네임스페이스에서 애플리케이션을 모니터링하거나 Fuse Console Operator가 설치된 네임스페이스에서만 애플리케이션을 모니터링하도록 선택할 수 있습니다.
 - 승인 전략 의 경우 자동 또는 수동 을 선택하여 OpenShift에서 Fuse Console Operator에 대한 업데이트를 처리하는 방법을 구성할 수 있습니다.
 - 자동 업데이트를 선택하면 새 버전의 Fuse Console Operator가 사용 가능할 때 OLM(Operator Lifecycle Manager)은 개입 없이 Fuse Console의 실행 중인 인스턴스를 자동으로 업그레이드합니다.
 - 수동 업데이트를 선택하면 최신 버전의 Operator가 사용 가능할 때 OLM에서 업데이트 요청을 생성합니다. 클러스터 관리자는 Fuse Console Operator가 새 버전으로 업데이트되도록 해당 업데이트 요청을 수동으로 승인해야 합니다.
7. Subscribe를 클릭합니다.
OpenShift는 현재 네임스페이스에 Fuse Console Operator를 설치합니다.
8. 설치를 확인하려면 Operator를 클릭한 다음 Installed Operators 를 클릭합니다. Operator 목록에서 Fuse Console을 볼 수 있습니다.
9. 터미널 창에서 [OpenShift 4.x에서 Fuse Console 보안에서 생성한 인증서](#)를 사용하여 시크릿을 생성하고, APP_NAME 이 Fuse Console 배포 이름(예: **fuse-console**)을 사용하여 Fuse 콘솔에 마운트합니다.

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

예를 들어 Fuse Console 애플리케이션의 이름이 **fuse-console** 인 경우 다음 명령을 입력합니다.

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

성공하면 이 명령은 보안이 생성되었는지 확인하는 응답을 반환합니다. 예를 들면 다음과 같습니다.

```
secret/fuse-console-operator-tls-proxying created
```

10. OpenShift 웹 콘솔을 사용하여 Fuse 콘솔을 배포하려면 다음을 수행합니다.
 - a. 설치된 Operator 목록에서 이름 옆에서 Fuse Console 을 클릭합니다.
 - b. 제공된 API 아래의 개요 페이지에서 인스턴스 생성을 클릭합니다. 새 CRD(Custom Resource Definition) 파일이 열립니다.
기본적으로 Fuse Console은 현재 네임스페이스에 배포됩니다.
 - c. Fuse Console을 배포하여 현재 네임스페이스 내에서 애플리케이션을 검색하고 관리하려면 다음 단계로 건너뛴니다.
선택적으로 클러스터(및 인증된 사용자)에 있는 모든 네임스페이스에서 애플리케이션을 검색하고 관리하려면 **spec.type** 필드의 값을 네임스페이스 에서 클러스터로 변경하여 CRD 파일을 편집합니다.

- d. 생성을 클릭합니다.
Fuse Console CRD 페이지가 열리고 새 Fuse Console 서비스가 표시됩니다.
11. Fuse 콘솔을 열려면 다음을 수행합니다.
 - a. Fuse 콘솔 URL을 가져옵니다.
 - OpenShift 웹 콘솔 내에서 네트워킹 > 경로를 선택합니다.
 - 명령줄에서 `oc get routes` 명령을 입력합니다.
 - b. 웹 브라우저에서 Fuse Console URL을 연 다음 로그인합니다.
필요한 권한이 나열된 브라우저에서 권한 부여 페이지가 열립니다.
 - c. 선택한 권한 허용을 클릭합니다.
브라우저에서 Fuse Console이 열리고 액세스할 수 있는 권한이 있는 Fuse 애플리케이션 포드가 표시됩니다.
 12. 확인할 애플리케이션에 대한 연결을 클릭합니다.
Fuse Console에 애플리케이션이 표시되는 새 브라우저 창이 열립니다.

1.2.3. 명령줄을 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포

OpenShift 4.x에서는 명령줄에서 Fuse 콘솔을 설치하고 배포할 다음 배포 옵션 중 하나를 선택할 수 있습니다.

- 클러스터 - Fuse Console은 OpenShift 클러스터의 여러 네임스페이스(프로젝트)에 배포된 Fuse 애플리케이션을 검색하고 연결할 수 있습니다. 이 템플릿을 배포하려면 OpenShift 클러스터에 대한 관리자 역할이 있어야 합니다.
- 네임스페이스 - Fuse Console은 특정 OpenShift 프로젝트(네임스페이스)에 액세스할 수 있습니다. 이 템플릿을 배포하려면 OpenShift 프로젝트에 대한 관리자 역할이 있어야 합니다.

Fuse Console 템플릿의 매개변수 목록을 보려면 다음 OpenShift 명령을 실행합니다.

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-760043-redhat-00003/fuse-console-namespace-os4.json
```

사전 요구 사항

- Fuse 콘솔을 설치하고 배포하기 전에 OpenShift 4.x에서 Fuse Console을 보호하기 위해 인증서 생성에 설명된 대로 서비스 서명 인증 기관으로 서명된 클라이언트 인증서를 생성해야 합니다.
- OpenShift 클러스터에 대한 클러스터 관리자 역할이 있어야 합니다.
- OpenShift 4.x 서버에 Fuse 이미지 스트림 설치 및 템플릿에 설명된 대로 Fuse 콘솔 이미지 스트림 (다른 Fuse 이미지 스트림과 함께)이 설치됩니다.

절차

1. 다음 명령을 사용하여 모든 템플릿 목록을 검색하여 Fuse Console 이미지 스트림이 설치되었는지 확인합니다.

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

- 선택적으로 이미 설치된 이미지 스트림을 새 릴리스 태그로 업데이트하려면 다음 명령을 사용하여 Fuse Console 이미지를 openshift 네임스페이스로 가져옵니다.

```
oc import-image fuse7/fuse7-console:1.6 --from=registry.redhat.io/fuse7/fuse-console:1.6 --confirm -n openshift
```

- 다음 명령을 실행하여 Fuse Console APP_NAME 값을 가져옵니다.

- 클러스터 템플릿의 경우:

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-760043-redhat-00003/fuse-console-cluster-os4.json
```

- 네임스페이스 템플릿의 경우:

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-760043-redhat-00003/fuse-console-namespace-os4.json
```

- OpenShift 4.x에서 Fuse Console 보안에서 생성한 인증서에서 다음 명령을 사용하여 시크릿을 생성하고 Fuse Console에 마운트합니다(여기서 APP_NAME 은 Fuse Console 애플리케이션의 이름입니다).

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

- 다음 명령을 실행하여 새 애플리케이션을 생성합니다(\$project 는 OpenShift 프로젝트의 이름, \$APP_NAME 은 위 단계에서 얻은 애플리케이션의 이름이며 \$DOMAIN_NAME 은 Fuse Console 에 액세스할 수 있는 호스트 이름입니다).

- 클러스터 템플릿의 경우:

```
oc new-app -n $project -f {templates-base-url}/fuse-console-cluster-os4.json -p ROUTE_HOSTNAME=$APP_NAME.$DOMAIN_NAME -p APP_NAME=$APP_NAME
```

- 네임스페이스 템플릿의 경우:

```
oc new-app -n myproject -f {templates-base-url}/fuse-console-namespace-os4.json
```

- 다음 명령을 실행하여 Fuse Console 배포의 상태 및 URL을 가져옵니다.

```
oc status
```

- 브라우저에서 Fuse 콘솔에 액세스하려면 6단계에서 반환된 URL(예: <https://fuse-console.192.168.64.12.nip.io>)을 사용합니다.

1.2.4. OpenShift 4.x에서 Fuse Console 업그레이드

Red Hat OpenShift 4.x는 Red Hat Fuse Operator를 포함하여 Operator에 대한 업데이트를 처리합니다. 자세한 내용은 [Operator OpenShift 설명서](#)를 참조하십시오.

Operator 업데이트는 애플리케이션 구성 방법에 따라 애플리케이션 업그레이드를 트리거할 수 있습니다.

Fuse Console 애플리케이션의 경우 애플리케이션 사용자 정의 리소스 정의의 **.spec.version** 필드를 편집하여 애플리케이션으로의 업그레이드를 트리거할 수도 있습니다.

사전 요구 사항

- OpenShift 클러스터 관리자 권한이 있어야 합니다.

절차

Fuse Console 애플리케이션을 업그레이드하려면 다음을 수행합니다.

1. 터미널 창에서 다음 명령을 사용하여 애플리케이션 사용자 정의 리소스 정의의 **.spec.version** 필드를 변경합니다.

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

예를 들면 다음과 같습니다.

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

2. 애플리케이션 상태가 업데이트되었는지 확인합니다.

```
oc get myproject
```

응답에는 버전 번호를 포함하여 애플리케이션에 대한 정보가 표시됩니다.

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

.spec.version 필드의 값을 변경하면 OpenShift에서 애플리케이션을 자동으로 재배포합니다.

3. 버전 변경으로 트리거되는 재배포 상태를 확인하려면 다음을 수행합니다.

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

성공적인 배포에는 다음 응답이 표시됩니다.

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

1.3. OPENSIFT 3.11에서 FUSE 콘솔 설정

OpenShift 3.11에서는 다음 두 가지 방법으로 Fuse Console을 설정할 수 있습니다.

- 프로젝트에 중앙 집중식 Fuse Console 카탈로그 항목을 추가하여 프로젝트에서 실행 중인 모든 Fuse 컨테이너를 모니터링할 수 있습니다.
- 특정 포드에서 실행 중인 단일 Fuse 컨테이너를 모니터링할 수 있습니다.

OpenShift 콘솔 또는 명령줄에서 Fuse 콘솔을 배포할 수 있습니다.



참고

Minishift 또는 CDK 기반 환경에 Fuse 콘솔을 설치하려면 아래 KCS 문서에 설명된 단계를 따르십시오.

- Minishift 또는 CDK 기반 환경에 Fuse Console을 설치하려면 [KCS 4998441](#)을 참조하십시오.
- Jolokia 인증을 비활성화해야 하는 경우 [KCS 3988671](#)에 설명된 해결방법을 참조하십시오.

사전 요구 사항

- OpenShift의 Fuse에 설명된 대로 OpenShift 이미지 스트림에 Fuse 및 [Fuse 콘솔 템플릿을 설치합니다](#).
- OpenShift 3.11의 클러스터 모드인 경우 클러스터 관리자 역할 및 클러스터 모드 템플릿이 필요합니다. 다음 명령을 실행합니다.

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



참고

- 클러스터 모드 템플릿은 기본적으로 최신 버전의 OpenShift Container Platform에서만 사용할 수 있습니다. OpenShift Online 기본 카탈로그는 제공되지 않습니다.
- Fuse Console 템플릿은 브라우저에서 클러스터 내 서비스로의 보안 엔드 투 엔드 요청이 되도록 기본적으로 엔드 투 엔드 암호화를 구성합니다.
- Fuse Console의 사용자 관리는 OpenShift에서 처리합니다.
- 배포 후 Fuse Console에 액세스하는 사용자의 경우 OpenShift에서 Fuse를 사용할 수 없습니다.

1.3.1절. "OpenShift 3.11 콘솔에서 Fuse 콘솔 배포"

1.3.2절. "OpenShift 3.11의 Fuse Console에서 단일 Fuse Pod 모니터링"

1.3.3절. "명령줄에서 Fuse 콘솔 배포"

1.3.1. OpenShift 3.11 콘솔에서 Fuse 콘솔 배포

OpenShift 3.11 콘솔의 OpenShift 클러스터에 Fuse Console을 배포하려면 다음 단계를 따르십시오.

절차

1. OpenShift 콘솔에서 기존 프로젝트를 열거나 새 프로젝트를 생성합니다.
2. OpenShift 프로젝트에 Fuse 콘솔을 추가합니다.
 - a. Add to Project → Browse Catalog 를 선택합니다.
현재 프로젝트에 추가할 항목 선택 페이지가 열립니다.

- b. 검색 필드에 Fuse Console 을 입력합니다.
Red Hat Fuse 7.x 콘솔과 Red Hat Fuse 7.x 콘솔(cluster)항목이 검색 결과에 표시됩니다.



참고

Red Hat Fuse Console 항목이 검색 결과로 나타나지 않거나 표시되는 항목이 최신 버전이 아닌 경우 OpenShift Guide의 "OpenShift 서버 전" 섹션에 설명된 대로 수동으로 Fuse Console 템플릿을 설치할 수 있습니다.

- a. Red Hat Fuse Console 항목 중 하나를 클릭합니다.
- Red Hat Fuse 7.x 콘솔- 이 버전의 Fuse Console은 현재 OpenShift 프로젝트에 배포된 Fuse 애플리케이션을 검색하고 연결합니다.
 - Red Hat Fuse 7.x Console(클러스터)- 이 버전의 Fuse Console은 OpenShift 클러스터의 여러 프로젝트에 배포된 Fuse 애플리케이션을 검색하고 연결할 수 있습니다.
- b. Red Hat Fuse Console 마법사에서 다음을 클릭합니다. 마법사의 구성 페이지가 열립니다. 선택적으로 구성 매개변수의 기본값을 변경할 수 있습니다.
1. 생성을 클릭합니다.
마법사의 결과 페이지에는 Red Hat Fuse Console이 생성되었음을 나타냅니다.
 2. Continue to the project overview 링크를 클릭하여 Fuse Console 애플리케이션이 프로젝트에 추가되었는지 확인합니다.
 3. Fuse 콘솔을 열려면 제공된 URL 링크를 클릭한 다음 로그인합니다.
필요한 권한이 나열된 브라우저에서 권한 부여 페이지가 열립니다.
 4. 선택한 권한 허용을 클릭합니다.
브라우저에서 Fuse Console이 열리고 프로젝트에서 실행 중인 Fuse Pod가 표시됩니다.
 5. 확인할 애플리케이션에 대한 연결을 클릭합니다.
Fuse Console에 애플리케이션이 표시되는 새 브라우저 창이 열립니다.

1.3.2. OpenShift 3.11의 Fuse Console에서 단일 Fuse Pod 모니터링

OpenShift 3.11에서 실행되는 Fuse Pod의 Fuse Console을 열 수 있습니다.

1. OpenShift 프로젝트의 애플리케이션 → 포드 보기에서 Pod 이름을 클릭하여 실행 중인 Fuse Pod의 세부 정보를 확인합니다. 이 페이지 오른쪽에는 컨테이너 템플릿에 대한 요약이 표시됩니다.

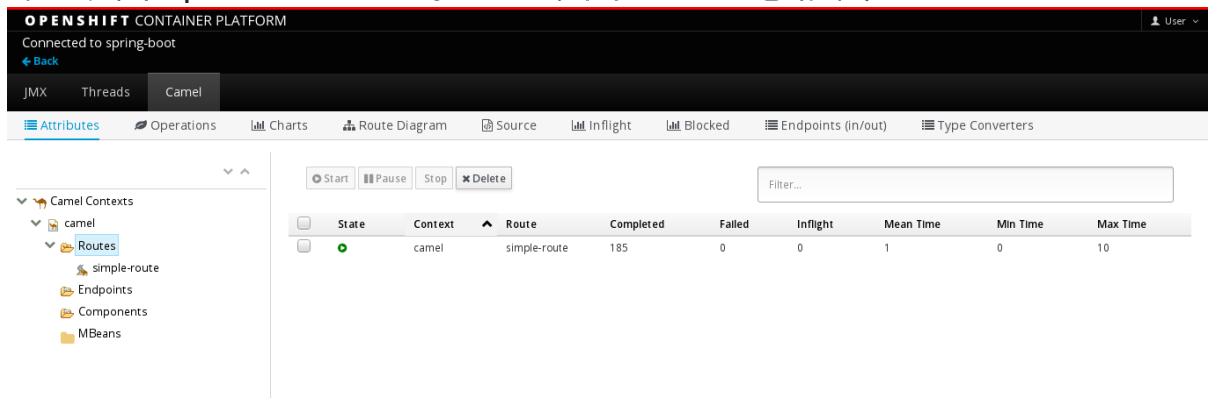
Template

Containers

CONTAINER: SPRING-BOOT

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

2. 이 보기에서 **Open Java Console** 링크를 클릭하여 Fuse 콘솔을 엽니다.



The screenshot shows the OpenShift Container Platform console interface. The top navigation bar includes 'Attributes', 'Operations', 'Charts', 'Route Diagram', 'Source', 'Inflight', 'Blocked', 'Endpoints (in/out)', and 'Type Converters'. The main content area displays the Camel route configuration for 'simple-route'. The route is in a 'Running' state, with 185 completed messages, 0 failed messages, and 0 messages in flight. The mean time for the route is 1 second, with a minimum time of 0 and a maximum time of 10 seconds.

State	Context	Route	Completed	Failed	Inflight	Mean Time	Min Time	Max Time
Running	camel	simple-route	185	0	0	1	0	10



참고

포드 뷰에 **Fuse Console**에 대한 링크를 표시하도록 OpenShift를 구성하려면 OpenShift 이미지에서 Fuse를 실행하는 Pod에서 **jolokia** 로 설정된 name 속성 내에서 TCP 포트를 선언해야 합니다.

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

1.3.3. 명령줄에서 Fuse 콘솔 배포

표 1.1. "Fuse 콘솔 템플릿" Fuse 애플리케이션 배포 유형에 따라 명령줄에서 Fuse Console을 배포하는 데 사용할 수 있는 OpenShift 3.1 템플릿을 설명합니다.

표 1.1. Fuse 콘솔 템플릿

유형	설명
fis-console-cluster-template.json	Fuse Console은 여러 네임스페이스 또는 프로젝트에 배포된 Fuse 애플리케이션을 검색하고 연결할 수 있습니다. 이 템플릿을 배포하려면 OpenShift cluster-admin 역할이 있어야 합니다.
fis-console-namespace-template.json	이 템플릿은 현재 OpenShift 프로젝트(네임스페이스)에 대한 Fuse Console 액세스를 제한하므로 단일 테넌트 배포 역할을 합니다. 이 템플릿을 배포하려면 현재 OpenShift 프로젝트에 대한 admin 역할이 있어야 합니다.

선택적으로 다음 명령을 실행하여 모든 템플릿에 대한 매개변수 목록을 볼 수 있습니다.

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-760043-redhat-00003/fis-console-namespace-template.json
```

절차

명령줄에서 Fuse 콘솔을 배포하려면 다음을 수행합니다.

1. 다음 명령 중 하나를 실행하여 Fuse Console 템플릿을 기반으로 새 애플리케이션을 생성합니다(여기서 **myproject**는 프로젝트 이름임).

- Fuse Console 클러스터 템플릿의 경우 **myhost**는 Fuse Console에 액세스할 수 있는 호스트 이름입니다.

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-760043-redhat-00003/fis-console-cluster-template.json -p ROUTE_HOSTNAME=myhost
```

- Fuse Console 네임스페이스 템플릿의 경우:

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-760043-redhat-00003/fis-console-namespace-template.json
```



참고

OpenShift가 자동으로 생성되므로 네임스페이스 템플릿의 `route_hostname` 매개변수를 생략할 수 있습니다.

2. 다음 명령을 실행하여 Fuse Console 배포의 상태 및 URL을 가져옵니다.

oc status

3. 브라우저에서 Fuse 콘솔에 액세스하려면 제공된 URL(예: <https://fuse-console.192.168.64.12.nip.io>)을 사용합니다.

1.4. 컨테이너 및 애플리케이션 보기

OpenShift용 Fuse Console에 로그인하면 Fuse Console 홈 페이지에 사용 가능한 컨테이너가 표시됩니다.

절차

- 컨테이너를 관리(생성, 편집 또는 삭제)하려면 OpenShift 콘솔을 사용합니다.
- OpenShift 클러스터에서 Fuse 애플리케이션을 보려면 온라인 탭을 클릭합니다.

1.5. APACHE CAMEL 애플리케이션 보기 및 관리

Fuse 콘솔의 Camel 탭에서 Apache Camel 컨텍스트, 경로 및 종속성을 보고 관리합니다.

다음 세부 정보를 볼 수 있습니다.

- 실행 중인 모든 Camel 컨텍스트 목록
- Camel 버전 번호 및 런타임 정적과 같은 각 Camel 컨텍스트에 대한 자세한 정보
- 각 Camel 애플리케이션의 모든 경로 목록 및 런타임 통계
- 실행 중인 경로의 그래픽 표현과 실시간 메트릭

다음은 통해 Camel 애플리케이션과 상호 작용할 수도 있습니다.

- 컨텍스트 시작 및 일시 중단
- 모든 Camel 애플리케이션 및 해당 경로의 라이프사이클을 관리하여 재시작, 중지, 일시 중지, 재개 등을 수행할 수 있습니다.
- 실행 중인 경로의 실시간 추적 및 디버깅
- Camel 엔드포인트에 메시지 검색 및 전송

사전 요구 사항

Camel 탭은 하나 이상의 Camel 경로를 사용하는 컨테이너에 연결할 때만 사용할 수 있습니다.

1.5.1. 컨텍스트 시작, 일시 중지 또는 삭제

1. Camel 탭의 트리 보기에서 Camel Contexts 를 클릭합니다.
2. 목록에서 하나 이상의 컨텍스트 옆에 있는 확인란을 선택합니다.
3. 시작 또는 일시 중지를 클릭합니다.
4. 컨텍스트를 삭제하려면 다음을 수행합니다.

- a. 컨텍스트를 중지합니다.
- b. 아이콘을 클릭한 다음 드롭다운 메뉴에서 삭제 를 선택합니다.



참고

컨텍스트를 삭제하면 배포된 애플리케이션에서 해당 컨텍스트를 제거합니다.

1.5.2. Camel 애플리케이션 세부 정보 보기

1. Camel 탭의 트리 뷰에서 Camel 애플리케이션을 클릭합니다.
2. 애플리케이션 특성 및 값 목록을 보려면 속성을 클릭합니다.
3. 애플리케이션 특성의 그래픽 표시를 보려면 차트를 클릭한 다음 편집을 클릭하여 차트에서 볼 속성을 선택합니다.
4. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
5. 애플리케이션 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향으로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 Camel 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 유형 다운로드를 클릭합니다.
7. XML에서 경로 추가 또는 업데이트 또는 classpath에서 사용 가능한 모든 Camel 구성 요소를 찾는 등 CRYostat 작업을 보고 실행하려면 Operations 를 클릭합니다.

1.5.3. Camel 경로 목록 보기 및 상호 작용

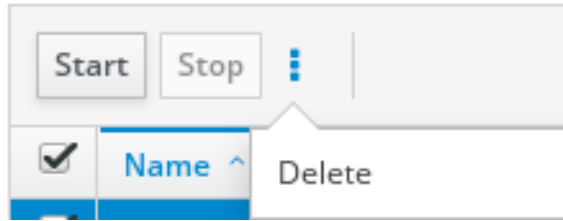
1. 경로 목록을 보려면 다음을 수행합니다.
 - a. Camel 탭을 클릭합니다.
 - b. 트리 뷰에서 애플리케이션의 경로 폴더를 클릭합니다.

Routes

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

2. 하나 이상의 경로를 시작, 중지 또는 삭제하려면 다음을 수행합니다.
 - a. 목록에서 하나 이상의 경로 옆에 있는 확인란을 선택합니다.
 - b. 시작 또는 중지 를 클릭합니다.
 - c. 경로를 삭제하려면 먼저 중지해야 합니다. 그런 다음 아이콘을 클릭하고 드롭다운 메뉴에서 삭제 를 선택합니다.

Routes

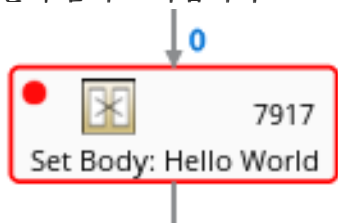


참고

- 경로를 삭제하면 배포된 애플리케이션에서 해당 경로를 제거합니다.
 - 트리 보기에서 특정 경로를 선택한 다음 오른쪽 상단 메뉴를 클릭하여 시작, 중지 또는 삭제할 수도 있습니다.
3. 경로의 그래픽 다이어그램을 보려면 **Route Diagram** 을 클릭합니다.
 4. 진행 중 및 차단된 교환을 보려면 **교환**을 클릭합니다.
 5. 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향별로 목록을 필터링할 수 있습니다.
 6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 **Camel** 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 **Type Cryostat**를 클릭합니다.
 7. 특정 경로와 상호 작용하려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 보기에서 경로를 선택합니다.
 - b. 경로 속성 및 값 목록을 보려면 속성을 클릭합니다.
 - c. 경로 속성의 그래픽 표시를 보려면 **차트** 를 클릭합니다. 편집을 클릭하여 차트에서 표시할 속성을 선택할 수 있습니다.
 - d. 진행 중 및 차단된 교환을 보려면 **교환**을 클릭합니다.
 - e. 작업에서 경로를 XML로 덤프하거나 경로의 **Camel ID** 값을 가져오는 등 경로에서 **Cryostat** 작업을 보고 실행합니다.
 8. 경로를 통해 메시지를 추적하려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 보기에서 경로를 선택합니다.
 - b. 추적 을 선택한 다음 추적 시작을 클릭합니다.
 9. 경로에 메시지를 보내려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 뷰에서 컨텍스트의 엔드포인트 폴더를 연 다음 엔드포인트를 선택합니다.
 - b. **Send** 를 클릭합니다.
 - c. 메시지를 **JSON** 또는 **XML** 형식으로 구성합니다.
 - d. 전송을 클릭합니다.
 - e. 경로의 추적 탭으로 돌아가 경로를 통한 메시지 흐름을 확인합니다.

1.5.4. 경로 디버깅

1. Camel 탭의 트리 보기에서 경로를 선택합니다.
2. 디버그를 선택한 다음 디버깅 시작을 클릭합니다.
3. Cryostat를 추가하려면 다이어그램에서 노드를 선택한 다음 Add Cryostat를 클릭합니다. 노드에 빨간색 점이 표시됩니다.



노드가 Cryostat 목록에 추가됩니다.

Breakpoints	
setBody1	×
log1	×

4. 아래쪽 화살표를 클릭하여 다음 노드 또는 재생 버튼으로 이동하여 경로 실행을 다시 시작합니다.
5. 일시 중지 버튼을 클릭하여 경로에 대한 모든 스레드를 일시 중지합니다.
6. 완료되면 디버깅 중지를 클릭합니다. 모든 Cryostat가 지워집니다.

1.6. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리

JMX(Java Management Extensions)는 런타임 시 리소스(서비스, 장치 및 애플리케이션)를 동적으로 관리할 수 있는 Java 기술입니다. 리소스는 Cryostats라는 오브젝트로 표시됩니다(관리됨의 경우). 리소스를 생성, 구현 또는 설치하는 즉시 리소스를 관리하고 모니터링할 수 있습니다.

Fuse Console에서 Cryostat 플러그인을 사용하면 Cryostat 도메인 및 Cryostat를 보고 관리할 수 있습니다. Cryostat 특성을 보고, 명령을 실행하고, 통계를 표시하는 차트를 만들 수 있습니다.

Cryostat 탭에서는 폴더에 구성된 활성 Cryostat 도메인 및 Cryostat의 트리 뷰를 제공합니다. Cryostat에서 세부 정보를 보고 명령을 실행할 수 있습니다.

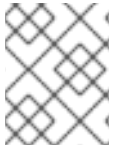
절차

1. Cryostat 특성을 보고 편집하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. 특성 탭을 클릭합니다.
 - c. 속성을 클릭하여 세부 정보를 확인합니다.
2. 작업을 수행하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. Operations 탭을 클릭하고 나열된 작업 중 하나를 확장합니다.

- c. 실행을 클릭하여 작업을 실행합니다.
3. 차트를 보려면 다음을 수행합니다.
 - a. 트리 뷰에서 항목을 선택합니다.
 - b. 차트 탭을 클릭합니다.

1.7. 진단 보기

JVM Diagnostic Command 및 CryostatDiagnostic 인터페이스를 통해 JVM에 대한 진단 정보를 보려면 Cryostat 탭을 사용합니다.



참고

기능은 jmc(Java Mission Control) 또는 명령줄 도구 jcmd의 진단 명령 보기와 유사합니다. 플러그인은 일부 시나리오에서 해당 jcmd 명령을 제공합니다.

절차

1. 로드된 클래스의 인스턴스 수와 필요한 바이트 수를 검색하려면 클래스 histogram을 클릭합니다. 작업이 반복되면 탭에 마지막 실행 이후의 차이점이 표시됩니다.
2. JVM 진단 플래그 설정을 보려면 JVM 플래그를 클릭합니다.
3. 실행 중인 JVM의 경우 플래그 설정도 수정할 수 있습니다.

추가 리소스

지원되는 JVM은 플랫폼에 따라 다릅니다. 자세한 내용은 다음 소스 중 하나로 이동합니다.

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

1.8. 스레드 보기

스레드 상태를 보고 모니터링할 수 있습니다.

절차

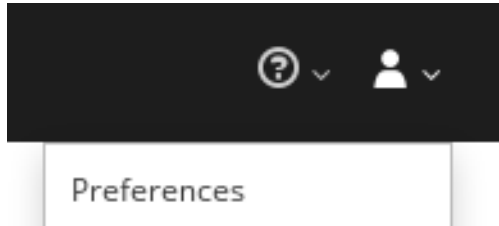
1. Runtime 탭을 클릭한 다음 Threads 를 클릭합니다. Threads 페이지에는 각 스레드에 대한 활성 스레드 및 스택 추적 세부 정보가 나열됩니다. 기본적으로 스레드 목록은 모든 스레드를 내림차순으로 표시합니다.
2. ID를 눌러 목록을 정렬하려면 ID 열 레이블을 클릭합니다.
3. 선택적으로 스레드 상태(예: Blocked) 또는 스레드 이름으로 목록을 필터링합니다.
4. 해당 스레드의 잠금 클래스 이름 및 전체 스택 추적과 같은 특정 스레드에 대한 자세한 정보를 드릴 다운하려면 Actions 열에서 More 를 클릭합니다.

1.9. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.

Fuse Console의 대기열 및 연결 표시에 큐가 없거나 연결이 누락되었거나 일관성 없는 아이콘을 표시하는 경우 Jolokia 컬렉션 크기 매개변수를 조정하여 Jolokia marshals를 응답에 마샬링하는 배열의 최대 요소 수를 지정합니다.

절차

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 최대 컬렉션 크기 옵션의 값을 늘립니다(기본값은 50,000).
3. 닫기를 클릭합니다.

2장. SPRING BOOT에서 RED HAT FUSE 애플리케이션 모니터링 및 관리

2.1. FUSE 콘솔 정보

Red Hat Fuse Console은 Hawtio 오픈 소스 소프트웨어를 기반으로 하는 웹 콘솔입니다. 지원되는 브라우저 목록은 [지원되는 구성으로 이동합니다](#).

Fuse Console은 배포된 하나 이상의 Fuse 컨테이너의 세부 정보를 검사하고 관리하는 중앙 인터페이스를 제공합니다. 또한 Red Hat Fuse 및 시스템 리소스를 모니터링하고 업데이트를 수행하며 서비스를 시작하거나 중지할 수 있습니다.

Red Hat Fuse 독립 실행형을 설치하거나 OpenShift에 Fuse를 사용하는 경우 Fuse Console을 사용할 수 있습니다. Fuse 콘솔에서 보고 관리할 수 있는 통합은 실행 중인 플러그인에 따라 다릅니다. 가능한 플러그인은 다음과 같습니다.

- Camel
- JMX
- Spring Boot
- OSGI
- 런타임
- 로그

2.2. SPRING BOOT 2.X용 FUSE 콘솔에 액세스

독립 실행형 Fuse Spring Boot 2.x 배포를 위해 Fuse Console에 액세스할 수 있습니다.

절차

1. Fuse 애플리케이션의 **pom.xml** 파일에 다음 종속성을 추가합니다.

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

Maven BOM에서 제공하므로 정확한 버전을 지정할 필요가 없습니다.

2. **src/main/resources/application.properties** 파일을 편집합니다.

a. 다음 속성을 설정합니다.

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

- b. 필요한 경우 `management.endpoints.web.base-path` 속성을 설정합니다. 기본적으로 Spring Boot 2.x의 URL에는 관리 끝점의 컨텍스트 경로(/actuator)가 포함됩니다. 예를 들면 다음과 같습니다.

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

이 기본 URL을 변경하려면 예를 들어 기본 Spring Boot 1.x URL(<http://localhost:10001/hawtio>)과 동일한 형식을 지정하려면 다음과 같이 `management.endpoints.web.base-path` 속성을 설정합니다.

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

`application.properties` 설정은 다음 예와 유사해야 합니다.

```
# 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=/
```



참고

기본적으로 Spring Boot에서 Fuse Console에 대한 인증은 비활성화되어 있습니다. 선택적으로 Fuse Console 배포와 관련된 코드를 작성하여 인증을 활성화할 수 있습니다. 지침에 사용할 수 있는 예는 다음과 같습니다.

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

3. Fuse 애플리케이션을 실행합니다.

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

4. Fuse Console URL의 포트 번호를 확인하려면 `src/main/resources/application.properties` 파일에 설정된 값을 확인하여 `management.server.port` 값을 가져옵니다. 예를 들면 다음과 같습니다.

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

5. 브라우저에서 Fuse 콘솔을 열려면 `nnnnn` 이 `management.server.port` 속성의 값인 다음 URL 구문을 사용합니다.

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

예를 들어 `management.server.port` 속성 값이 `10001` 이고 `management.endpoints.web.base-path` 속성을 설정하지 않은 경우 URL은 다음과 같습니다.

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

2.3. SPRING BOOT 1.X용 FUSE 콘솔에 액세스

독립 실행형 Fuse Spring Boot 1.x 배포를 위해 Fuse Console에 액세스할 수 있습니다.

절차

1. Fuse 애플리케이션의 `pom.xml` 파일에 다음 종속성을 추가합니다.

```
<dependency>
  <groupId>io.hawt</groupId>
  <artifactId>hawtio-springboot-1</artifactId>
</dependency>
```

Maven BOM에서 제공하므로 정확한 버전을 지정할 필요가 없습니다.

2. `src/main/resources/application.properties` 파일을 편집합니다.

- a. 다음 속성을 `false` 로 설정합니다.

- `endpoints.jolokia.sensitive`
- `endpoints.hawtio.sensitive`
- `hawtio.authenticationEnabled`

- b. 다음 속성을 `true`로 설정합니다. Set the following properties to true:

- `endpoints.hawtio.enabled`
 - `endpoints.jolokia.enabled`
- `application.properties` 설정은 다음 예와 유사해야 합니다.

```
# ports
server.port=8080
management.port=10001
```

```
# enable management endpoints for healthchecks and hawtio

endpoints.enabled = false

endpoints.hawtio.enabled = true

endpoints.jolokia.enabled = true

endpoints.health.enabled = true

management.health.defaults.enabled=false

camel.health.enabled=false

camel.health.indicator.enabled=true

endpoints.jolokia.sensitive=false

endpoints.hawtio.sensitive=false

hawtio.authenticationEnabled=false
```



참고

기본적으로 Spring Boot에서 Fuse Console에 대한 인증은 비활성화되어 있습니다. 선택적으로 Fuse Console 배포와 관련된 코드를 작성하여 인증을 활성화할 수 있습니다. 지침에 사용할 수 있는 예는 다음과 같습니다.

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

3. Fuse 애플리케이션을 실행합니다.

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

4. Fuse Console URL의 포트 번호를 확인하려면 `src/main/resources/application.properties` 파일에 설정된 값을 확인하여 `management.port` 값을 가져옵니다. 예를 들면 다음과 같습니다.

```
management.port = 10001
```

5. 브라우저에서 Fuse 콘솔을 열려면 `nnnnn` 이 `management.port` 속성의 값인 다음 URL 구문을 사용합니다.

<http://localhost:nnnnn/hawtio/index.html>

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

2.4. FUSE 콘솔 브랜딩 사용자 정의

Spring Boot 독립 실행형 애플리케이션의 Fuse에 `hawtconfig.json` 파일을 추가하여 제목, 로고 및 로그인 페이지 정보와 같은 Fuse Console 브랜딩 정보를 사용자 지정할 수 있습니다.

절차

1. Spring Boot 독립 실행형 애플리케이션의 `src/main/webapp` 디렉터리에 로컬 Fuse에 `hawtconfig.json` 이라는 JSON 파일을 생성합니다.
2. 선택한 편집기에서 `src/main/webapp/hawtconfig.json` 을 열고 다음 내용을 추가합니다.

```
{
  "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. 표 A.1. "Fuse 콘솔 구현 구성 속성"에 나열된 구성 속성 값을 변경합니다.
4. 변경 사항을 저장하십시오.
5. 다음 명령을 사용하여 Spring Boot에서 Fuse를 실행합니다.

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

6. 웹 브라우저에서 다음 URL을 사용하여 Fuse 콘솔을 엽니다.
<http://localhost:10001/actuator/hawtio/index.html>



참고

웹 브라우저에서 Fuse Console을 이미 실행한 경우 브랜드가 브라우저의 로컬 스토리지에 저장됩니다. 새로운 브랜딩 설정을 사용하려면 브라우저의 로컬 스토리지를 지워야 합니다.

2.5. FUSE CONSOLE 보안

Spring Boot에서 Fuse 콘솔을 보호하려면 다음을 수행할 수 있습니다.

- 필요한 프로토콜로 HTTPS 설정
`hawtio.http.strictTransportSecurity` 속성을 사용하면 웹 브라우저가 보안 HTTPS 프로토콜을 사용하여 Fuse Console에 액세스하도록 할 수 있습니다. 이 속성은 HTTP를 사용하여 Fuse 콘솔에 액세스하려는 웹 브라우저가 HTTPS를 사용하도록 요청을 자동으로 변환해야 함을 지정합니다.

- 공개 키를 사용하여 응답 보안
`hawtio.http.publicKeyPins` 속성을 사용하여 특정 암호화 공개 키를 Fuse 콘솔과 연결하도록 웹 브라우저에 고정 인증서와 "man-in-the-middle" 공격 위험을 줄임으로써 HTTPS 프로토콜을 보호할 수 있습니다.

절차

다음 예와 같이 `hawtio.http.strictTransportSecurity` 및 `hawtio.http.publicKeyPins` 속성을 설정합니다.

```
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);
}
```

추가 리소스

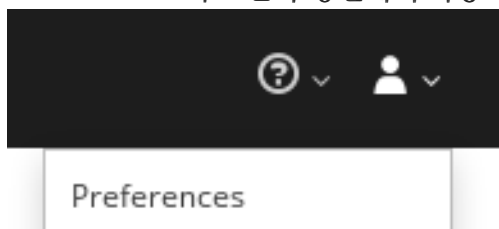
- `hawtio.http.strictTransportSecurity` 속성 구문에 대한 설명은 [HSTS\(HTTP Strict Transport Security\)](#) 응답 헤더에 대한 설명 페이지를 참조하십시오.
- Base64로 인코딩된 공개 키를 추출하는 방법에 대한 지침을 포함하여 `hawtio.http.publicKeyPins` 속성 구문에 대한 설명은 [HTTP 공개 키 고정](#) 응답 헤더에 대한 설명 페이지를 참조하십시오.

2.6. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.

Fuse Console의 대기열 및 연결 표시에 큐가 없거나 연결이 누락되었거나 일관성 없는 아이콘을 표시하는 경우 Jolokia 컬렉션 크기 매개변수를 조정하여 Jolokia marshals를 응답에 마샬링하는 배열의 최대 요소 수를 지정합니다.

절차

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 최대 컬렉션 크기 옵션의 값을 늘립니다(기본값은 50,000).
3. 닫기를 클릭합니다.

2.7. 원격 FUSE 애플리케이션에 연결

Fuse 콘솔은 클라이언트에 추가 소프트웨어(에이전트)가 설치되어 있어야 하는 JMX(Java Management Extensions)에 대한 에이전트 기반 접근 방식인 Jolokia를 사용합니다. 기본적으로 Red Hat Fuse에는 jolokia 에이전트가 포함되어 있습니다.

독립 실행형 Fuse Console 배포를 사용하면 이미 jolokia 에이전트가 있는 원격 통합에 연결할 수 있습니다 (<https://jolokia.org/>). 연결하려는 프로세스에는 jolokia 에이전트가 없는 경우 jolokia 문서 (<http://jolokia.org/agent.html>)를 참조하십시오.

절차

Fuse 콘솔의 프록시 서블릿은 허용 목록 호스트 보호를 사용하며 기본적으로 Fuse Console은 localhost에 만 연결할 수 있습니다. Fuse Console을 다른 원격 Fuse 인스턴스에 연결하려면 Spring Boot 애플리케이션의 main() 메서드에서 hawtio.proxyWhitelist 시스템 속성을 구성해야 합니다.

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

2.7.1. 원격 Jolokia 에이전트에 연결

시작하기 전에 원격 Jolokia 에이전트의 연결 세부 정보(호스트 이름, 포트 및 경로)를 알아야 합니다.

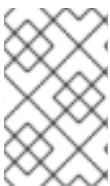
Spring Boot의 Jolokia 에이전트의 기본 연결 URL은 <http://<host>:8080/jolokia>입니다.

시스템 관리자는 이 기본값을 변경할 수 있습니다.

일반적으로 Jolokia 에이전트에 원격으로 연결하는 URL은 Fuse Console과 /jolokia 를 여는 URL입니다. 예를 들어, Fuse Console을 여는 URL이 <http://<host>:1234/hawtio> 인 경우 원격으로 연결할 URL은 <http://<host>:1234/hawtio/jolokia> 일 것입니다.

JVM을 검사할 수 있도록 원격 Jolokia 인스턴스에 연결하려면 다음을 수행합니다.

1. 연결 탭을 클릭합니다.
2. 원격 탭을 클릭한 다음 연결 추가 를 클릭합니다.
3. 이름, 스키마 (HTTP 또는 HTTPS) 및 호스트 이름을 입력합니다.
4. 연결 테스트를 클릭합니다.
5. 추가를 클릭합니다.



참고

Fuse 콘솔은 localhost 및 127.0.0.1 이외의 로컬 네트워크 인터페이스를 자동으로 조사하고 허용 목록에 추가합니다. 따라서 로컬 머신의 주소를 허용 목록에 수동으로 등록할 필요가 없습니다.

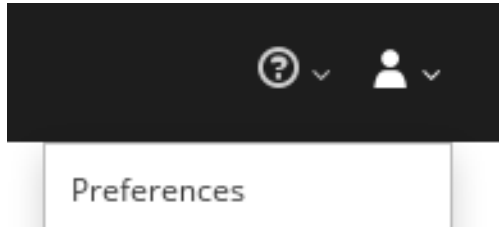
2.7.2. 데이터 이동 기본 설정

예를 들어 Fuse Console에 표시되는 데이터를 더 자주 새로 고치려는 경우 다음과 같은 Jolokia 기본 설정을 변경할 수 있습니다. 데이터 업데이트 빈도를 늘리면 네트워크 트래픽에 영향을 미치고 서버에 대한 요청 수를 늘립니다.

- 업데이트 속도 - Jolokia에 폴링하여 Cryostat 데이터를 가져오는 간격(기본값은 5초)입니다.
- 최대 깊이 - Jolokia가 반환하기 전에 서버 측의 JSON으로 객체를 마샬링하는 수준 (기본값은 7)입니다.
- 최대 컬렉션 크기 - Jolokia가 응답에서 마샬링하는 배열의 최대 요소 수입니다(기본값은 50,000).

이러한 설정의 값을 변경하려면 다음을 수행합니다.

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 옵션을 편집한 다음 단기를 클릭합니다.

2.7.3. JVM 런타임 정보 보기

시스템 속성, 메트릭 및 스레드와 같은 JVM 런타임 정보를 보려면 런타임 탭을 클릭합니다.

2.8. APACHE CAMEL 애플리케이션 보기 및 관리

Fuse 콘솔의 Camel 탭에서 Apache Camel 컨텍스트, 경로 및 종속성을 보고 관리합니다.

다음 세부 정보를 볼 수 있습니다.

- 실행 중인 모든 Camel 컨텍스트 목록
- Camel 버전 번호 및 런타임 정적과 같은 각 Camel 컨텍스트에 대한 자세한 정보
- 각 Camel 애플리케이션의 모든 경로 목록 및 런타임 통계
- 실행 중인 경로의 그래픽 표현과 실시간 메트릭

다음을 통해 Camel 애플리케이션과 상호 작용할 수도 있습니다.

- 컨텍스트 시작 및 일시 중단
- 모든 Camel 애플리케이션 및 해당 경로의 라이프사이클을 관리하여 재시작, 중지, 일시 중지, 재개 등을 수행할 수 있습니다.
- 실행 중인 경로의 실시간 추적 및 디버깅
- Camel 엔드포인트에 메시지 검색 및 전송

사전 요구 사항

Camel 탭은 하나 이상의 Camel 경로를 사용하는 컨테이너에 연결할 때만 사용할 수 있습니다.

2.8.1. 컨텍스트 시작, 일시 중지 또는 삭제

1. Camel 탭의 트리 보기에서 Camel Contexts 를 클릭합니다.
2. 목록에서 하나 이상의 컨텍스트 옆에 있는 확인란을 선택합니다.
3. 시작 또는 일시 중지를 클릭합니다.
4. 컨텍스트를 삭제하려면 다음을 수행합니다.
 - a. 컨텍스트를 중지합니다.

- b. 아이콘을 클릭한 다음 드롭다운 메뉴에서 삭제 를 선택합니다.



참고

컨텍스트를 삭제하면 배포된 애플리케이션에서 해당 컨텍스트를 제거합니다.

2.8.2. Camel 애플리케이션 세부 정보 보기

1. Camel 탭의 트리 뷰에서 Camel 애플리케이션을 클릭합니다.
2. 애플리케이션 특성 및 값 목록을 보려면 속성을 클릭합니다.
3. 애플리케이션 특성의 그래픽 표시를 보려면 차트 를 클릭한 다음 편집을 클릭하여 차트에서 볼 속성을 선택합니다.
4. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
5. 애플리케이션 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향으로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 Camel 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 유형 다운로드를 클릭합니다.
7. XML에서 경로 추가 또는 업데이트 또는 classpath에서 사용 가능한 모든 Camel 구성 요소를 찾는 등 Cryostat 작업을 보고 실행하려면 Operations 를 클릭합니다.

2.8.3. Camel 경로 목록 보기 및 상호 작용

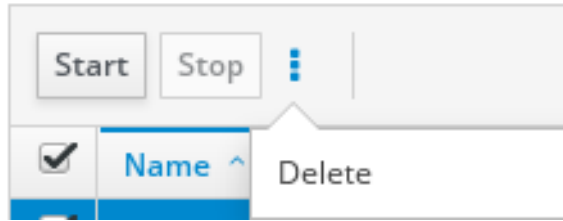
1. 경로 목록을 보려면 다음을 수행합니다.
 - a. Camel 탭을 클릭합니다.
 - b. 트리 뷰에서 애플리케이션의 경로 폴더를 클릭합니다.

Routes

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

2. 하나 이상의 경로를 시작, 중지 또는 삭제하려면 다음을 수행합니다.
 - a. 목록에서 하나 이상의 경로 옆에 있는 확인란을 선택합니다.
 - b. 시작 또는 중지 를 클릭합니다.
 - c. 경로를 삭제하려면 먼저 중지해야 합니다. 그런 다음 아이콘을 클릭하고 드롭다운 메뉴에서 삭제 를 선택합니다.

Routes



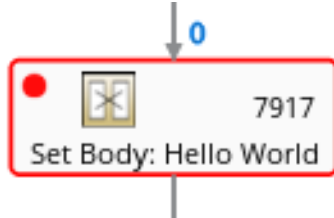
참고

- 경로를 삭제하면 배포된 애플리케이션에서 해당 경로를 제거합니다.
- 트리 보기에서 특정 경로를 선택한 다음 오른쪽 상단 메뉴를 클릭하여 시작, 중지 또는 삭제할 수도 있습니다.

3. 경로의 그래픽 다이어그램을 보려면 **Route Diagram** 을 클릭합니다.
4. 진행 중 및 차단된 교환을 보려면 **교환** 을 클릭합니다.
5. 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향별로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 **Camel** 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 **Type Cryostat** 를 클릭합니다.
7. 특정 경로와 상호 작용하려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 보기에서 경로를 선택합니다.
 - b. 경로 속성 및 값 목록을 보려면 속성을 클릭합니다.
 - c. 경로 속성의 그래픽 표시를 보려면 차트를 클릭합니다. 편집을 클릭하여 차트에서 표시할 속성을 선택할 수 있습니다.
 - d. 진행 중 및 차단된 교환을 보려면 **교환** 을 클릭합니다.
 - e. 작업에서 경로를 XML로 덤프하거나 경로의 **Camel ID** 값을 가져오는 등 경로에서 **Cryostat** 작업을 보고 실행합니다.
8. 경로를 통해 메시지를 추적하려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 보기에서 경로를 선택합니다.
 - b. 추적 을 선택한 다음 추적 시작을 클릭합니다.
9. 경로에 메시지를 보내려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 뷰에서 컨텍스트의 엔드포인트 폴더를 연 다음 엔드포인트를 선택합니다.
 - b. **Send** 를 클릭합니다.
 - c. 메시지를 **JSON** 또는 **XML** 형식으로 구성합니다.
 - d. 전송을 클릭합니다.
 - e. 경로의 추적 탭으로 돌아가 경로를 통한 메시지 흐름을 확인합니다.

2.8.4. 경로 디버깅

1. Camel 탭의 트리 보기에서 경로를 선택합니다.
2. 디버그를 선택한 다음 디버깅 시작을 클릭합니다.
3. Cryostat를 추가하려면 다이어그램에서 노드를 선택한 다음 Add Cryostat를 클릭합니다. 노드에 빨간색 점이 표시됩니다.



노드가 Cryostat 목록에 추가됩니다.

Breakpoints	
setBody1	×
log1	×

4. 아래쪽 화살표를 클릭하여 다음 노드 또는 재생 버튼으로 이동하여 경로 실행을 다시 시작합니다.
5. 일시 중지 버튼을 클릭하여 경로에 대한 모든 스레드를 일시 중지합니다.
6. 완료되면 디버깅 중지를 클릭합니다. 모든 Cryostat가 지워집니다.

2.9. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리

JMX(Java Management Extensions)는 런타임 시 리소스(서비스, 장치 및 애플리케이션)를 동적으로 관리할 수 있는 Java 기술입니다. 리소스는 Cryostats라는 오브젝트로 표시됩니다(관리됨의 경우). 리소스를 생성, 구현 또는 설치하는 즉시 리소스를 관리하고 모니터링할 수 있습니다.

Fuse Console에서 Cryostat 플러그인을 사용하면 Cryostat 도메인 및 Cryostat를 보고 관리할 수 있습니다. Cryostat 특성을 보고, 명령을 실행하고, 통계를 표시하는 차트를 만들 수 있습니다.

Cryostat 탭에서는 폴더에 구성된 활성 Cryostat 도메인 및 Cryostat의 트리 뷰를 제공합니다. Cryostat에서 세부 정보를 보고 명령을 실행할 수 있습니다.

절차

1. Cryostat 특성을 보고 편집하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. 특성 탭을 클릭합니다.
 - c. 속성을 클릭하여 세부 정보를 확인합니다.
2. 작업을 수행하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. Operations 탭을 클릭하고 나열된 작업 중 하나를 확장합니다.

- c. 실행을 클릭하여 작업을 실행합니다.
3. 차트를 보려면 다음을 수행합니다.
 - a. 트리 뷰에서 항목을 선택합니다.
 - b. 차트 탭을 클릭합니다.

2.10. 진단 보기

JVM Diagnostic Command 및 CryostatDiagnostic 인터페이스를 통해 JVM에 대한 진단 정보를 보려면 Cryostat 탭을 사용합니다.



참고

기능은 jmc(Java Mission Control) 또는 명령줄 도구 jcmd의 진단 명령 보기와 유사합니다. 플러그인은 일부 시나리오에서 해당 jcmd 명령을 제공합니다.

절차

1. 로드된 클래스의 인스턴스 수와 필요한 바이트 수를 검색하려면 클래스 histogram을 클릭합니다. 작업이 반복되면 탭에 마지막 실행 이후의 차이점이 표시됩니다.
2. JVM 진단 플래그 설정을 보려면 JVM 플래그를 클릭합니다.
3. 실행 중인 JVM의 경우 플래그 설정도 수정할 수 있습니다.

추가 리소스

지원되는 JVM은 플랫폼에 따라 다릅니다. 자세한 내용은 다음 소스 중 하나로 이동합니다.

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

2.11. 스레드 보기

스레드 상태를 보고 모니터링할 수 있습니다.

절차

1. Runtime 탭을 클릭한 다음 Threads 를 클릭합니다. Threads 페이지에는 각 스레드에 대한 활성 스레드 및 스택 추적 세부 정보가 나열됩니다. 기본적으로 스레드 목록은 모든 스레드를 내림차순으로 표시합니다.
2. ID를 눌러 목록을 정렬하려면 ID 열 레이블을 클릭합니다.
3. 선택적으로 스레드 상태(예: Blocked) 또는 스레드 이름으로 목록을 필터링합니다.
4. 해당 스레드의 잠금 클래스 이름 및 전체 스택 추적과 같은 특정 스레드에 대한 자세한 정보를 드릴 다운하려면 Actions 열에서 More 를 클릭합니다.

3장. KARAF 독립 실행형에서 RED HAT FUSE 애플리케이션 모니터링 및 관리

3.1. FUSE 콘솔 정보

Red Hat Fuse Console은 Hawtio 오픈 소스 소프트웨어를 기반으로 하는 웹 콘솔입니다. 지원되는 브라우저 목록은 [지원되는 구성으로 이동합니다](#).

Fuse Console은 배포된 하나 이상의 Fuse 컨테이너의 세부 정보를 검사하고 관리하는 중앙 인터페이스를 제공합니다. 또한 Red Hat Fuse 및 시스템 리소스를 모니터링하고 업데이트를 수행하며 서비스를 시작하거나 중지할 수 있습니다.

Red Hat Fuse 독립 실행형을 설치하거나 OpenShift에 Fuse를 사용하는 경우 Fuse Console을 사용할 수 있습니다. Fuse 콘솔에서 보고 관리할 수 있는 통합은 실행 중인 플러그인에 따라 다릅니다. 가능한 플러그인은 다음과 같습니다.

- Camel
- JMX
- Spring Boot
- OSGI
- 런타임
- 로그

3.2. FUSE CONSOLE에 액세스

Apache Karaf 독립 실행형용 Fuse 콘솔에 액세스하려면 다음 단계를 따르십시오.

사전 요구 사항

Karaf 컨테이너에 Fuse를 설치합니다. 단계별 지침은 [Apache Karaf에 설치](#)를 참조하십시오.

절차

1. 명령줄에서 Red Hat Fuse를 설치한 디렉터리로 이동하여 다음 명령을 실행하여 Fuse 독립 실행형을 시작합니다.

```
./bin/fuse
```

Karaf 콘솔은 버전 정보, 기본 Fuse Console URL 및 공통 명령 목록을 시작하고 표시합니다.

2. 브라우저에서 Fuse Console에 연결할 URL을 입력합니다. 예: <http://localhost:8181/hawtio>
3. 로그인 페이지에서 사용자 이름과 암호를 입력한 다음 로그인 을 클릭합니다.

기본적으로 Fuse Console에는 홈 페이지가 표시됩니다. 왼쪽 탐색 탭에는 실행 중인 플러그인이 표시됩니다.

3.3. FUSE CONSOLE 보안

Apache Karaf에서 Fuse 콘솔을 보호하려면 다음을 수행할 수 있습니다.

- **필요한 프로토콜로 HTTPS 설정**
hawtio.http.strictTransportSecurity 속성을 사용하면 웹 브라우저가 보안 HTTPS 프로토콜을 사용하여 Fuse Console에 액세스하도록 할 수 있습니다. 이 속성은 HTTP를 사용하여 Fuse 콘솔에 액세스하려는 웹 브라우저가 HTTPS를 사용하도록 요청을 자동으로 변환해야 함을 지정합니다.
- **공개 키를 사용하여 응답 보안**
hawtio.http.publicKeyPins 속성을 사용하여 특정 암호화 공개 키를 Fuse 콘솔과 연결하도록 웹 브라우저에 고정 인증서와 "man-in-the-middle" 공격 위험을 줄임으로써 HTTPS 프로토콜을 보호할 수 있습니다.
- **SSL/TLS 보안 활성화**
 Fuse Console에 대해 SSL/TLS 보안은 기본적으로 활성화되어 있지 않습니다. Fuse 콘솔에서 SSL/TLS 보안을 활성화하여 사용자 이름/암호 자격 증명을 스누핑에서 보호하는 것이 좋습니다.
- **Red Hat Single Sign On 구현**
- **사용자 액세스 제어**
 인증된 사용자가 수행할 수 있는 작업은 표 3.1. "Karaf 독립 실행형 역할 기반 액세스"에 나열된 대로 해당 사용자에게 할당된 역할(또는 역할)에 따라 달라집니다.

절차

1. 필요한 프로토콜로 HTTPS를 설정하려면 다음 예와 같이 **\$KARAF_HOME/etc/system.properties** 파일에서 **hawtio.http.strictTransportSecurity** 속성을 설정합니다.


```
hawtio.http.strictTransportSecurity = max-age=31536000; includeSubDomains; preload
```
2. 공개 키를 사용하여 응답을 보호하려면 다음 예와 같이 **\$KARAF_HOME/etc/system.properties** 파일에서 **hawtio.http.publicKeyPins** 속성을 설정합니다.


```
hawtio.http.publicKeyPins = pin-sha256="cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs"; max-age=5184000; includeSubDomains
```
3. SSL/TLS 보안을 활성화하는 방법에 대한 자세한 내용은 Apache Karaf 보안 가이드의 "[Apache Karaf 컨테이너의 SSL/TLS 활성화](#)" 섹션을 참조하십시오.
4. Red Hat Single Sign-On을 사용하여 Fuse 콘솔을 보호하는 방법에 대한 자세한 내용은 [Red Hat Single Sign-on Securing Applications and Services Guide](#) 에서 Hawtio 관리 콘솔 보안 섹션을 참조하십시오.
5. 사용자가 사용자가 수행해야 하는 Fuse Console 작업을 수행하는 데 필요한 사용자 역할 권한이 있는지 확인하려면 다음 단계를 수행하여 사용자 역할을 설정합니다.
 - a. 편집기에서 Red Hat Fuse **etc/users.properties** 파일을 엽니다.
 - b. 사용자 이름, 암호 및 역할에 대한 항목을 추가합니다.
 예를 들어 **etc/users.properties** 파일의 다음 항목은 admin 사용자를 정의하고 admin 역할을 부여합니다.

```
admin = secretpass,admin
```

c. 파일을 저장합니다.

추가 리소스

- `hawtio.http.strictTransportSecurity` 속성 구문에 대한 설명은 [HSTS\(HTTP Strict Transport Security\)](#) 응답 헤더에 대한 설명 페이지를 참조하십시오.
- Base64로 인코딩된 공개 키를 추출하는 방법에 대한 지침을 포함하여 `hawtio.http.publicKeyPins` 속성 구문에 대한 설명은 [HTTP 공개 키 고정](#) 응답 헤더에 대한 설명 페이지를 참조하십시오.

3.4. 역할 기반 액세스 참조

인증된 사용자가 수행할 수 있는 작업은 표 3.1. "Karaf 독립 실행형 역할 기반 액세스"에 나열된 대로 해당 사용자에게 할당된 역할(또는 역할)에 따라 달라집니다.

표 3.1. Karaf 독립 실행형 역할 기반 액세스

작업	admin	관리자	뷰어
로그인/로그인	Y	Y	Y
도움말 주제 보기	Y	Y	Y
사용자 기본 설정 설정	Y	Y	Y
연결			
원격 통합 검색 및 연결	Y	Y	Y
로컬 통합 검색 및 연결	Y	Y	Y
Camel			
실행 중인 모든 Camel 애플리케이션 보기	Y	Y	Y
Camel 컨텍스트 시작, 일시 중지, 재개 및 삭제	Y	Y	
메시지 전송	Y	Y	
끝점 추가	Y	Y	
경로, 경로 다이어그램 및 런타임 통계 보기	Y	Y	Y
경로 시작 및 중지	Y	Y	
경로 삭제	Y	Y	

작업	admin	관리자	뷰어
JMX			
속성 값 변경	Y	Y	
시간 기반 차트에서 특성 선택 및 보기	Y	Y	Y
작업 보기	Y	Y	Y
OSGI			
번들, 기능, 패키지, 서비스, 서버, 프레임워크 및 구성 보기	Y	Y	Y
번들 추가 및 삭제	Y	Y	
구성 추가	Y	Y	
기능 설치 및 제거	Y		
런타임			
시스템 속성, 메트릭 및 스태트 보기	Y	Y	Y
로그			
로그 보기	Y	Y	Y

추가 리소스

역할 기반 액세스 제어에 대한 자세한 내용은 [Apache Karaf에 배포를 참조하십시오](#).

3.5. FUSE 콘솔 브랜딩 사용자 정의

Fuse Console 브랜딩 플러그인을 사용하여 제목, 로고 및 로그인 페이지 정보와 같은 Fuse Console 브랜딩 정보를 사용자 지정할 수 있습니다.

기본적으로 Fuse 콘솔 브랜딩은 Fuse 콘솔 WAR 파일 (`karaf-install-dir/system/io/hawtio-war/<version>/hawtio-war/<version>.war`)에 있는 `hawtconfig.json`에 정의되어 있습니다. Fuse Console 브랜딩 플러그인을 구현할 때 기본 브랜딩을 자체 사용자 정의 브랜딩으로 재정의할 수 있습니다.

절차

1. <https://github.com/hawtio/hawtio/tree/hawtio-2.9.1/examples/branding-plugin>에서 선택한 로컬 디렉터리로 브랜딩 플러그인 예제를 다운로드합니다.

2. 선택한 편집기에서 Fuse Console 브랜딩 플러그인의 `src/main/webapp/plugin/brandingPlugin.js` 파일을 열어 Fuse Console 브랜딩을 사용자 지정합니다.
표 A.1. "Fuse 콘솔 구현 구성 속성"에 나열된 구성 속성 값을 변경할 수 있습니다.

3. 변경 사항을 저장하십시오.

4. 선택한 편집기에서 Fuse Console 브랜딩 플러그인의 `pom.xml` 파일을 `<parent>` 섹션으로 엽니다.

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

5. 다음과 같이 `<parent>` 섹션을 편집합니다.

- a. Karaf 설치 시 Fuse의 버전과 일치하도록 `<version>` 속성의 값을 변경합니다. 예를 들어 Fuse on Karaf 설치 디렉터리 이름이 `2.0.0.fuse-760015` 인 경우 버전을 `2.0.0.fuse-760015` 로 설정합니다.
- b. `<relativePath>...</relativePath>` 행을 제거합니다.
예를 들면 다음과 같습니다.

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

6. 터미널 창에서 다음 명령을 실행하여 `branding-plugin` 프로젝트를 빌드합니다.

```
mvn clean install
```

7. Fuse가 아직 실행되고 있지 않은 경우 다음 명령을 실행하여 시작합니다.
Linux/Unix: `bin/fuse`

Windows: `bin\fuse.bat``

8. Karaf CLI 프롬프트에서 다음 명령을 입력하여 Fuse 콘솔 브랜딩 플러그인을 설치합니다(`<version>`은 Karaf 설치 시 Fuse 버전임).

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

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

9. 웹 브라우저에서 시작 명령이 7단계에서 반환된 URL을 사용하여 Fuse 콘솔을 엽니다(기본 URL은 <http://localhost:8181/hawtio/>).



참고

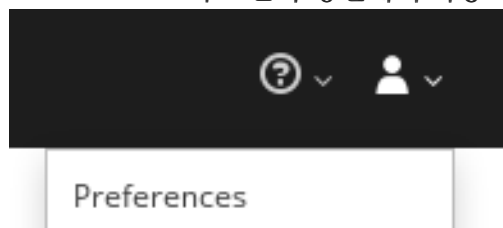
웹 브라우저에서 Fuse Console을 이미 실행한 경우 브랜드가 브라우저의 로컬 스토리지에 저장됩니다. 새로운 브랜딩 설정을 사용하려면 브라우저의 로컬 스토리지를 지워야 합니다.

3.6. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.

Fuse Console의 대기열 및 연결 표시에 큐가 없거나 연결이 누락되었거나 일관성 없는 아이콘을 표시하는 경우 Jolokia 컬렉션 크기 매개변수를 조정하여 Jolokia marshals를 응답에 마샬링하는 배열의 최대 요소 수를 지정합니다.

절차

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 최대 컬렉션 크기 옵션의 값을 늘립니다(기본값은 50,000).
3. 닫기를 클릭합니다.

3.7. FUSE 콘솔 비활성화

다른 구성 요소에 영향을 주지 않고 모든 사용자가 액세스할 수 없도록 Karaf에서 Fuse 콘솔을 비활성화할 수 있습니다.

절차

1. hawtio-web 번들 ID를 확인하려면 다음 명령을 사용하여 Fuse Console에서 사용하는 Fuse 번들을 나열합니다.
OSGi:list | grep hawtio
2. 번들을 중지하려면 `osgi:stop` 명령을 사용합니다. 예를 들어 hawtio :: 웹 콘솔번들의 ID가 246인 경우 다음 명령을 입력합니다.
osgi:stop 246

번들이 해결된 상태로 전환되고 더 이상 Fuse Console에 액세스할 수 없습니다.

추가 리소스

번들 관리에 대한 자세한 내용은 [Apache Karaf에 배포](#)의 "라이프 관리" 장을 참조하십시오.

3.8. 원격 FUSE 애플리케이션에 연결

Fuse 콘솔은 클라이언트에 추가 소프트웨어(에이전트)가 설치되어 있어야 하는 JMX(Java Management Extensions)에 대한 에이전트 기반 접근 방식인 Jolokia를 사용합니다. 기본적으로 Red Hat Fuse에는 jolokia 에이전트가 포함되어 있습니다.

독립 실행형 Fuse Console 배포를 사용하면 이미 jolokia 에이전트가 있는 원격 통합에 연결할 수 있습니다 (<https://jolokia.org/>). 연결하려는 프로세스에는 jolokia 에이전트가 없는 경우 jolokia 문서 (<http://jolokia.org/agent.html>)를 참조하십시오.

3.8.1. Fuse 콘솔 잠금 해제

기본적으로 Apache Karaf에서 Fuse 7 독립 실행형용 Jolokia는 잠겼으며 Fuse Console은 원격으로 액세스할 수 없습니다.

localhost 또는 127.0.0.1 이외의 호스트 이름 또는 IP 주소에 대해 Fuse Console의 잠금을 해제하려면 다음 단계를 따르십시오.

1. 편집기에서 `$KARAF_HOME/etc/jolokia-access.xml` 파일을 엽니다.
2. `<cors>` 섹션에 추가하여 Fuse 콘솔로 액세스할 Fuse 통합의 호스트 이름 또는 IP 주소를 등록합니다.

예를 들어 Fuse 콘솔에서 호스트 이름 0.0.0.3 에 액세스하려면 다음을 추가합니다.

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

다음과 같이 행합니다.

```
<!--
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. 파일을 저장합니다.

3.8.2. 원격 액세스 제한

선택적으로 특정 호스트 및 IP 주소에 대해 Fuse Console에 대한 원격 액세스를 제한할 수 있습니다.

HTTP 클라이언트의 IP 주소를 기반으로 전체 액세스 권한을 부여할 수 있습니다. 다음 제한 사항을 지정하려면 다음을 수행합니다.

`jolokia-access.xml` 파일에서 하나 이상의 `<host>` 요소가 포함된 `<remote>` 섹션을 추가하거나 편집합니다. `<host>` 요소의 경우 CIDR 형식으로 지정된 IP 주소, 호스트 이름 또는 넷마스크를 지정할 수 있습니다 (예: 10.0 네트워크에서 들어오는 모든 클라이언트의 경우 `10.0.0.0/16`).

다음 예제에서는 `localhost`와 IP 주소가 `10.0` 으로 시작하는 모든 클라이언트에서 액세스할 수 있습니다. 다른 모든 IP 주소의 경우 액세스가 거부됩니다.

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

자세한 내용은 Jolokia 보안 문서 (<https://jolokia.org/reference/html/security.html>)를 참조하십시오.

3.8.3. 원격 Fuse 인스턴스에 연결 허용

Fuse 콘솔의 프록시 서블릿은 허용 목록 호스트 보호를 사용하며 기본적으로 Fuse Console은 `localhost`에 만 연결할 수 있습니다. Fuse Console을 다른 원격 Fuse 인스턴스에 연결하려면 다음과 같이 허용 목록을 구성해야 합니다.

Apache Karaf의 경우 `etc/system.properties` 파일에서 다음과 같은 구성을 변경합니다.

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

3.8.4. 원격 Jolokia 에이전트에 연결

시작하기 전에 원격 Jolokia 에이전트의 연결 세부 정보(호스트 이름, 포트 및 경로)를 알아야 합니다.

Apache Karaf의 Fuse용 Jolokia 에이전트의 기본 연결 URL은 `http://<host>:8181/hawtio/jolokia`입니다.

시스템 관리자는 이 기본값을 변경할 수 있습니다.

일반적으로 Jolokia 에이전트에 원격으로 연결하는 URL은 Fuse Console과 `/jolokia` 를 여는 URL입니다. 예를 들어, Fuse Console을 여는 URL이 `http://<host>:1234/hawtio` 인 경우 원격으로 연결할 URL은 `http://<host>:1234/hawtio/jolokia` 일 것입니다.

JVM을 검사할 수 있도록 원격 Jolokia 인스턴스에 연결하려면 다음을 수행합니다.

1. 연결 탭을 클릭합니다.
2. 원격 탭을 클릭한 다음 연결 추가 를 클릭합니다.
3. 이름, 스키마 (HTTP 또는 HTTPS) 및 호스트 이름을 입력합니다.
4. 연결 테스트를 클릭합니다.
5. 추가를 클릭합니다.



참고

Fuse 콘솔은 localhost 및 127.0.0.1 이외의 로컬 네트워크 인터페이스를 자동으로 조사하고 허용 목록에 추가합니다. 따라서 로컬 머신의 주소를 허용 목록에 수동으로 등록할 필요가 없습니다.

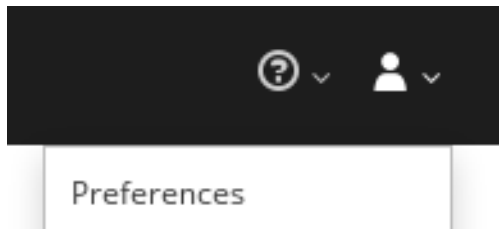
3.8.5. 데이터 이동 기본 설정

예를 들어 Fuse Console에 표시되는 데이터를 더 자주 새로 고치려는 경우 다음과 같은 Jolokia 기본 설정을 변경할 수 있습니다. 데이터 업데이트 빈도를 늘리면 네트워크 트래픽에 영향을 미치고 서버에 대한 요청 수를 늘립니다.

- 업데이트 속도 - Jolokia에 폴링하여 CRYostat 데이터를 가져오는 간격(기본값은 5초)입니다.
- 최대 깊이 - Jolokia가 반환하기 전에 서버 측의 JSON으로 객체를 마샬링하는 수준(기본값은 7)입니다.
- 최대 컬렉션 크기 - Jolokia가 응답에서 마샬링하는 배열의 최대 요소 수입니다(기본값은 50,000).

이러한 설정의 값을 변경하려면 다음을 수행합니다.

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 옵션을 편집한 다음 단기를 클릭합니다.

3.8.6. JVM 런타임 정보 보기

시스템 속성, 메트릭 및 스레드와 같은 JVM 런타임 정보를 보려면 런타임 탭을 클릭합니다.

3.9. APACHE CAMEL 애플리케이션 보기 및 관리

Fuse 콘솔의 Camel 탭에서 Apache Camel 컨텍스트, 경로 및 종속성을 보고 관리합니다.

다음 세부 정보를 볼 수 있습니다.

- 실행 중인 모든 Camel 컨텍스트 목록
- Camel 버전 번호 및 런타임 정적과 같은 각 Camel 컨텍스트에 대한 자세한 정보
- 각 Camel 애플리케이션의 모든 경로 목록 및 런타임 통계
- 실행 중인 경로의 그래픽 표현과 실시간 메트릭

다음을 통해 Camel 애플리케이션과 상호 작용할 수도 있습니다.

- 컨텍스트 시작 및 일시 중단

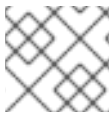
- 모든 Camel 애플리케이션 및 해당 경로의 라이프사이클을 관리하여 재시작, 중지, 일시 중지, 재개 등을 수행할 수 있습니다.
- 실행 중인 경로의 실시간 추적 및 디버깅
- Camel 엔드포인트에 메시지 검색 및 전송

사전 요구 사항

Camel 탭은 하나 이상의 Camel 경로를 사용하는 컨테이너에 연결할 때만 사용할 수 있습니다.

3.9.1. 컨텍스트 시작, 일시 중지 또는 삭제

1. Camel 탭의 트리 보기에서 Camel Contexts 를 클릭합니다.
2. 목록에서 하나 이상의 컨텍스트 옆에 있는 확인란을 선택합니다.
3. 시작 또는 일시 중지를 클릭합니다.
4. 컨텍스트를 삭제하려면 다음을 수행합니다.
 - a. 컨텍스트를 중지합니다.
 - b. 아이콘을 클릭한 다음 드롭다운 메뉴에서 삭제를 선택합니다.



참고

컨텍스트를 삭제하면 배포된 애플리케이션에서 해당 컨텍스트를 제거합니다.

3.9.2. Camel 애플리케이션 세부 정보 보기

1. Camel 탭의 트리 뷰에서 Camel 애플리케이션을 클릭합니다.
2. 애플리케이션 특성 및 값 목록을 보려면 속성을 클릭합니다.
3. 애플리케이션 특성의 그래픽 표시를 보려면 차트를 클릭한 다음 편집을 클릭하여 차트에서 볼 속성을 선택합니다.
4. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
5. 애플리케이션 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향으로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 Camel 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 유형 다운로드를 클릭합니다.
7. XML에서 경로 추가 또는 업데이트 또는 classpath에서 사용 가능한 모든 Camel 구성 요소를 찾는 등 Cryostat 작업을 보고 실행하려면 Operations 를 클릭합니다.

3.9.3. Camel 경로 목록 보기 및 상호 작용

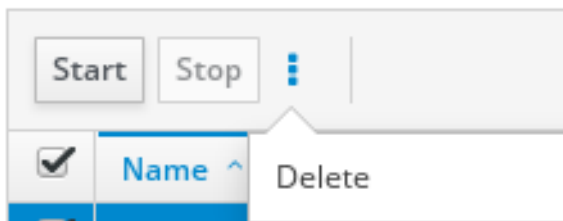
1. 경로 목록을 보려면 다음을 수행합니다.
 - a. Camel 탭을 클릭합니다.
 - b. 트리 뷰에서 애플리케이션의 경로 폴더를 클릭합니다.

Routes

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

2. 하나 이상의 경로를 시작, 중지 또는 삭제하려면 다음을 수행합니다.
 - a. 목록에서 하나 이상의 경로 옆에 있는 확인란을 선택합니다.
 - b. 시작 또는 중지를 클릭합니다.
 - c. 경로를 삭제하려면 먼저 중지해야 합니다. 그런 다음 아이콘을 클릭하고 드롭다운 메뉴에서 삭제를 선택합니다.

Routes



참고

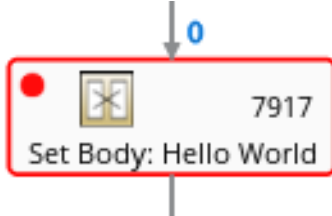
- 경로를 삭제하면 배포된 애플리케이션에서 해당 경로를 제거합니다.
- 트리 보기에서 특정 경로를 선택한 다음 오른쪽 상단 메뉴를 클릭하여 시작, 중지 또는 삭제할 수도 있습니다.

3. 경로의 그래픽 다이어그램을 보려면 Route Diagram 을 클릭합니다.
4. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
5. 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향별로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 Camel 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 Type Crystallizer를 클릭합니다.
7. 특정 경로와 상호 작용하려면 다음을 수행합니다.
 - a. Camel 탭의 트리 보기에서 경로를 선택합니다.
 - b. 경로 속성 및 값 목록을 보려면 속성을 클릭합니다.
 - c. 경로 속성의 그래픽 표시를 보려면 차트를 클릭합니다. 편집을 클릭하여 차트에서 표시할 속성을 선택할 수 있습니다.
 - d. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.

- e. 작업에서 경로를 XML로 덤프하거나 경로의 Camel ID 값을 가져오는 등 경로에서 Cryostat 작업을 보고 실행합니다.
8. 경로를 통해 메시지를 추적하려면 다음을 수행합니다.
 - a. Camel 탭의 트리 보기에서 경로를 선택합니다.
 - b. 추적을 선택한 다음 추적 시작을 클릭합니다.
 9. 경로에 메시지를 보내려면 다음을 수행합니다.
 - a. Camel 탭의 트리 뷰에서 컨텍스트의 엔드포인트 폴더를 연 다음 엔드포인트를 선택합니다.
 - b. Send 를 클릭합니다.
 - c. 메시지를 JSON 또는 XML 형식으로 구성합니다.
 - d. 전송을 클릭합니다.
 - e. 경로의 추적 탭으로 돌아가 경로를 통한 메시지 흐름을 확인합니다.

3.9.4. 경로 디버깅

1. Camel 탭의 트리 보기에서 경로를 선택합니다.
2. 디버깅을 선택한 다음 디버깅 시작을 클릭합니다.
3. Cryostat를 추가하려면 다이어그램에서 노드를 선택한 다음 Add Cryostat를 클릭합니다. 노드에 빨간색 점이 표시됩니다.



노드가 Cryostat 목록에 추가됩니다.

Breakpoints	
setBody1	×
log1	×

4. 아래쪽 화살표를 클릭하여 다음 노드 또는 재생 버튼으로 이동하여 경로 실행을 다시 시작합니다.
5. 일시 중지 버튼을 클릭하여 경로에 대한 모든 스레드를 일시 중지합니다.
6. 완료되면 디버깅 중지를 클릭합니다. 모든 Cryostat가 지워집니다.

3.10. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리

JMX(Java Management Extensions)는 런타임 시 리소스(서비스, 장치 및 애플리케이션)를 동적으로 관리할 수 있는 Java 기술입니다. 리소스는 Cryostats라는 오브젝트로 표시됩니다(관리됨의 경우). 리소스를 생성, 구형 또는 설치하는 즉시 리소스를 관리하고 모니터링할 수 있습니다.

Fuse Console에서 Cryostat 플러그인을 사용하면 Cryostat 도메인 및 Cryostat를 보고 관리할 수 있습니다. Cryostat 특성을 보고, 명령을 실행하고, 통계를 표시하는 차트를 만들 수 있습니다.

Cryo stat 탭에서는 폴더에 구성된 활성 Cryostat 도메인 및 Cryostat의 트리 뷰를 제공합니다. Cryostat에서 세부 정보를 보고 명령을 실행할 수 있습니다.

절차

1. Cryostat 특성을 보고 편집하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. 특성 탭을 클릭합니다.
 - c. 속성을 클릭하여 세부 정보를 확인합니다.
2. 작업을 수행하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. Operations 탭을 클릭하고 나열된 작업 중 하나를 확장합니다.
 - c. 실행을 클릭하여 작업을 실행합니다.
3. 차트를 보려면 다음을 수행합니다.
 - a. 트리 뷰에서 항목을 선택합니다.
 - b. 차트 탭을 클릭합니다.

3.11. OSGI 환경 보기 및 관리

Apache Karaf 독립 실행형 배포의 경우 Red Hat Fuse OSGi 환경을 보고 관리할 수 있습니다. 컨테이너 번들, 기능 및 구성뿐만 아니라 Java 패키지 및 OSGi 서비스를 보고 관리할 수 있습니다.

OSGi 탭에는 각 컨테이너 구성 요소에 대한 옵션이 포함된 일련의 기능이 포함되어 있습니다.

번들

설치된 번들 목록입니다. 번들을 설치 및 제거하고 번들을 시작 및 중지하고 번들 속성을 편집할 수 있습니다. 목록 및 그리드 보기 간의 토글을 필터링할 수도 있습니다.

기능

사용 가능한 기능 목록입니다. 기능 또는 기능 리포지토리를 설치 및 제거하고 드릴다운하여 기능 세부 정보를 볼 수 있습니다.

패키지

설치된 Java 패키지 목록입니다. 패키지 버전 및 관련 번들을 볼 수 있습니다.

서비스

실행 중인 서비스 목록입니다. 서비스 ID, 관련 번들 및 오브젝트 클래스를 볼 수 있습니다.

선언적 서비스

선언적 OSGi 서비스 목록입니다. 서비스 상태를 보고 드릴다운하여 서비스 세부 정보를 볼 수 있습니다. 또한 서비스를 활성화 및 비활성화할 수 있습니다.

서버

로컬 또는 원격 호스트에 대한 자세한 정보는 읽기 전용 모드입니다.

프레임워크

컨테이너 OSGi 프레임워크의 구성 옵션입니다. 프레임워크 시작 수준 및 초기 번들 시작 수준을 설정할 수 있습니다.

설정

구성 오브젝트 목록입니다. 각 개체의 상태를 보고 드릴다운하여 오브젝트 세부 정보를 보거나 편집할 수 있습니다. 새 구성 오브젝트를 생성할 수도 있습니다.

3.12. 진단 보기

JVM Diagnostic Command 및 CryostatDiagnostic 인터페이스를 통해 JVM에 대한 진단 정보를 보려면 Cryostat 탭을 사용합니다.



참고

기능은 jmc(Java Mission Control) 또는 명령줄 도구 jcmd의 진단 명령 보기와 유사합니다. 플러그인은 일부 시나리오에서 해당 jcmd 명령을 제공합니다.

절차

1. 로드된 클래스의 인스턴스 수와 필요한 바이트 수를 검색하려면 클래스 histogram을 클릭합니다. 작업이 반복되면 탭에 마지막 실행 이후의 차이점이 표시됩니다.
2. JVM 진단 플래그 설정을 보려면 JVM 플래그를 클릭합니다.
3. 실행 중인 JVM의 경우 플래그 설정도 수정할 수 있습니다.

추가 리소스

지원되는 JVM은 플랫폼에 따라 다릅니다. 자세한 내용은 다음 소스 중 하나로 이동합니다.

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

3.13. 스레드 보기

스레드 상태를 보고 모니터링할 수 있습니다.

절차

1. Runtime 탭을 클릭한 다음 Threads 를 클릭합니다. Threads 페이지에는 각 스레드에 대한 활성 스레드 및 스택 추적 세부 정보가 나열됩니다. 기본적으로 스레드 목록은 모든 스레드를 내림차순으로 표시합니다.
2. ID를 눌러 목록을 정렬하려면 ID 열 레이블을 클릭합니다.
3. 선택적으로 스레드 상태(예: Blocked) 또는 스레드 이름으로 목록을 필터링합니다.
4. 해당 스레드의 잠금 클래스 이름 및 전체 스택 추적과 같은 특정 스레드에 대한 자세한 정보를 드릴다운하려면 Actions 열에서 More 를 클릭합니다.

3.14. 로그 항목 보기

로그 탭에서 Red Hat Fuse의 로그 항목을 볼 수 있습니다.

사전 요구 사항

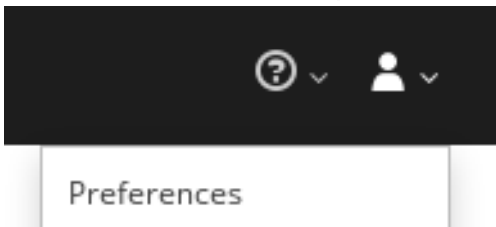
로그 탭은 Java 애플리케이션에 로그가 포함된 경우 사용할 수 있습니다.

절차

1. 로그 항목 목록을 보려면 로그 항목 탭을 클릭합니다.
기본적으로 목록에는 로그 항목이 오름차순으로 표시됩니다.

각 로그 항목으로 드릴다운하여 로그 항목에 대한 자세한 정보를 볼 수 있습니다.
2. 특정 로그 유형을 표시하도록 로그 목록을 필터링하려면 작업 표시줄을 클릭합니다. 텍스트 문자열 또는 로깅 수준에 따라 로그 항목 섹션을 필터링할 수 있습니다.
3. Fuse Console 기본 설정을 변경하려면 다음을 수행합니다.

- a. Fuse Console의 오른쪽 상단에 있는 사용자 아이콘을 클릭한 다음 드롭다운 메뉴에서 Preferences를 클릭합니다.



- b. 기본 정렬 순서를 변경하려면 서버 로그를 선택한 다음 로그 항목 링크를 클릭하여 번들 이름, 스투드 및 전체 메시지 텍스트와 같은 로그 항목에 대한 세부 정보를 드릴다운합니다.
- c. 선택적으로 로그 메시지를 저장하기 위해 이러한 설정을 사용자 지정할 수 있습니다.
 - Fuse Console에 보관할 로그 문의 수입니다(기본값: 100).
 - 글로벌 로그 수준: INFO (기본값), OFF, ERROR, WARN, DEBUG.
 - hawtio-oauth 및 hawtio-core-utils 와 같이 포함할 하위 수준 메시지입니다.
- d. Fuse 콘솔 로그 설정을 기본값으로 재설정하려면 재설정 → 설정 재설정을 클릭합니다.

4장. EAP 독립 실행형에서 RED HAT FUSE 애플리케이션 모니터링 및 관리

4.1. FUSE 콘솔 정보

Red Hat Fuse Console은 Hawtio 오픈 소스 소프트웨어를 기반으로 하는 웹 콘솔입니다. 지원되는 브라우저 목록은 [지원되는 구성으로 이동합니다](#).

Fuse Console은 배포된 하나 이상의 Fuse 컨테이너의 세부 정보를 검사하고 관리하는 중앙 인터페이스를 제공합니다. 또한 Red Hat Fuse 및 시스템 리소스를 모니터링하고 업데이트를 수행하며 서비스를 시작하거나 중지할 수 있습니다.

Red Hat Fuse 독립 실행형을 설치하거나 OpenShift에 Fuse를 사용하는 경우 Fuse Console을 사용할 수 있습니다. Fuse 콘솔에서 보고 관리할 수 있는 통합은 실행 중인 플러그인에 따라 다릅니다. 가능한 플러그인은 다음과 같습니다.

- Camel
- JMX
- Spring Boot
- OSGI
- 런타임
- 로그

4.2. FUSE CONSOLE에 액세스

다음 단계에 따라 Red Hat JBoss Enterprise Application Platform용 Fuse Console에 액세스하십시오.

사전 요구 사항

JBoss EAP 컨테이너에 Fuse를 설치해야 합니다. 단계별 지침은 [JBoss EAP에 설치](#)를 참조하십시오.

절차

독립 실행형 JBoss EAP 배포를 위해 Fuse 콘솔에 액세스하려면 다음을 수행합니다.

1. 다음 명령을 사용하여 Red Hat Fuse 독립 실행형을 시작합니다.
On Linux/Mac OS: `./bin/standalone.sh`

Windows의 경우: `./bin/standalone.skip`
2. 웹 브라우저에서 Fuse 콘솔에 연결할 URL을 입력합니다. 예: <http://localhost:8080/hawtio>
3. 로그인 페이지에서 사용자 이름과 암호를 입력한 다음 로그인 을 클릭합니다.

기본적으로 Fuse Console에는 홈 페이지가 표시됩니다. 왼쪽 탐색 탭에는 실행 중인 플러그인이 표시됩니다.



참고

기본 Fuse Console 페이지가 브라우저에 표시되는 데 시간이 오래 걸리는 경우 로그 파일의 수와 크기를 줄여야 할 수 있습니다. `periodic-size-rotating-file-handler` 를 사용하여 파일을 최대 크기(`rotate-size`)에 도달하고 여러 파일(`max-backup-index`)을 유지 관리할 수 있습니다. 이 처리기 사용 방법에 대한 자세한 내용은 Red Hat JBoss Enterprise Application Platform 제품 설명서를 참조하십시오.

4.3. FUSE 콘솔 브랜딩 사용자 정의

Fuse Console 브랜딩 플러그인을 사용하여 제목, 로고 및 로그인 페이지 정보와 같은 Fuse Console 브랜딩 정보를 사용자 지정할 수 있습니다.

기본적으로 Fuse 콘솔 브랜딩은 Fuse 콘솔 WAR 파일에 있는 `hawtconfig.json` 에 정의되어 있습니다(`Cryostat-install-dir/standalone/deployments/hawtio-wildfly-<version>.war`). Fuse Console 브랜딩 플러그인을 구현할 때 기본 브랜딩을 자체 사용자 정의 브랜딩으로 재정의할 수 있습니다.

절차

1. <https://github.com/hawtio/hawtio/tree/master/examples/branding-plugin> 에서 선택한 로컬 디렉터리로 브랜딩 플러그인 예제를 다운로드합니다.
2. 선택한 편집기에서 Fuse Console 브랜딩 플러그인의 `src/main/webapp/plugin/brandingPlugin.js` 파일을 열어 Fuse Console 브랜딩을 사용자 지정합니다.
[표 A.1. "Fuse 콘솔 구현 구성 속성"](#)에 나열된 구성 속성 값을 변경할 수 있습니다.
3. 변경 사항을 저장하십시오.
4. 선택한 편집기에서 Fuse Console 브랜딩 플러그인의 `pom.xml` 파일을 `<parent>` 섹션으로 엽니다.

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

5. 다음과 같이 `<parent>` 섹션을 편집합니다.
 - a. EAP 설치 시 Fuse의 버전과 일치하도록 `<version>` 속성의 값을 변경합니다. 예를 들어, Fuse on EAP 설치 디렉터리 이름이 `2.0.0.fuse-760015` 인 경우 버전을 `2.0.0.fuse-760015` 로 설정합니다.
 - b. `<relativePath>../..</relativePath>` 행을 제거합니다.
예를 들면 다음과 같습니다.

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

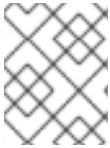
6. 터미널 창에서 다음 명령을 실행하여 `branding-plugin` 프로젝트를 빌드합니다.

```
mvn clean install
```

이 명령은 프로젝트의 `/target` 폴더에 `branding-plugin.war` 파일을 생성합니다.

7. `branding-plugin.war` 파일을 EAP 설치의 `standalone/deployments` 디렉터리에 복사합니다.
8. Fuse가 아직 실행되고 있지 않은 경우 다음 명령을 실행하여 시작합니다.
On Linux/Mac OS: `./bin/standalone.sh`

Windows의 경우: `./bin/standalone.skip`
9. 웹 브라우저에서 시작 명령이 이전 단계에서 반환된 URL을 사용하여 Fuse 콘솔을 엽니다(기본 URL은 <http://localhost:8080/hawtio>).



참고

웹 브라우저에서 Fuse Console을 이미 실행한 경우 브랜드가 브라우저의 로컬 스토리지에 저장됩니다. 새로운 브랜딩 설정을 사용하려면 브라우저의 로컬 스토리지를 지워야 합니다.

4.4. FUSE CONSOLE 보안

EAP에서 Fuse 콘솔을 보호하려면 다음을 수행할 수 있습니다.

- 필요한 프로토콜로 HTTPS 설정
`hawtio.http.strictTransportSecurity` 속성을 사용하면 웹 브라우저가 보안 HTTPS 프로토콜을 사용하여 Fuse Console에 액세스하도록 할 수 있습니다. 이 속성은 HTTP를 사용하여 Fuse 콘솔에 액세스하려는 웹 브라우저가 HTTPS를 사용하도록 요청을 자동으로 변환해야 함을 지정합니다.
- 공개 키를 사용하여 응답 보안
`hawtio.http.publicKeyPins` 속성을 사용하여 특정 암호화 공개 키를 Fuse 콘솔과 연결하도록 웹 브라우저에 고정 인증서와 "man-in-the-middle" 공격 위험을 줄임으로써 HTTPS 프로토콜을 보호할 수 있습니다.

절차

다음 예와 같이 `$EAP_HOME/standalone/configuration/standalone*.xml` 파일의 `system-properties` 섹션에 있는 `hawtio.http.publicKeyPins` 속성을 설정합니다.

```
<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"/>
```

추가 리소스

- `hawtio.http.strictTransportSecurity` 속성 구문에 대한 설명은 [HSTS\(HTTP Strict Transport Security\)](#) 응답 헤더에 대한 설명 페이지를 참조하십시오.
- Base64로 인코딩된 공개 키를 추출하는 방법에 대한 지침을 포함하여 `hawtio.http.publicKeyPins` 속성 구문에 대한 설명은 [HTTP 공개 키 고정](#) 응답 헤더에 대한 설명

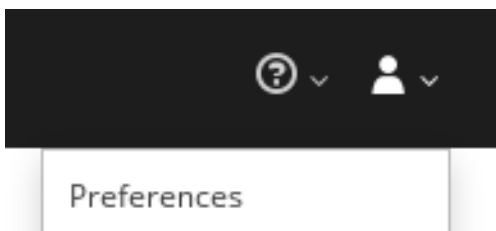
페이지를 참조하십시오.

4.5. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.

Fuse Console의 대기열 및 연결 표시에 큐가 없거나 연결이 누락되었거나 일관성 없는 아이콘을 표시하는 경우 Jolokia 컬렉션 크기 매개변수를 조정하여 Jolokia marshals를 응답에 마샬링하는 배열의 최대 요소 수를 지정합니다.

절차

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 최대 컬렉션 크기 옵션의 값을 늘립니다(기본값은 50,000).
3. 닫기를 클릭합니다.

4.6. FUSE 콘솔 비활성화

다른 구성 요소에 영향을 주지 않고 모든 사용자가 액세스할 수 없도록 JBoss EAP에서 Fuse 콘솔을 비활성화할 수 있습니다.

절차

JBoss EAP에서 Fuse 콘솔을 비활성화하려면 다음 중 하나를 수행하십시오.

- Fuse Console 배포 파일 제거: `$EAP_HOME/standalone/deployments/hawtio-wildfly-xxxxx.war`
- JBoss EAP 관리 콘솔 또는 명령줄 인터페이스를 사용하여 Fuse 콘솔 배포를 취소합니다.

4.7. 원격 FUSE 애플리케이션에 연결

Fuse 콘솔은 클라이언트에 추가 소프트웨어(에이전트)가 설치되어 있어야 하는 **JMX(Java Management Extensions)**에 대한 에이전트 기반 접근 방식인 **Jolokia**를 사용합니다. 기본적으로 **Red Hat Fuse**에는 **jolokia** 에이전트가 포함되어 있습니다.

독립 실행형 **Fuse Console** 배포를 사용하면 이미 **jolokia** 에이전트가 있는 원격 통합에 연결할 수 있습니다(<https://jolokia.org/>). 연결하려는 프로세스에는 **jolokia** 에이전트가 없는 경우 **jolokia** 문서 (<http://jolokia.org/agent.html>)를 참조하십시오.

4.7.1. Fuse 콘솔 잠금 해제

기본적으로 **JBoss EAP**에서 **Fuse 7** 독립 실행형용 **Jolokia**는 잠겼으며 **Fuse Console**은 원격으로 액세스할 수 없습니다.

localhost 또는 **127.0.0.1** 이외의 호스트 이름 또는 **IP** 주소에 대해 **Fuse Console**의 잠금을 해제하려면 다음 단계를 따르십시오.

1. 편집기에서 **\$EAP_HOME/standalone/configuration/jolokia-access.xml** 파일을 엽니다.
2. **< cors >** 섹션에 추가하여 **Fuse** 콘솔로 액세스할 **Fuse** 통합의 호스트 이름 또는 **IP** 주소를 등록합니다.

예를 들어 **Fuse** 콘솔에서 호스트 이름 **0.0.0.3** 에 액세스하려면 다음을 추가합니다.

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

다음과 같이 행합니다.

```
<!--
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.

파일을 저장합니다.

4.7.2. 원격 액세스 제한

선택적으로 특정 호스트 및 IP 주소에 대해 **Fuse Console**에 대한 원격 액세스를 제한할 수 있습니다.

HTTP 클라이언트의 **IP** 주소를 기반으로 전체 액세스 권한을 부여할 수 있습니다. 다음 제한 사항을 지정하려면 다음을 수행합니다.

jolokia-access.xml 파일에서 하나 이상의 **< host >** 요소가 포함된 **< remote >** 섹션을 추가하거나 편집합니다. **< host >** 요소의 경우 **CIDR** 형식으로 지정된 **IP** 주소, 호스트 이름 또는 넷마스크를 지정할 수 있습니다(예: **10.0** 네트워크에서 들어오는 모든 클라이언트의 경우 **10.0.0.0/16**).

다음 예제에서는 **localhost**와 **IP** 주소가 **10.0** 으로 시작하는 모든 클라이언트에서 액세스할 수 있습니다. 다른 모든 **IP** 주소의 경우 액세스가 거부됩니다.

```

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

```

자세한 내용은 Jolokia 보안 문서 (<https://jolokia.org/reference/html/security.html>)를 참조하십시오.

4.7.3. 원격 Fuse 인스턴스에 연결 허용

Fuse 콘솔의 프록시 서블릿은 허용 목록 호스트 보호를 사용하며 기본적으로 Fuse Console은 localhost에만 연결할 수 있습니다. Fuse Console을 다른 원격 Fuse 인스턴스에 연결하려면 standalone/configuration/standalone-*.xml 파일에서 다음과 같은 구성을 변경합니다.

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

4.7.4. 원격 Jolokia 에이전트에 연결

시작하기 전에 원격 Jolokia 에이전트의 연결 세부 정보(호스트 이름, 포트 및 경로)를 알아야 합니다.

Red Hat JBoss EAP의 Jolokia 에이전트의 기본 연결 URL은 <http://<host>:8080/hawtio/jolokia>입니다.

시스템 관리자는 이 기본값을 변경할 수 있습니다.

일반적으로 Jolokia 에이전트에 원격으로 연결하는 URL은 Fuse Console과 /jolokia 를 여는 URL입니다. 예를 들어, Fuse Console을 여는 URL이 <http://<host>:1234/hawtio> 인 경우 원격으로 연결할 URL은 <http://<host>:1234/hawtio/jolokia> 일 것입니다.

JVM을 검사할 수 있도록 원격 Jolokia 인스턴스에 연결하려면 다음을 수행합니다.

1. 연결 탭을 클릭합니다.
2. 원격 탭을 클릭한 다음 연결 추가 를 클릭합니다.
3. 이름, 스키마 (HTTP 또는 HTTPS) 및 호스트 이름을 입력합니다.
4. 연결 테스트를 클릭합니다.

5. 추가를 클릭합니다.



참고

Fuse 콘솔은 localhost 및 127.0.0.1 이외의 로컬 네트워크 인터페이스를 자동으로 조사하고 허용 목록에 추가합니다. 따라서 로컬 머신의 주소를 허용 목록에 수동으로 등록할 필요가 없습니다.

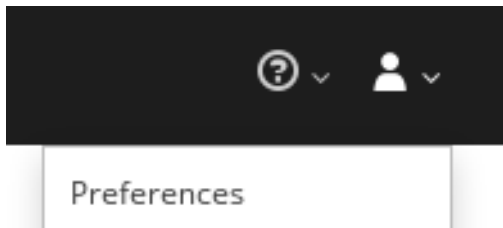
4.7.5. 데이터 이동 기본 설정

예를 들어 Fuse Console에 표시되는 데이터를 더 자주 새로 고치려는 경우 다음과 같은 Jolokia 기본 설정을 변경할 수 있습니다. 데이터 업데이트 빈도를 늘리면 네트워킹 트래픽에 영향을 미치고 서버에 대한 요청 수를 늘립니다.

- 업데이트 속도 - Jolokia에 폴링하여 CRYostat 데이터를 가져오는 간격(기본값은 5초)입니다.
- 최대 깊이 - Jolokia가 반환하기 전에 서버 측의 JSON으로 객체를 마샬링하는 수준 (기본값은 7)입니다.
- 최대 컬렉션 크기 - Jolokia가 응답에서 마샬링하는 배열의 최대 요소 수입니다(기본값은 50,000).

이러한 설정의 값을 변경하려면 다음을 수행합니다.

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 옵션을 편집한 다음 단기를 클릭합니다.

4.7.6. JVM 런타임 정보 보기

시스템 속성, 메트릭 및 스레드와 같은 **JVM** 런타임 정보를 보려면 런타임 탭을 클릭합니다.

4.8. APACHE CAMEL 애플리케이션 보기 및 관리

Fuse 콘솔의 **Camel** 탭에서 **Apache Camel** 컨텍스트, 경로 및 종속성을 보고 관리합니다.

다음 세부 정보를 볼 수 있습니다.

- 실행 중인 모든 **Camel** 컨텍스트 목록
- **Camel** 버전 번호 및 런타임 정적과 같은 각 **Camel** 컨텍스트에 대한 자세한 정보
- 각 **Camel** 애플리케이션의 모든 경로 목록 및 런타임 통계
- 실행 중인 경로의 그래픽 표현과 실시간 메트릭

다음은 통해 **Camel** 애플리케이션과 상호 작용할 수도 있습니다.

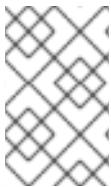
- 컨텍스트 시작 및 일시 중단
- 모든 **Camel** 애플리케이션 및 해당 경로의 라이프사이클을 관리하여 재시작, 중지, 일시 중지, 재개 등을 수행할 수 있습니다.
- 실행 중인 경로의 실시간 추적 및 디버깅
- **Camel** 엔드포인트에 메시지 검색 및 전송

사전 요구 사항

Camel 탭은 하나 이상의 **Camel** 경로를 사용하는 컨테이너에 연결할 때만 사용할 수 있습니다.

4.8.1. 컨텍스트 시작, 일시 중지 또는 삭제

1. **Camel** 탭의 트리 보기에서 **Camel Contexts** 를 클릭합니다.
2. 목록에서 하나 이상의 컨텍스트 옆에 있는 확인란을 선택합니다.
3. 시작 또는 일시 중지 를 클릭합니다.
4. 컨텍스트를 삭제하려면 다음을 수행합니다.
 - a. 컨텍스트를 중지합니다.
 - b. 아이콘을 클릭한 다음 드롭다운 메뉴에서 삭제 를 선택합니다.



참고

컨텍스트를 삭제하면 배포된 애플리케이션에서 해당 컨텍스트를 제거합니다.

4.8.2. Camel 애플리케이션 세부 정보 보기

1. **Camel** 탭의 트리 뷰에서 **Camel** 애플리케이션을 클릭합니다.
2. 애플리케이션 특성 및 값 목록을 보려면 속성을 클릭합니다.
3. 애플리케이션 특성의 그래픽 표시를 보려면 차트 를 클릭한 다음 편집을 클릭하여 차트에서 볼 속성을 선택합니다.
4. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
- 5.

애플리케이션 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향으로 목록을 필터링할 수 있습니다.

6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 **Camel** 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 유형 다운로드를 클릭합니다.
7. XML에서 경로 추가 또는 업데이트 또는 **classpath**에서 사용 가능한 모든 **Camel** 구성 요소를 찾는 등 **Cryostat** 작업을 보고 실행하려면 **Operations** 를 클릭합니다.

4.8.3. Camel 경로 목록 보기 및 상호 작용

1. 경로 목록을 보려면 다음을 수행합니다.
 - a. **Camel** 탭을 클릭합니다.
 - b. 트리 뷰에서 애플리케이션의 경로 폴더를 클릭합니다.

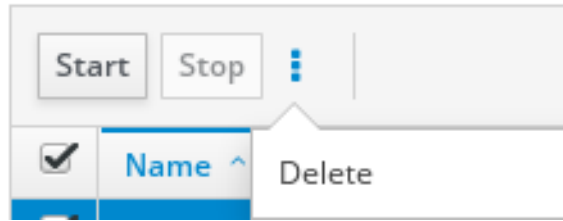
Routes

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

2. 하나 이상의 경로를 시작, 중지 또는 삭제하려면 다음을 수행합니다.
 - a. 목록에서 하나 이상의 경로 옆에 있는 확인란을 선택합니다.
 - b. 시작 또는 중지를 클릭합니다.
 - c.

경로를 삭제하려면 먼저 중지해야 합니다. 그런 다음 아이콘을 클릭하고 드롭다운 메뉴에서 삭제를 선택합니다.

Routes



참고

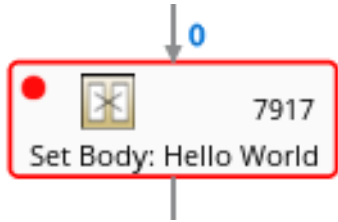
- 경로를 삭제하면 배포된 애플리케이션에서 해당 경로를 제거합니다.
- 트리 보기에서 특정 경로를 선택한 다음 오른쪽 상단 메뉴를 클릭하여 시작, 중지 또는 삭제할 수도 있습니다.

3. 경로의 그래픽 다이어그램을 보려면 **Route Diagram** 을 클릭합니다.
4. 진행 중 및 차단된 교환을 보려면 **교환** 을 클릭합니다.
5. 엔드포인트를 보려면 끝점을 클릭합니다. **URL**, 경로 **ID** 및 방향별로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 **Camel** 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 **Type Cryostat**를 클릭합니다.
7. 특정 경로와 상호 작용하려면 다음을 수행합니다.
 - a. **Camel** 탭의 트리 보기에서 경로를 선택합니다.

- b. 경로 속성 및 값 목록을 보려면 속성을 클릭합니다.
 - c. 경로 속성의 그래픽 표시를 보려면 차트를 클릭합니다. 편집을 클릭하여 차트에서 표시할 속성을 선택할 수 있습니다.
 - d. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
 - e. 작업에서 경로를 **XML**로 덤프하거나 경로의 **Camel ID** 값을 가져오는 등 경로에서 **Cryostat** 작업을 보고 실행합니다.
8. 경로를 통해 메시지를 추적하려면 다음을 수행합니다.
- a. **Camel** 탭의 트리 보기에서 경로를 선택합니다.
 - b. 추적을 선택한 다음 추적 시작을 클릭합니다.
9. 경로에 메시지를 보내려면 다음을 수행합니다.
- a. **Camel** 탭의 트리 뷰에서 컨텍스트의 엔드포인트 폴더를 연 다음 엔드포인트를 선택합니다.
 - b. **Send** 를 클릭합니다.
 - c. 메시지를 **JSON** 또는 **XML** 형식으로 구성합니다.
 - d. 전송을 클릭합니다.
 - e. 경로의 추적 탭으로 돌아가 경로를 통한 메시지 흐름을 확인합니다.

4.8.4. 경로 디버깅

1. **Camel** 탭의 트리 보기에서 경로를 선택합니다.
2. 디버그를 선택한 다음 디버깅 시작을 클릭합니다.
3. **Cryostat**를 추가하려면 다이어그램에서 노드를 선택한 다음 **Add Cryostat**를 클릭합니다. 노드에 빨간색 점이 표시됩니다.



노드가 **Cryostat** 목록에 추가됩니다.

Breakpoints	
setBody1	✕
log1	✕

4. 아래쪽 화살표를 클릭하여 다음 노드 또는 재생 버튼으로 이동하여 경로 실행을 다시 시작합니다.
5. 일시 중지 버튼을 클릭하여 경로에 대한 모든 스레드를 일시 중지합니다.
6. 완료되면 디버깅 중지 를 클릭합니다. 모든 **Cryostat**가 지워집니다.

4.9. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리

JMX(Java Management Extensions)는 런타임 시 리소스(서비스, 장치 및 애플리케이션)를 동적으로 관리할 수 있는 **Java** 기술입니다. 리소스는 **Cryostats**라는 오브젝트로 표시됩니다(관리됨의 경우). 리소스를 생성, 구현 또는 설치하는 즉시 리소스를 관리하고 모니터링할 수 있습니다.

Fuse Console에서 **Cryostat** 플러그인을 사용하면 **Cryostat** 도메인 및 **Cryostat**를 보고 관리할 수 있습니다. **Cryostat** 특성을 보고, 명령을 실행하고, 통계를 표시하는 차트를 만들 수 있습니다.

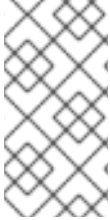
Cryo stat 탭에서는 폴더에 구성된 활성 **Cryostat** 도메인 및 **Cryostat**의 트리 뷰를 제공합니다. **Cryostat**에서 세부 정보를 보고 명령을 실행할 수 있습니다.

절차

1. **Cryostat** 특성을 보고 편집하려면 다음을 수행합니다.
 - a. 트리 보기에서 **na**를 선택합니다.**In the tree view, select an value.**
 - b. 특성 탭을 클릭합니다.
 - c. 속성을 클릭하여 세부 정보를 확인합니다.
2. 작업을 수행하려면 다음을 수행합니다.
 - a. 트리 보기에서 **na**를 선택합니다.**In the tree view, select an value.**
 - b. **Operations** 탭을 클릭하고 나열된 작업 중 하나를 확장합니다.
 - c. 실행을 클릭하여 작업을 실행합니다.
3. 차트를 보려면 다음을 수행합니다.
 - a. 트리 뷰에서 항목을 선택합니다.
 - b. 차트 탭을 클릭합니다.

4.10. 진단 보기

JVM Diagnostic Command 및 **CryostatDiangostic** 인터페이스를 통해 **JVM**에 대한 진단 정보를 보려면 **Cryostat** 탭을 사용합니다.



참고

기능은 **jmc(Java Mission Control)** 또는 명령줄 도구 **jcmd**의 진단 명령 보기와 유사합니다. 플러그인은 일부 시나리오에서 해당 **jcmd** 명령을 제공합니다.

절차

1. 로드된 클래스의 인스턴스 수와 필요한 바이트 수를 검색하려면 클래스 **histogram**을 클릭합니다. 작업이 반복되면 탭에 마지막 실행 이후의 차이점이 표시됩니다.
2. **JVM** 진단 플래그 설정을 보려면 **JVM** 플래그 를 클릭합니다.
3. 실행 중인 **JVM**의 경우 플래그 설정도 수정할 수 있습니다.

추가 리소스

지원되는 **JVM**은 플랫폼에 따라 다릅니다. 자세한 내용은 다음 소스 중 하나로 이동합니다.

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

4.11. 스레드 보기

스레드 상태를 보고 모니터링할 수 있습니다.

절차

1. **Runtime** 탭을 클릭한 다음 **Threads** 를 클릭합니다. **Threads** 페이지에는 각 스레드에 대한 활성 스레드 및 스택 추적 세부 정보가 나열됩니다. 기본적으로 스레드 목록은 모든 스레드를 내

림차순으로 표시합니다.

2. **ID**를 눌러 목록을 정렬하려면 **ID 열 레이블**을 클릭합니다.
3. 선택적으로 스레드 상태(예: **Blocked**) 또는 스레드 이름으로 목록을 필터링합니다.
4. 해당 스레드의 잠금 클래스 이름 및 전체 스택 추적과 같은 특정 스레드에 대한 자세한 정보를 드릴다운하려면 **Actions** 열에서 **More** 를 클릭합니다.

4.12. 로그 항목 보기

로그 탭에서 **Red Hat Fuse**의 로그 항목을 볼 수 있습니다.

사전 요구 사항

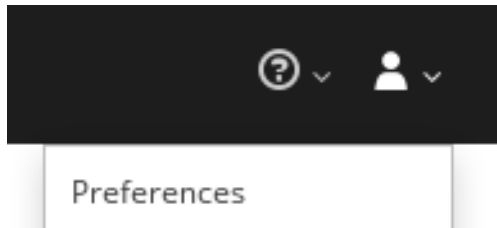
로그 탭은 **Java** 애플리케이션에 로그가 포함된 경우 사용할 수 있습니다.

절차

1. 로그 항목 목록을 보려면 로그 항목 탭을 클릭합니다.

기본적으로 목록에는 로그 항목이 오름차순으로 표시됩니다.

각 로그 항목으로 드릴다운하여 로그 항목에 대한 자세한 정보를 볼 수 있습니다.
2. 특정 로그 유형을 표시하도록 로그 목록을 필터링하려면 작업 표시줄 을 클릭합니다. 텍스트 문자열 또는 로깅 수준에 따라 로그 항목 섹션을 필터링할 수 있습니다.
3. **Fuse Console** 기본 설정을 변경하려면 다음을 수행합니다.
 - a. **Fuse Console**의 오른쪽 상단에 있는 사용자 아이콘을 클릭한 다음 드롭다운 메뉴에서 **Preferences** 를 클릭합니다.



- b. 기본 정렬 순서를 변경하려면 서버 로그를 선택한 다음 로그 항목 링크를 클릭하여 번들 이름, 스레드 및 전체 메시지 텍스트와 같은 로그 항목에 대한 세부 정보를 드릴다운합니다.
- c. 선택적으로 로그 메시지를 저장하기 위해 이러한 설정을 사용자 지정할 수 있습니다.
- **Fuse Console**에 보관할 로그 문의 수입니다(기본값: 100).
 - 글로벌 로그 수준: **INFO** (기본값), **OFF**, **ERROR**, **WARN**, **DEBUG**.
 - **hawtio-oauth** 및 **hawtio-core-utils** 와 같이 포함할 하위 수준 메시지입니다.
- d. **Fuse** 콘솔 로그 설정을 기본값으로 재설정하려면 재설정 → 설정 재설정 을 클릭합니다.

부록 A. FUSE 콘솔 구현 구성 속성

기본적으로 **Fuse Console** 브랜딩은 **Fuse Console WAR** 파일에 있는 **hawtconfig.json** 파일에 정의되어 있습니다. **Fuse Console** 브랜딩 플러그인을 사용하여 제목, 로고 및 로그인 페이지 정보와 같은 **Fuse Console** 브랜딩 정보를 사용자 지정할 수 있습니다.

표 A.1. “Fuse 콘솔 구현 구성 속성” 각 속성에 값이 필요한지 여부에 관계없이 속성 및 목록에 대한 설명을 제공합니다.

표 A.1. Fuse 콘솔 구현 구성 속성

섹션	속성 이름	기본 hawtconfig.json 파일의 값	설명	Value Required?
브랜딩	appName	Red Hat Fuse Management Console	애플리케이션 이름입니다. 이 이름은 Fuse Console의 제목 표시줄에 표시됩니다.	필수 항목
	appLogoUrl	img/Logo-Red_Hat-Fuse-A-Reverse- RGB.png	애플리케이션 로고 이미지 파일의 경로입니다.	필수 항목
	companyLogoUrl	img/Logo-RedHat-A-Reverse- RGB.png	회사 로고 이미지 파일의 경로입니다.	필수 항목
login	description	빈 값	Fuse 콘솔 로그인 페이지에 표시되는 설명 텍스트(예: http://localhost:8181/hawtio).	선택 사항
	링크	[]	"url" 및 "text" 쌍의 배열을 지정하여 사용자가 더 많은 정보 또는 도움말을 가져올 수 있는 페이지에 대한 추가 링크를 제공합니다.	선택 사항
정보	title	Red Hat Fuse Management Console	Fuse Console의 정보 페이지에 표시되는 제목입니다.	필수 항목

섹션	속성 이름	기본 hawtconfig.json 파일의 값	설명	Value Required?
	productInfo	빈 값	Fuse 콘솔의 정보 페이지에 표시되는 제품 정보입니다.	선택 사항
	additionalInfo	빈 값	Fuse Console의 정보 페이지에 표시되는 추가 정보입니다.	선택 사항
	저작권	빈 값	Fuse Console의 정보 페이지에 표시되는 저작권 정보.	선택 사항
	imgSrc	img/Logo-RedHat-A-Reverse- RGB.png	Fuse 콘솔의 정보 페이지에 표시되는 이미지입니다.	필수 항목
disabledRoutes	<i>none</i>	[]	콘솔에서 특정 경로 (예: 플러그인)를 비활성화합니다. 이 섹션을 변경하지 마십시오. 모든 변경 사항은 지원되지 않습니다.	필수 항목