

Fuseサンプル - AMQ Brokerへ(AMQP)メッセージ送受信

- [前提条件](#)
- [本サンプルについて](#)
- [Camel Routeの処理概要](#)
 - [1. amqp-producer](#)
 - [2. amqp-consumer](#)
- [本サンプルの実行方法](#)
 - [1. ローカルPCで実行する場合](#)
 - [2. Openshift環境へデプロイして実行する場合](#)
- [ゼロから作成の手順](#)

前提条件

- AMQ Broker 7.x (ローカル、またはOpenshift上)
- Fuse 7.x (ローカル、またはOpenshift上)

本サンプルについて

本サンプルでは、[camel-amqpコンポーネント](#)を使って、AMQ Broker 7.xのキューへ送受信の実装方法を示します。

[camel-amqp](#)のオプションが非常に多いですが、今回は初心者向けのため、最低限のAMQP接続設定だけ行っています。詳細なオプションはリンク先の製品ドキュメントをご参照ください。

Camel Routeの処理概要

1. amqp-producer

このCamel Routeは、2秒間隔で3桁の乱数を生成します。生成された乱数がAMQPメッセージのボディとなり、AMQPプロトコルを利用してAMQ-Broker上の `demoQueue` へ送信します。送信が成功すると、以下のログ出力します。

```
INFO amqp-producer - SEND AMQP request. 779
```

2. amqp-consumer

このCamel Routeは、AMQ-Broker上の demoQueue をリスニングします。キューに到達したメッセージをトリガーに、取得したメッセージのボディをログに出力します。

```
INFO amqp-consumer - GOT AMQP request. 779
```

本サンプルの実行方法

1. ローカルPCで実行する場合

前提条件:

- AMQ Broker 7.5 がローカルPCで構築済み、起動していること
- Maven 3.x がローカルPCでインストール済みであること
- Maven Remote Repositoryがアクセスできること

実行手順:

1. AMQP接続情報を設定します。

application.properties へ以下の内容を追加してください。

```
# AMQP connectivity
amqp.uri=amqp://localhost:5672
amqp.username=admin
amqp.password=admin
```

※ username と password は自分のAMQ設定に合わせて変更すること

2. コマンドプロンプトにて実行します。

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

3. コマンドプロンプトで2秒間隔に以下のような出力を確認してください。

```
... INFO amqp-producer - SEND AMQP request. 779
... INFO amqp-consumer - GOT AMQP request. 779
```

2. Openshift環境へデプロイして実行する場合

前提条件:

- AMQ Broker 7.5 がOpenshiftで構築済み、起動していること
- Maven 3.x がローカルPCでインストール済みであること
- Maven Remote Repositoryがアクセスできること
- [oc\(openshift client\)](#)がローカルPCでインストール済みであること

実行手順:

1. AMQP接続情報を設定します。

application.properties へ以下の内容を追加してください。

```
# AMQP connectivity
amqp.uri=amqp://broker-amq-amqp.amq-demo-non-ssl.svc.cluster.local:5672
amqp.username=admin
amqp.password=admin
```

※ 上記usernameとpasswordはOpenshift上のAMQ設定に合わせて変更すること

※ 上記amqp.url内の"amq-demo-non-ssl"は、Openshift上のAMQプロジェクト名に置き換えてください

2. ocコマンドでOpenshiftへログインして、AMQ Brokerプロジェクトに切り替えします。

```
oc login https://openshift-server:port
oc project amq-demo-non-ssl
```

※ 上記の"amq-demo-non-ssl"は、Openshift上のAMQプロジェクト名に置き換えてください

3. コマンドプロンプトにて実行します。

このコマンドでは、ローカルPCのocコマンドでログインしたOpenshiftプロジェクト(AMQ Brokerのプロジェクト)にFuseアプリをデプロイします。

```
mvn fabric8:deploy -Popenshift
```

4. コマンドプロンプトで以下のような出力を確認してください。

```
[INFO] F8: Pushing image docker-registry.default.
...
[INFO] F8: Build fuse-amq-demo-s2i-4 in status Complete
...
[INFO] Updated DeploymentConfig: target/fabric8/applyJson/amq-demo-non-ssl/deploymentconfig-fuse-a
...

-----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 47.076 s
[INFO] Finished at: 2020-02-14T14:13:18+09:00
[INFO] -----
```

ゼロから作成の手順

以下は本サンプルをゼロから作成する場合の手順を説明します。

※ 本手順の実施には、インターネット接続が必要。

1. Maven archetypeより、Fuseアプリの雛形を生成します。

```
# Config archetype to use
archetypeVersion=2.2.0.fuse-sb2-750011-redhat-00006
archetypeCatalog=https://maven.repository.redhat.com/ga/io/fabric8/archetypes/archetype:

# Config Fuse project name to generate
MY_PROJECT=fuse-amq-demo

# Generate from archetype, which is SpringBoot2 based Camel Application using XML DSL
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate -B \
  -DarchetypeCatalog=${archetypeCatalog} \
  -DarchetypeGroupId=org.jboss.fuse.fis.archetypes \
  -DarchetypeVersion=${archetypeVersion} \
  -DarchetypeArtifactId=spring-boot-camel-xml-archetype \
  -DgroupId=com.sample \
  -DartifactId=${MY_PROJECT} \
  -DoutputDirectory=${MY_PROJECT}
```

2. pom.xml へ依存ライブラリcamel-amqpを追加

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-amqp</artifactId>
</dependency>
```

3. src/main/resources/spring/camel-context.xml へ <camelContext> の前に、bean定義を追加

```
<!-- AMQP connectivity -->
<bean id="jmsConnectionFactory"
  class="org.apache.qpid.jms.JmsConnectionFactory"
  primary="true">
  <property name="remoteURI" value="${amqp.uri}" />
  <property name="username" value="${amqp.username}" />
  <property name="password" value="${amqp.password}" />
</bean>
<bean id="jmsCachingConnectionFactory"
  class="org.springframework.jms.connection.CachingConnectionFactory">
  <property name="targetConnectionFactory" ref="jmsConnectionFactory" />
</bean>
<bean id="jmsConfig"
  class="org.apache.camel.component.jms.JmsConfiguration" >
  <constructor-arg ref="jmsCachingConnectionFactory" />
  <property name="cacheLevelName" value="CACHE_CONSUMER" />
</bean>
<bean id="amqp"
  class="org.apache.camel.component.amqp.AMQPComponent">
  <property name="configuration" ref="jmsConfig" />
</bean>
```

4. src/main/resources/spring/camel-context.xml <camelContext> 内のCamel Routeをすべて削除して、以下の2つRouteを追加

```
<route id="amqp-producer">
  <from id="route-timer" uri="timer:foo?period=2000"/>
  <transform id="route-transform">
    <method ref="myTransformer"/>
  </transform>
  <to uri="amqp:queue:demoQueue"/>
  <log id="route-log" message="SEND AMQP request. ${body}"/>
</route>

<route id="amqp-consumer">
  <from uri="amqp:queue:demoQueue"/>
  <log message="GOT AMQP request. ${body}"/>
</route>
```

5. src/main/resources/application.properties へAMQP接続情報を追加

```
# AMQP connectivity
amqp.uri=amqp://localhost:5672
amqp.username=admin
amqp.password=admin
```