# ActiveMQ Authentication and Setup

## Authentication

The particular Docker image of ActiveMQ that I'm currently using doesn't support authentication by default, you will need to enable the authentication plugin in order to enforce some kind of authentication.

To do so you will need to add the following plugin into the `activemq.xml` file and under the `broker` section tag.

**activemq.xml**

```
    <plugins>
       <simpleAuthenticationPlugin anonymousAccessAllowed="false">
         <users>
             <authenticationUser username="system" password="system"
groups="users,admins"/>
             <authenticationUser username="test01" password="test01"
groups="users,admins"/>
         </users>
       </simpleAuthenticationPlugin>
    </plugins>
```

You can create your own Docker image and copy this particular file overriding the default one that's in the container.

An example of Docker image and Docker compose file:

**Docker image**

```
FROM symptoma/activemq:5.15.13


COPY ./activemq.xml /opt/apache-activemq-5.15.13/conf/activemq.xml
```

**Docker compose**

```
services:
  amq:
    build: .
    ports:
      - 61616:61616
      - 8161:8161
```

Now you will actually have to provide a valid username and password in order to access to the broker, without this you can access the broker without any authentication.

To set the username and password with Spring Boot JMS you can create a ConnectionFactory bean like below

```
@Bean
public ConnectionFactory connetionFactory() {
    var factory = new ActiveMQConnectionFactory();
    factory.setUserName("system");
    factory.setPassword("system");
    factory.setBrokerURL("tcp://localhost:61616");
    return factory;
}
```

Then you can do a constructor injection of `JmsTemplate` object which gives you access to send and receive messages from the ActiveMQ.

> For whatever reason if you don't configure your own ConnectionFactory and instead use the default auto-configured ConnectionFactory the Spring boot application will actually not terminate immediately after your Spring Boot application finishes. Perhaps this isn't a problem if your application is web application, but for those application that just runs and finishes you will want to consider the option of configuring your own ConnectionFactory.

---

Revision #4
Created 2023-11-10 13:40:41 UTC by Tamarine
Updated 2023-11-10 14:00:15 UTC by Tamarine