

# APS RabbitMQ Message Service Provider

User Guide

Version: 1.0.0

Author: Tommy Svensson

Copyright © 2012 Natusoft AB

## Table of Contents

<b>1 APS RabbitMQ Message Service Provider</b>	<b>1</b>
1.1 APSMessageService API	1

# 1 APS RabbitMQ Message Service Provider

This service provides an implementation of APSMessageService using RabbitMQ.

**Note:** This implementation does not support *contentType* in the API. When sending messages the *contentType* will be ignored, and when messages are received the *contentType* will always be "UNKNOWN".

A good suggestion is to always use JSON or XML as content.

## 1.1 APSMessageService API

---

Javadoc

```
public interface APSSimpleMessageService [se.natusoft.osgi.aps.api.net.messaging.service] {
```

This defines a simple message service. Can be implemented by using a message bus like RabbitMQ, Active MQ, etc or just a simple tcpip server or whatever.

Since the actual members are outside of this service API, it doesn't really know who they are and doesn't care, all members are defined by configuration.

The term 'target' is used in this API. This can be anything relevant for routing a message to its destination. It depends a lot on the underlying implementation. JMS for example have both a Queues that can be sent to using queue names, but also topics. If JMS is used then the target can be either a queue or a topic. The target just represents some destination of the message.

**@SuppressWarnings("unused") String APS\_MESSAGE\_SERVICE\_PROVIDER = "aps-message-service-provider"**

Multiple providers of this service can be deployed at the same time. Using this property when registering services for a provider allows clients to lookup a specific provider.

**void messageReceived(String target, byte[] message)**

This is called when a message is received.

*Parameters*

*target* - The target the message was sent to.

*message* - The received message.

**void addMessageListener(String target, MessageListener listener)**

Adds a listener for types.

*Parameters*

*target* - The target to listen to.

*listener* - The listener to add.

**void removeMessageListener(String target, MessageListener listener)**

Removes a messaging listener.

*Parameters*

*target* - The target to stop listening to.

*listener* - The listener to remove.

**void sendMessage(String target, byte[] message) throws APSMessagingException**

Sends a message.

*Parameters*

*target* - The target of the message.

*message* - The message to send.

*Throws*

*APSMessagingException* - on failure.

}

---