APS RabbitMQ Message Service Provider

User Guide

Version: 1.0.0

Author: Tommy Svensson

Table of Contents

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* **APSSimpleClusterService** [se.natusoft.osgi.aps.api.net.messaging.service] {

This service defines a synchronized cluster.

void clusterUpdated(String clusterName, String name, TypedData data)

Receives an updated value.

Parameters

clusterName *- The name of the cluster the updated data belongs to.*

name *- The name of the updated data.*

data *- The updated data.*

void provideData(String clusterName, String name, TypedData typedData)

Creates/updates a value in a cluster.

Parameters

clusterName *- The name of a cluster to store in.*

name *- The name of the value to store.*

typedData *- The value to store.*

Throws

IllegalArgumentException *- on any problem with clusterName.*

TypedData retrieveData(String clusterName, String name)

Gets a value stored in a named cluster. Returns null if it does not exists.

Parameters

clusterName *- The name of the cluster to get data from.*

name *- The name of the cluster data to get.*

void addUpdateListener(String clusterName, UpdateListener updateListener)

Adds an update listener.

Parameters

clusterName *- The name of the cluster to listen for changes in.*

updateListener *- The update listener to add.*

void removeUpdateListener(String clusterName, UpdateListener updateListener)

Removes an update listener.

Parameters

clusterName *- The name of the cluster to remove update listener from.*

updateListener *- The listener to remove.*

Map<String, List<UpdateListener>> listeners = Collections.synchronizedMap(new HashMap<>())

The listeners.

protected void updateListeners(String clusterName, String name, TypedData data)

Updates all listeners.

Parameters

clusterName *- The name of the cluster the updated data belongs to.*

name *- The name of the updated data.*

data *- The actual data.*

protected List<UpdateListener> getListeners(String clusterName)

Returns the listeners.

Parameters

clusterName *- The name of the cluster to get listeners for.*

}

---

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.

void messageReceived(String topic, TypedData message)

This is called when a message is received.

Parameters

> *topic* *- The topic the message belongs to.*
>
> *message* *- The received message.*

void addMessageListener(String topic, MessageListener listener)

Adds a listener for types.

Parameters

> *topic* *- The topic to listen to.*
>
> *listener* *- The listener to add.*

void removeMessageListener(String topic, MessageListener listener)

Removes a messaging listener.

Parameters

> *topic* *- The topic to stop listening to.*
>
> *listener* *- The listener to remove.*

void sendMessage(String topic, TypedData message) throws APSMessagingException

Sends a message.

Parameters

> *topic* *- The topic of the message.*
>
> *message* *- The message to send.*

Throws

> *APSMessagingException* *- on failure.*

protected void sendToListeners(String topic, TypedData message)

Sends a message to the registered listeners.

Parameters

message *- The message to send.*

protected List<MessageListener> lookupMessageListeners(String topic)

Returns the message listeners for a topic.

Parameters

topic *- The topic to get listeners for.*

}