

APS RabbitMQ Message Service Provider

User Guide

Version: 1.0.0

Author: Tommy Svensson

Copyright © 2012 Natusoft AB

Table of Contents

1 APS RabbitMQ Message Service Provider	1
1.1 APSMessageService API	1

1 APS RabbitMQ Message Service Provider

This service provides an implementation of APSMessageService using RabbitMQ.

1.1 APSMessageService API

Javadoc

```
public interface APSClusterService extends APSMessageService
[se.natusoft.osgi.aps.api.net.messaging.service] {
```

All of APSMessageService, APSSynchronizedMapService, and APSSyncService are part of some cluster which they communicate with. This represents that cluster in a general form.

Since APSClusterService extends APSMessageService both are provided by the same bundle. The sync services can also be part of the same bundle, but can also be implemented on top of APSMessageService.

String getName()

Returns the name of the cluster.

boolean isMasterNode()

If the implementation has the notion of a master and this node is the master then true is returned. In all other cases false is returned.

APSCommonDateTime getCommonDateTime()

Returns the network common DateTime that is independent of local machine times.

```
}
```

```
public interface APSMessageService [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 to make a cluster of members.

```
public static final 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.

```
public static final String APS_MESSAGE_SERVICE_INSTANCE_NAME = "aps-message-service-instance-name"
```

Each configured instance of this service should have this property with a unique instance name so that client can lookup a specific instance of the service.

```
void addMessageListener(APSMessageListener listener)
```

Adds a listener for types.

Parameters

listener - The listener to add.

void removeMessageListener(APSMessenger listener)

Removes a messaging listener.

Parameters

listener - The listener to remove.

void sendMessage(APSMessenger message) throws APSMessagingException

Sends a message.

Parameters

message - The message to send.

Throws

se.natusoft.osgi.aps.api.net.messaging.exception.APSMessagingException - on failure.

```
public static abstract class AbstractMessageServiceProvider implements APSMessengerService
[se.natusoft.osgi.aps.api.net.messaging.service] {
```

Provides an abstract implementation of the APSMessengerService interface.

protected void sendToListeners(APSMessenger message)

Sends a message to the registered listeners.

Parameters

message - The message to send.

```
}
```

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

This service makes a synchronized named map available.

Map<String, String> getNamedMap(String name)

Returns a named map into which objects can be stored with a name.

If the named map does not exist it should be created and an empty map be returned.

List<String> getAvailableNamedMaps()

Returns the available names.

boolean supportsNamedMapPersistence()

Returns true if the implementation supports persistence for stored objects. If false is returned objects are in memory only.

```
}
```

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

This defines a data synchronization service.

public static final String SYNC_PROVIDER = "aps-sync-provider"

A property key that should be registered with each service instance to indicate the specific implementation of the service. This to allow multiple implementations to be deployed and clients can ask for a specific if needed.

public static final String SYNC_INSTANCE_NAME = "aps-sync-instance-name"

There should be one service instance registered for each configured synchronization group. Each instance should include this property with a unique name so that clients can get the synchronizer for the correct group.

void syncData(APSSyncDataEvent syncEvent) throws APSMessagingException

Synchronizes data.

Parameters

syncEvent - The sync event to send.

Throws

APSMessagingException - on failure.

void resync()

Triggers a resynchronization.

void resync(String key)

Triggers a resynchronization of the specified key.

Parameters

key - The key to resync.

void addSyncListener(Listener listener)

Adds a synchronization listener.

Parameters

listener - The listener to add.

void removeSyncListener(Listener listener)

Removes a synchronization listener.

Parameters

listener - The listener to remove.

void syncDataReceived(APSSyncEvent syncEvent)

Called to deliver a sync event. This can currently be one of:

- APSSyncDataEvent
- APSReSyncEvent

Parameters

syncEvent - The received sync event.

}
