MigratoryData Presence API

Developer's Guide and Reference Manual

April 3, 2019

Contents

Chapter 1

Developer's Guide

This guide includes the following sections:

1.1 Overview

This Application Programming Interface (API) contains all the necessary operations for building presence extensions for MigratoryData Server.

Before reading this manual, it is recommended to read *MigratoryData Architecture Guide* (PDF, HTML).

1.2 Creating a Presence Extension for MigratoryData Server

A typical API usage is as follows:

1.2.1 Step 1 - Import in your application the API as follows:

```
import com.migratorydata.extension.MigratoryDataPresenceListener;
```

1.2.2 Step 2 - Implement the interface MigratoryDataPresenceListener

The interface MigratoryDataPresenceListener is available under the folder `doc/Extensions/presence/api` of the MigratoryData server installation.

1.2.3 Step 4 - Building the jar of the extension

Build a jar consisting in the object code of the class defined at Step 2. In addition, the jar must also include a folder named `services` in its `META-INF` folder. Moreover, the folder `services` must include a file named `com.migratorydata.extension.MigratoryDataPresenceListener` which must have as content the fully qualified name of the class defined at Step 2.

1.2.4 Step 5 - Install the extension

Rename the JAR built at Step 4 to `presence-extension.jar` and deploy it to the folder `extensions` of your MigratoryData server installation.

1.2.5 Step 6 - Configure the MigratoryData server to load the presence extension

In the configuration file of the MigratoryData server, configure the parameter `Extension.Presence` as follows:

```
Extension.Presence = true
```

For deployments using Custom entitlement rules, there are two additional optional parameters that could be configured `Extension.Presence.Subject` and `Extension.Presence.EntitlementToken` to enable user presence replication across the cluster. The default value of the parameter `Extension.Presence.Subject` is `/__migratorydata__/presence`, and the default value of the parameter `Extension.Presence.EntitlementToken` is the values of the parameter `EntitlementAllowToken`. Your custom entitlement rules should allow subscriptions and publications on the subject and entitlement token defined by these two parameters. See MigratoryData Configuration Guide for further details.

1.2.6 Step 7 - Restart the MigratyData server

1.2.7 Step 8 - Assign the external token to a mobile client

In order for a user to be taken into account by the presence extension, the user needs an external token which is typically used by the presence extension to send messages through an external service such as Firebase Cloud Messaging (FCM) or Apple Push Notification Service (APNS) when the user goes offline (e.g. its mobile app goes to background or its mobile devices enters into sleep mode).

The API method `MigratoryDataClient.setExternalToken()` should be used to attach an external token to a mobile client of the MigratoryData server. This API method is available for the client libraries for Android and iOS.

1.3 Examples

An example built with this API is available in the folder `doc/Extensions/presence/examples` of your MigratoryData server installation. Start with the README file which explains how to compile and run the example.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 MigratoryDataPresenceListener.Message Interface Reference

Public Member Functions

- String getSubject ()
- byte[ ] getContent ()
- Map< String, Object > getAdditionalInfo ()

3.1.1 Detailed Description

Interface to handle messages accepted by the cluster.

3.1.2 Member Function Documentation

3.1.2.1 String MigratoryDataPresenceListener.Message.getSubject ( )

Get the subject of the message.

Returns

the subject of the message

3.1.2.2 byte [ ] MigratoryDataPresenceListener.Message.getContent ( )

Get the content of the message.

Returns

the content of the message

3.1.2.3 Map<String, Object> MigratoryDataPresenceListener.Message.getAdditionalInfo ( )

Get additional attributes of the message as key/value pairs.

Currently the following attributes are available:

- seq - the sequence number assigned by the cluster member which is the coordinator of the subject of message
- epoch - the epoch number assigned by the cluster member which is the coordinator of the subject of message
- closure - the closure of the message assigned by the publisher of the message
- publisherAddress - the Internet address of the publisher of the message

Returns

a map with additional attributes of the message

3.2 MigratoryDataPresenceListener Interface Reference

The implementation of this interface can handle user presence updates such as user connections and user disconnections, and messages received from publishers and accepted by the cluster to be distributed to subscribers.

Classes

- interface Message
- interface User

Public Member Functions

- void onClusterMessage (Message message)
- void onUserPresence (User user)

3.2.1 Detailed Description

The implementation of this interface can handle user presence updates such as user connections and user disconnections, and messages received from publishers and accepted by the cluster to be distributed to subscribers.

It is guaranteed that each message received from a publisher is accepted by exactly one server of the cluster, which is the coordinator of the subject of the message. Publishers may connect to any server of the cluster. However, each message must pass through the coordinator of the subject of the message to be accepted by the cluster.

Thread safety

The methods exposed by this interface are always called from by same thread.

3.2.2 Member Function Documentation

3.2.2.1 void MigratoryDataPresenceListener.onClusterMessage (Message *message*)

This method is called whenever a MigratoryData cluster accepts a message received from a MigratoryData client. The message is always accepted by the cluster member which is the coordinator of the subject of the message. So, this method is called once, only for the coordinator of the subject of the accepted message.

A MigratoryData client may publish a message to any cluster member, however, the MigratoryData cluster will forward the message to the cluster member which is the coordinator of the subject of the message in order to accept the message.

Each cluster member coordinates a distinct subset of subjects. In this way, at any given time, a subject is coordinated by at most one cluster member, always the same until that cluster member fails or is stopped. When a cluster member fails or is stopped, its subjects are automatically redistributed the remaining cluster members such that there are no two cluster members which coordinate simultaneously any given subject.

Parameters

message	the accepted message

3.2.2.2 void MigratoryDataPresenceListener.onUserPresence (User *user*)

This method is called whenever a user connects to or disconnects from a cluster member. This method is also called whenever a connected user changes its list of subscribed subjects (by subscribing to new subjects or unsubscribing from existing subjects). MigratoryData replicates this user presence update across the cluster and calls this method for each cluster member, not only for the cluster member to which the user connected to or disconnected from.

Parameters

user	the details about the user

3.3 MigratoryDataPresenceListener.User Interface Reference

Public Member Functions

- String getExternalToken ()
- long getSessionId ()
- List< String > getSubjects ()
- boolean isOffline ()
- Map< String, Object > getAdditionalInfo ()

3.3.1 Detailed Description

Interface to handle user presence updates.

3.3.2 Member Function Documentation

3.3.2.1 String MigratoryDataPresenceListener.User.getExternalToken ( )

Get the external token of the user.

Returns

the external token of the user.

3.3.2.2 long MigratoryDataPresenceListener.User.getSessionId ( )

Get the session id assigned by the MigratoryData server to the user.

Returns

the session id of the user

3.3.2.3 List<String> MigratoryDataPresenceListener.User.getSubjects ( )

Get the list of subscribed subjects by the user.

Returns

the list of subscribed subjects of the user

3.3.2.4 boolean MigratoryDataPresenceListener.User.isOffline ( )

Determine if the user is offline.

Returns

true if the user is offline, and false otherwise

3.3.2.5 Map<String, Object> MigratoryDataPresenceListener.User.getAdditionalInfo ( )

Get additional attributes of the user presence update as key/value pairs.

Currently the following attributes are available:

- address - the Internet address of the user
- entitlementToken - the entitlement token of the user assigned by the client library
- local - inform whether the user presence update has been generated by the local cluster member or has been received from another cluster member as part of presence replication across the cluster

Returns

a map with additional attributes of the user presence update

Index