Grouper messaging built in

Grouper has a built in messaging system which is not as robust as a messaging middleware product such as ActiveMQ or RabbitMQ.

A message is just a record in a database table which has a timestamp (which determines the ordering), and other fields.

A sender can send to a queue or a topic. A queue has a receiver pulling messages. A topic will just distribute to 1 or more queues.

Receivers need to receive the message then mark it as processed when done.

Message table: grouper_message: If using the default internal messaging with Grouper, this is the table that holds the messages and state of messages

> (i) See Grouper Messaging System Developers Guide for info on how to integrate the Grouper Messaging System with another messaging system.
>
> GSH to manage built in messaging
>
> GSH to send / receive messages

Setup queues

- Queues can be sent to directly or from topics
- A queue shouldnt have the same name as a topic.
- There is a built in folder for queues in grouper: <ATTRIBUTE_ROOT_STEM_CONFIGURED_NAME>:grouperMessageQueues.
- A queue is a permission resource in that folder of attributeDef <ATTRIBUTE_ROOT_STEM_CONFIGURED_NAME>:messages: grouperMessageQueueDef
- The action to send to a topic is "send_to_queue", grant that to a subject who is allowed to send messages to the queue
- The action to receive from a queue is "receive", grant that to a subject who is allowed to pull messages off the queue

```
grouperSession = GrouperSession.startRootSession();

GrouperBuiltinMessagingSystem.createQueue("abc");
GrouperBuiltinMessagingSystem.deleteQueue("abc");


// permissions on objects
GrouperBuiltinMessagingSystem.allowSendToQueue("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.allowReceiveFromQueue("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.disallowSendToQueue("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.disallowReceiveFromQueue("abc", SubjectTestHelper.SUBJ0);
```

Setup topics

- There is a built in folder for topics in grouper: <ATTRIBUTE_ROOT_STEM_CONFIGURED_NAME>:grouperMessageTopics.
- A topic is a permission resource in that folder of attributeDef <ATTRIBUTE_ROOT_STEM_CONFIGURED_NAME>:messages: grouperMessageTopicDef
- The action to send to a topic is "send_to_topic", grant that to a subject who is allowed to send messages to the topic
- You cannot read from a topic, the topic will send to queues, and you can grant that on queues
- To setup the relationship between a topic and queue(s), setup a permission resource implied relationship between the topic and the queues

```
grouperSession = GrouperSession.startRootSession();

GrouperBuiltinMessagingSystem.createTopic("def");
GrouperBuiltinMessagingSystem.deleteTopic("def");

// permissions on objects
GrouperBuiltinMessagingSystem.allowSendToTopic("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.disallowSendToTopic("abc", SubjectTestHelper.SUBJ0);

// topics send to queues
GrouperBuiltinMessagingSystem.topicAddSendToQueue("def", "abc");
Collection<String> queues = GrouperBuiltinMessagingSystem.queuesTopicSendsTo("def");
GrouperBuiltinMessagingSystem.topicRemoveSendToQueue("def", "abc");
```

Performance

- This system (with local HSQL database) can process (send / receive / acknowledge)
    - 125 messages per second in one thread
    - 312 messages per second in 10 threads

Message columns

Column name	Description	Index	Foreign key
ID	db uuid for this row		
HIBERNATE_VERSION_NUMBER	incrementing number so two updates dont occur at once (optimistic locking)		
SENT_TIME_MICROS	microseconds since 1970 this message was sent (note this is unique for one jvm, this is probably unique across jvms, but not necessarily)		
GET_ATTEMPT_TIME_MILLIS	milliseconds that the message was attempted to be received. If the message is not confirmed in a certain amount of time, the state will be set back to IN_QUEUE to try again		
GET_ATTEMPT_COUNT	how many times this message has been attempted to be retrieved		
STATE	state of this message: IN_QUEUE, GET_ATTEMPTED, PROCESSED		
GET_TIME_MILLIS	millis since 1970 that this message was successfully received		
FROM_MEMBER_ID	member id of user who sent the message		foreign key to grouper_members.id
QUEUE_NAME	queue name for the message that it is delivered to (note, topics can send to multiple queues, which will duplicate the message)		
MESSAGE_BODY	message body		
ATTEMPT_TIME_EXPIRES_MILLIS	millis since 1970 that this attempt will expire before it is acknowledged. After it expires it will be delivered again when receive() is called		

See Also:

[Grouper Messaging System](#)

[Message Format Detail](#)

[Message Format Config Example](#)

[Grouper Messaging System Development Guide](#)