

Infrastructure - Task #8080

Story # 8061 (New): develop queue-based processing system for the CN

ioslate queue creation logic from processing logic from the queue definition logic

2017-04-24 22:50 - Rob Nahf

<b>Status:</b>	In Progress	<b>Start date:</b>	2017-04-27
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Rob Nahf	<b>% Done:</b>	30%
<b>Category:</b>	d1_indexer	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Story Points:</b>	
<b>Milestone:</b>	None		
<b>Product Version:</b>	*		
<b>Description</b>			
Use factory methods to define the queues and channels that will be shared by multiple processors (that represent steps in the pipeline).			
d1_cn_index_common should hold the creation methods (Factories) and the queue name for the starting queue. d1_cn_index_processor should hold the queue names specific to queue processing.			
<b>Subtasks:</b>			
Task # 8086: upgrade Spring dependencies			In Progress

Associated revisions

Revision 18838 - 2017-06-29 20:31 - Rob Nahf

refs #8080: first prototype of d1\_index\_processing using messaging queues. Classes in the new org.dataone.cn.index.messaging hold the configurations and manage the asynchronous consumers, which are for now in o.d.c.index.processor package. Integration test IndexProcessingPrioritizationIT#testConsumerContainerSetup has full branching workflow "test". Imported Hazelcast test configurations; abstracted HttpService behind an interface. More work needed on test configurations - production config hijacked for now until I can get a clean separation.

Revision 18838 - 2017-06-29 20:31 - Rob Nahf

refs #8080: first prototype of d1\_index\_processing using messaging queues. Classes in the new org.dataone.cn.index.messaging hold the configurations and manage the asynchronous consumers, which are for now in o.d.c.index.processor package. Integration test IndexProcessingPrioritizationIT#testConsumerContainerSetup has full branching workflow "test". Imported Hazelcast test configurations; abstracted HttpService behind an interface. More work needed on test configurations - production config hijacked for now until I can get a clean separation.

History

#1 - 2017-04-24 22:51 - Rob Nahf

- Assignee changed from Dave Vieglaiss to Rob Nahf

#2 - 2017-04-26 18:01 - Rob Nahf

if we are going to utilize message queues for other process pipelines, then putting the factories in d1\_cn\_common makes more sense.

#3 - 2017-04-26 21:07 - Rob Nahf

It turns out that Connections and Channels are pretty well encapsulated, so should be passed into Consumers, and maybe Publishers. The actual durable queues can be created through code, or through configuration (or even through the command line. So, queue creation should probably happen in the postinst of dataone-cn-index-processor debian package. (and d1\_cn\_common for the newIndexTask)

<http://stackoverflow.com/questions/18418936/rabbitmq-and-relationship-between-channel-and-connection>

#### #4 - 2017-04-27 21:10 - Rob Nahf

For the d1\_cn\_common parts, Spring already has some pretty nice encapsulations that would both simplify application code (especially in publishing) and being able to push all of the application configuration stuff into xml..

<http://docs.spring.io/spring-amqp/reference/htmlsingle/#compatibility>

<https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-messaging.html>

Spring also has a high-level class called RabbitTemplate:

[http://docs.spring.io/spring-amqp/docs/latest\\_ga/api/org/springframework/amqp/rabbit/core/RabbitTemplate.html](http://docs.spring.io/spring-amqp/docs/latest_ga/api/org/springframework/amqp/rabbit/core/RabbitTemplate.html)

<http://stackoverflow.com/questions/34125116/spring-rabbittemplate-how-to-get-hold-of-the-published-message-for-nacks-in-pub>

using spring would require upgrade of Spring from 3.1.4 to 4.1.x

#### #5 - 2017-05-09 22:55 - Rob Nahf

- Status changed from New to In Progress

- % Done changed from 0 to 30

org.dataone.cn.messaging.QueueAccess is a new class in d1\_cn\_common that gives the user client access to the rabbitMQ queue. It's in d1\_cn\_common because it is more general purpose than indexing, and it only needs the name of a legitimate queue to work.

Indexing configuration will go into d1\_cn\_index\_common (where the IndexTask definitions will be as well).