

Infrastructure - Story #1652

Refactor Metacat MN and CN classes to reflect 0.6.2 API

2011-06-21 17:57 - Chris Jones

<b>Status:</b>	Closed	<b>Start date:</b>	2011-06-21
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Chris Jones	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Sprint-2011.35-Block.5		
<b>Story Points:</b>			
<b>Description</b> <p>At the moment Metacat's MN implementation is handled entirely by CrudService, and the CN implementation thus far is handled by CNCoreImpl. The implementation should be modularized to reflect the MN and CN service levels:</p> <p>From IRC and scratchpad with Ben:</p> <pre>abstract class MNBase {} MNCoreImpl extends MNBase implements MNCore MNReadImpl extends MNBase implements MNRead MNAuthorizationImpl extends MNBase implements MNAuthorization MNStorageImpl extends MNBase implements MNStorage MNReplicationImpl extends MNBase implements MNReplication abstract class CNBase {} CNCoreImpl extends CNBase implements CNCore -partial Metacat implementation CNReadImpl extends CNBase implements CNRead -all Metacat? CNAuthorizationImpl extends CNBase implements CNAuthorization -all Metacat (sysMeta) **CNIdentityImpl extends CNBase implements CNIdentity -implemented outside of Metacat **CNRegisterImpl extends CNBase implements CNRegister -implemented outside Metacat CNReplicationImpl extends CNBase implements CNReplication -proxies to Metacat</pre> <ul style="list-style-type: none"><li>many existing methods in CrudService refactored to MNBase</li><li>sysMeta-related methods in IdentityManager moved to SystemMetadadataManager</li><li>similar CN and MN methods will reuse classes/methods for that shared function (e.g., SystemMetadadataManager.getInstance().getSystemMetadadata()) rather than sharing the root of a class hierarchy for shared methods. We want to keep the distinction between CN and MN stacks very clear at least in the class hierarchy.</li><li>refactor ResourceHandler to minimize duplicate code when handling the REST calls</li><li>Metacat won't implement CNIdentity, CNRegister</li></ul>			
<b>Subtasks:</b>			
Task # 1653: Create Metacat MN impl classes			Closed
Task # 1654: Create Metacat CN impl classes			Closed

History

#1 - 2011-06-21 21:11 - Chris Jones

- Status changed from New to In Progress

After discussions with Matt and Ben, we decided to not break out each API section into classes in favor of two implementing classes and a common super class:

D1NodeService (abstract)  
CNodeService extends D1NodeService implements CNCore, CNRead, CNAuthorization, CNIdentity, CNRegister, CNReplication  
MNodeService extends D1NodeService implements MNCore, MNRead, MNAuthorization, MNStorage, MNReplication

Handling different D1 API versions is an outstanding issue, but will be handled within each of these service implementations.

**#2 - 2011-06-23 14:37 - Chris Jones**

- *Position set to 1*
- *Target version set to Sprint-2011.26-Block.4*

**#3 - 2011-06-23 14:37 - Chris Jones**

- *Position changed from 1 to 209*
- *Target version changed from Sprint-2011.26-Block.4 to Sprint-2011.23-Block.3*
- *Position deleted (1)*
- *Position set to 1*

**#4 - 2011-06-27 01:19 - Dave Vieglais**

- *Position deleted (210)*
- *Target version changed from Sprint-2011.23-Block.3 to Sprint-2011.26-Block.4*
- *Position set to 11*

**#5 - 2011-08-24 15:41 - Chris Jones**

- *Milestone set to None*

Will leave this open until the MNCORE outstanding method responses are defined and the Metacat MNodeService is complete.

**#6 - 2011-08-30 01:27 - Dave Vieglais**

- *Position deleted (30)*
- *Target version deleted (Sprint-2011.26-Block.4)*
- *Position set to 1*

**#7 - 2011-08-30 01:52 - Dave Vieglais**

- *Position deleted (25)*
- *Position set to 3*

**#8 - 2011-08-30 02:36 - Dave Vieglais**

- *Position deleted (11)*
- *Position set to 2*

**#9 - 2011-08-30 02:38 - Dave Vieglais**

- *Position set to 2*
- *Target version set to Sprint-2011.35-Block.5*
- *Position deleted (2)*

**#10 - 2011-09-01 15:33 - Chris Jones**

- *Status changed from In Progress to Closed*

Tasks are fulfilled