

Metacat - Bug #1217

Extend Metacat Interface and Client

11/21/2003 08:39 AM - Duane Costa

<b>Status:</b>	Closed	<b>Start date:</b>	11/21/2003
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Duane Costa	<b>% Done:</b>	0%
<b>Category:</b>	metacat	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.x.y	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	1217		

<b>Description</b> <p>The Metacat interface, implemented by the MetacatClient class, exposes a number of core actions of the MetacatServlet as HTTP requests to the servlet. The current list of supported actions includes:</p> <p>login logout read squery insert update delete</p> <p>A number of additional actions, most of which are not critical but are convenient, are not supported by the interface:</p> <p>query export readinlinedata validate setaccess getaccesscontrol getprincipals getdoctypes gettdtdschema getdataguide getlastdocid getrevisionanddoctype</p> <p>(Note: some of the methods listed above are obsolete and therefore should not be included in the extended interface. The complete list of methods that should be added to the interface is yet to be determined.)</p> <p>Some of the methods in the extended set are useful to other clients, such as Harvester. However, Harvester should interact exclusively with the MetacatClient, rather than directly with the Metacat servlet. Therefore, the current interface should be extended with the additional convenience methods so that they can be used by Harvester and potentially other clients.</p> <p>There are two ways that the current interface could be extended:</p> <p>(1) Add a new interface, 'MetacatExtendedInterface.java', to hold the definitions for the convenience methods. The MetacatClient class would implement both the current Metacat interface and the MetacatExtendedInterface. The advantage of this approach is that the current Metacat interface can be kept simple in that it will only define the core methods.</p> <p>(2) Add the new methods to the existing Metacat interface.</p>
--

The advantage of just adding these methods to Metacat.java is programmatically it's easier because you only have to cast to one interface (not two) to make method calls.

We will determine in the next few days which of these two approaches to use.

A secondary goal of this task is to improve the documentation for the methods in the extended interface.

A future goal (outside the scope of this task) is to refactor Morpho and other code to use the MetacatClient to interact with MetacatServlet.

## History

---

### #1 - 03/13/2008 06:12 PM - Matt Jones

Moving this up in priority, so that we can get around the problem of having all data one one logical volume. Should support a configurable list of filesystem directories for storage.

### #2 - 03/27/2013 02:16 PM - Redmine Admin

Original Bugzilla ID was 1217

### #3 - 04/10/2013 11:33 AM - ben leinfelder

- *Status changed from New to Closed*

Much if not all of this has been in the MetacatClient for a long time now. Since we are deprecating the Metacat API in favor of the DataONE API, I am going to close this because any critical repository features should be addressed in that more general API.