SEEK - Bug #1059

create testing harness for ecogrid

05/19/2003 12:57 PM - Matt Jones

Status:	New	**Start date:**	05/19/2003
Priority:	Immediate	**Due date:**	
Assignee:	Kevin Ruland	**% Done:**	0%
Category:	ecogrid	**Estimated time:**	0.00 hour
Target version:	ecogrid-1.0.0	**Spent time:**	0.00 hour
Bugzilla-Id:	1059		

Description

We need a common testing harness that can be used to launch client queries against ecogrid nodes and validate that the results returned are correct. This system would be a basic client application, and so the core of it could be reused in other clients if it is properly designed.

Who: Higgins
Target: June 9

History

#1 - 11/11/2003 12:42 PM - Matt Jones

Some more detail has emerged on this from the Santa Barbara meeting. The goal of this bug is to create a series of unit tests that test the published EcoGrid APIs for each of the wrapped services (e.g., metacat/srb/DiGIR) to make sure that they all implement the interfaces uniformly. This can be implemented in phases, as the EcoGrip APIs are implemented. So for example, because query APIs are being developed before data insertion APIs, it is ok to assume for now that data being queried is inserted through a non-ecogrid mechanism. Once the write APIs are established, the test should login, insert the test data, query the test data, modify and delete the test data, and test exceptional conditions (such as trying to insert after an invalid login, trying to delete/move/change data owned by another user, etc).

Running these tests should be accomplished through the ant build system by typing "ant test" on the commandline. We have some existing ant build files that run tests, so see me about how to set this up if you have questions. It would be best if the build file had a configurable parameter that lists the implementations to be tested (ie, list a metacat, srb, digir, etc implementation and run through each of those with all of the tests).

Some minimum steps needed to close this bug. It is not a complete list. Individual parts may be broken out into their own individual bugs if that makes it easier to track partial completion of the task.

1) Get test data and queries
a) Obtain representative EML and Darwin Core metadata and data, place in CVS at seek/projects/ecogrid/test/testfiles in a suitably documented way
b) Rewrite and extend the test queries already started in CVS to be a full suite of query test documents that test all of the features of the query language
2) Write a JUnit test suite that is configurable to point at specific implementations of the EcoGridAPIs. This might be tests/org.ecoinformatics.seek.ecogridtest.EcoGridAPITest.java
3) Have a separate function for testing each of the following functions:
a) insert metadata
b) insert data
c) modify metadata
d) modify data
e) query metadata -- get the right set of results (e.g., query method)
f) download data -- using an identifer (e.g., "get" method)
g) query data -- download a subset of data
h) delete data
i) register an ecogrid node
j) query registry for nodes
k) login (authenticate)
l) logout

m) test access control restrictions

Each of these involves several possible EcoGrid API method invocations. Sometimes more than one way to do it will exist (e.g., EcoGrid Query Level I and Level II interfaces) -- we need to test all legitimate approaches, and make sure illegitimate approaches are blocked.

As I said, these will probably not be developed in this order. The first methods to test will probably be the EcoGrid Level I query API, basically the search and get methods, as described in bug 1041.

Developing these tests will have some beneficial side-impacts. First, it will clarify our thinking about the APIs and make sure that they allow the expressiveness we want. Second, it will provide a client library (if properly designed) that can be used in other contexts. Third, it will provide immediate feedback to developers doing wrapping of new services as to whether they have gotten it right or not.

#2 - 11/29/2005 12:04 PM - Matt Jones

This still needs to be done before the release can occur. An additional design goal is to stress test with mutiple simultaneous client connections (probably a configurable number of client connections) to make sure the implementations can handle getting hit with many requests at the same time.

#3 - 03/27/2013 02:16 PM - Redmine Admin

Original Bugzilla ID was 1059