Infrastructure - Story #2645

Session handling in java service api

2012-04-25 21:29 - Rob Nahf

Status:	Closed	**Start date:**	2012-04-26
Priority:	Normal	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	100%
Category:	d1_common_java	**Estimated time:**	0.00 hour
Target version:	Sprint-2012.17-Block.3.1		
Story Points:			

Description

The java service api currently includes a Session parameter for most methods, used to indicate the identity of the client (determines the certificate used for setting up SSL). The behavior is that if null, the certificate in the default location used by CILogon is used.

Since most of the time the default is used, it was proposed that the Session parameter be removed from all d1 service api calls. The portal and integration testing both make use of non-default certificates/sessions, so the ability to set the Session is still a requirement. The secondary proposal is to move the Session parameter into the constructor of the D1Node.

Whatever the solution requires minimal to zero component refactoring, outside of common_java and libclient_java.

Subtasks:

Task # 2647: add corresponding session-less methods to the service api	**Closed**
Task # 2648: add corresponding session-less methods to the client implementations	**Closed**
Task # 2650: add a D1Node constructor that takes a Session object	**Closed**
Task # 2649: update architecture conformity tests	**Closed**

History

#1 - 2012-04-25 21:41 - Rob Nahf

- Category set to d1_common_java

- Assignee set to Rob Nahf

#2 - 2012-04-25 21:56 - Skye Roseboom

exposing a public getter/setter pair on the D1Node would also provide a means of changing the session / subject when needed.

#3 - 2012-04-25 22:30 - Robert Waltz

In the API implementations, the subject is pulled out of the session and used as local variable in business logic of a few methods. The subject will need to be passed in to the functions (having been pulled from the 'session') before calling them.

All authorization code must be pulled out of methods and occur as an aspect of (or orthogonally to) the service container.

#4 - 2012-04-25 22:57 - Rob Nahf

wrt/ set/getSession: It's not the setting/getting of the Session member variable that's an issue.

It's that both the session-less methods and the old methods use null to indicate different desired behaviors regarding what session to use.
for example:

```
public InputStream get(Identifier pid) {
return get((Session) null, pid);
}
```

In this case, you'd want a null Session passed to get(Session, Identifier) to use the getSession() value.

Whereas existing code expects that a null sent to get(Session, Identifier) uses that null value.

If we assume that existing code would not be using setSession(), or D1Node(baseurl, Session) - and thereby also returning null from getSession() - we leave open the possibility for unexpected behavior for the unknowing.

The issue is probably avoided by having the session-less methods pass a special Session object that indicates that the getSession() value be used, and a null Session uses null.

#5 - 2012-05-04 22:49 - Rob Nahf

- Status changed from New to Closed

updated all components, and updates architecture conformity tests in d1_integration to account for the session-less methods compared against the session-containing documentation.