Java Client - Task #7154

Add capability in DataPackage to make statements about anonymous resources in Resource maps

2015-06-04 21:54 - Chris Jones

Status:	Closed	**Start date:**	2015-06-04
Priority:	Normal	**Due date:**	
Assignee:	Chris Jones	**% Done:**	100%
Category:	d1_libclient_java	**Estimated time:**	0.00 hour
Target version:	CCI-2.0.0		
Story Points:			

Description

DataPackage currently provides insertRelationship() a method to provide any relationship using predicates from other namespaces (such as PROV) and URIs as subjects and objects. This doesn't allow for blank (anonymous) nodes. Overload the method to accept any blank nodes as subject and object components of the triple, and fix any tests that use this method. The underlying storage structure (tripleMap), will need to change the types it stores (currently just URIs), likely to a RDFNode subclasses.

Associated revisions

Revision 15736 - 2015-06-05 17:47 - Chris Jones

Fix the arguments for the static resource and property arguments. refs #7154

Revision 15736 - 2015-06-05 17:47 - Chris Jones

Fix the arguments for the static resource and property arguments. refs #7154

Revision 15737 - 2015-06-05 17:49 - Chris Jones

Update the testCreateResourceMapWithProvONE() test to use the new PROV syntax. refs #7154

Revision 15737 - 2015-06-05 17:49 - Chris Jones

Update the testCreateResourceMapWithProvONE() test to use the new PROV syntax. refs #7154

Revision 15738 - 2015-06-05 17:55 - Chris Jones

Fix the arguments for the static resource building (no namespace needed). refs #7154

Revision 15738 - 2015-06-05 17:55 - Chris Jones

Fix the arguments for the static resource building (no namespace needed). refs #7154

Revision 15742 - 2015-06-05 22:43 - Chris Jones

Update the ResourceMapFactory to use the CITO vocabulary and remove the CITO_DOCUMENTS and CITO_IS_DOCUMENTED_BY static predicates. While Ben and Rob and I discussed changing the init() method to a static block, it throws a URIsyntaxException, so I'd need to think about handling that. Leaving it for now. refs #7154

Revision 15742 - 2015-06-05 22:43 - Chris Jones

Update the ResourceMapFactory to use the CITO vocabulary and remove the CITO_DOCUMENTS and CITO_IS_DOCUMENTED_BY static predicates. While Ben and Rob and I discussed changing the init() method to a static block, it throws a URIsyntaxException, so I'd need to think about handling that. Leaving it for now. refs #7154

Revision 15743 - 2015-06-05 22:58 - Chris Jones

Clean up the ProvResourceMapBuilder so it no longer creates Predicates internally, but uses the PROV, ProvONE_V1, and CITO vocabularies. Remove init() (no longer needed), and initialize the rdfModel in the constructor. Move setNamespaces() to the bottom - for some reason I added it above the constructor, which seems odd.

Change the signature of insertRelationship(ResourceMap, URI, Predicate, List) to insertRelationship(ResourceMap, Resource, Property, List) which gives us more flexibility in using more than just Node_URI types in statements.

To support this, update DataPackage to use the new method signature. DataPackage now internally stores the tripleMap as a Map> (instead of URIs and Predicate). We'll then be able to add triple statements with anonymous nodes for the subject, and literals for the object. Stub out overloaded versions of the insertRelationship method to support these use cases. Work in progress. refs #7154

Revision 15743 - 2015-06-05 22:58 - Chris Jones

Clean up the ProvResourceMapBuilder so it no longer creates Predicates internally, but uses the PROV, ProvONE_V1, and CITO vocabularies. Remove init() (no longer needed), and initialize the rdfModel in the constructor. Move setNamespaces() to the bottom - for some reason I added it above the constructor, which seems odd.

Change the signature of insertRelationship(ResourceMap, URI, Predicate, List) to insertRelationship(ResourceMap, Resource, Property, List) which gives us more flexibility in using more than just Node_URI types in statements.

To support this, update DataPackage to use the new method signature. DataPackage now internally stores the tripleMap as a Map> (instead of URIs and Predicate). We'll then be able to add triple statements with anonymous nodes for the subject, and literals for the object. Stub out overloaded versions of the insertRelationship method to support these use cases. Work in progress. refs #7154

Revision 15746 - 2015-06-06 23:01 - Chris Jones

Modify ProvResourceMapBuilder.insertRelationship() to take a list of RDFNode objects as the objects for the triple statements. This will allow us to make statements using literal types, and not just Resources.

Update DataPackage to store triples in the tripleMap using RDFNode objects as well to conform to the change above. Overload insertRelationship() to take a single URI object to accommodate languages that don't support generics typing (like Matlab). Change the overloaded insertRelationship() method that took a String object to now be an Object literal. This is unimplemented yet - work in progress.
refs #7154

Revision 15746 - 2015-06-06 23:01 - Chris Jones

Modify ProvResourceMapBuilder.insertRelationship() to take a list of RDFNode objects as the objects for the triple statements. This will allow us to make statements using literal types, and not just Resources.

Update DataPackage to store triples in the tripleMap using RDFNode objects as well to conform to the change above. Overload insertRelationship() to take a single URI object to accommodate languages that don't support generics typing (like Matlab). Change the overloaded insertRelationship() method that took a String object to now be an Object literal. This is unimplemented yet - work in progress.
refs #7154

Revision 15754 - 2015-06-09 22:49 - Chris Jones

Fix minor copy/paste typos. refs #7154

Revision 15754 - 2015-06-09 22:49 - Chris Jones

Fix minor copy/paste typos. refs #7154

Revision 15756 - 2015-06-09 23:04 - Chris Jones

dd vocabulary classes for the Dublin Core Metadata Terms. refs #7154

Revision 15756 - 2015-06-09 23:04 - Chris Jones

dd vocabulary classes for the Dublin Core Metadata Terms. refs #7154

Revision 15762 - 2015-06-11 21:38 - Chris Jones

Overload RsourceMapFactory.createResourceMap() to allow for the creation of an unpopulated base resource map, and allow for a custom title. This will allow us to move away from the tripleMap structure in DataPackage that is too limiting when trying to insert non-aggregation related statements.

refs #7154

Revision 15762 - 2015-06-11 21:38 - Chris Jones

Overload RsourceMapFactory.createResourceMap() to allow for the creation of an unpopulated base resource map, and allow for a custom title. This will allow us to move away from the tripleMap structure in DataPackage that is too limiting when trying to insert non-aggregation related statements.

refs #7154

Revision 15763 - 2015-06-11 21:43 - Chris Jones

When calling ProvrresourceMapBuilder.insertRelationship(), ensure that the statements being made do not result in orphaned nodes in the graph. Also, change the overloaded insertrelationship() method used for inserting blank node subjects to call insertRelationship(Resource, Property, RDFNode) to reduce code duplication.

refs #7154

Revision 15763 - 2015-06-11 21:43 - Chris Jones

When calling ProvrresourceMapBuilder.insertRelationship(), ensure that the statements being made do not result in orphaned nodes in the graph. Also, change the overloaded insertrelationship() method used for inserting blank node subjects to call insertRelationship(Resource, Property, RDFNode) to reduce code duplication.

refs #7154

Revision 15764 - 2015-06-11 22:04 - Chris Jones

This is a significant change in DataPackage to redesign how we store the triple statements. After having a design discussion with Ben regarding the limitations of using a Map>> structure, change the internal storage to be the ResourceMap itself. This removes the need of having 1) a metadataMap, and 2) a tripleMap. The tripleMap itself was the most problematic, in that it was impossible to ensure insertion order when creating a resource map from the LinkedHashMap because of repeating Predicate keys. Both maps are replaced by the resourceMap, which is basically just a list of RDF statements anyway.

In support of this, change the constructor to create a base resource map from ResourceMapFactory, and modify insertRelationship(Identifier, List) to build the ResourceMap instance up rather than the metadataMap. It now aggregates resources into the map, and describes them with other calls to insertRelationship(). Add a private aggregate() method to help with this. change getMap() to getResourceMap(), and change setMetadataMap() to setResourceMap(). Keep getMetadataMap() since it's helpful in serializing/deserializing using ResourceMapFactory. Change getDocumentedBy() to search the resource map rather than the old metadata map whaen returning a metadata identifier.

lastly, finish out the version of insertRelationship() that allows for statements with literals as the object.

refs #7154

Revision 15764 - 2015-06-11 22:04 - Chris Jones

This is a significant change in DataPackage to redesign how we store the triple statements. After having a design discussion with Ben regarding the limitations of using a Map>> structure, change the internal storage to be the ResourceMap itself. This removes the need of having 1) a metadataMap, and 2) a tripleMap. The tripleMap itself was the most problematic, in that it was impossible to ensure insertion order when creating a resource map from the LinkedHashMap because of repeating Predicate keys. Both maps are replaced by the resourceMap, which is basically just a list of RDF statements anyway.

In support of this, change the constructor to create a base resource map from ResourceMapFactory, and modify insertRelationship(Identifier, List) to build the ResourceMap instance up rather than the metadataMap. It now aggregates resources into the map, and describes them with other calls to insertRelationship(). Add a private aggregate() method to help with this. change getMap() to getResourceMap(), and change setMetadataMap() to setResourceMap(). Keep getMetadataMap() since it's helpful in serializing/deserializing using ResourceMapFactory. Change getDocumentedBy() to search the resource map rather than the old metadata map whaen returning a metadata identifier.

lastly, finish out the version of insertRelationship() that allows for statements with literals as the object.

refs #7154

Revision 15765 - 2015-06-11 22:29 - Chris Jones

Add Andrei's changes to setting the D1Client.CN_URL. See r15758 of ResourceMapFactory. refs #7154

Revision 15765 - 2015-06-11 22:29 - Chris Jones

Add Andrei's changes to setting the D1Client.CN_URL. See r15758 of ResourceMapFactory. refs #7154

Revision 15766 - 2015-06-11 22:58 - Chris Jones

Overload DataPackage.insertRelationship() to allow for a blank node id string as the subject of a statement. This calls ProvresourceMapBuilder.insertRelationship() to do the same, and add the blank node to the resource map graph.

refs #7154

Revision 15766 - 2015-06-11 22:58 - Chris Jones

Overload DataPackage.insertRelationship() to allow for a blank node id string as the subject of a statement. This calls ProvresourceMapBuilder.insertRelationship() to do the same, and add the blank node to the resource map graph.

refs #7154

Revision 15787 - 2015-06-16 16:36 - Chris Jones

Fix the insertRelationship() method that allows for blank node creation. To avoid inserting multiple blank nodes for each property being added, track blank nodes with the DC_TERMS.identifier property. This allows us to identify blank nodes across serializations of the resource map, since anonymous node identifiers themselves do not persist from one model to the next 9like from Foresite to Jena) models. refs #7154

Revision 15787 - 2015-06-16 16:36 - Chris Jones

Fix the insertRelationship() method that allows for blank node creation. To avoid inserting multiple blank nodes for each property being added, track blank nodes with the DC_TERMS.identifier property. This allows us to identify blank nodes across serializations of the resource map, since

anonymous node identifiers themselves do not persist from one model to the next 9like from Foresite to Jena) models. refs #7154

Revision 15788 - 2015-06-16 16:38 - Chris Jones

Add the testInsertRelationshipBlankNode() test to exercise blank node creation. Add asPredicate() to help with this, as well as getsimpleSelector. The latter two methods could potentially be refactored into a super class. refs #7154

Revision 15788 - 2015-06-16 16:38 - Chris Jones

Add the testInsertRelationshipBlankNode() test to exercise blank node creation. Add asPredicate() to help with this, as well as getsimpleSelector. The latter two methods could potentially be refactored into a super class. refs #7154

Revision 15800 - 2015-06-17 20:29 - Chris Jones

Add the testInsertRelationshipObjectLiteral() test that exercises the insertRelationship() method using a literal type as the object of the RDF statement. refs and closes #7154

Revision 15800 - 2015-06-17 20:29 - Chris Jones

Add the testInsertRelationshipObjectLiteral() test that exercises the insertRelationship() method using a literal type as the object of the RDF statement. refs and closes #7154

History

#1 - 2015-06-17 20:41 - Chris Jones

- Status changed from New to Closed

- % Done changed from 0 to 100

- translation missing: en.field_remaining_hours set to 0.0

DataPackage.insertRelationship is now overloaded with a number of method options for adding triple statements. Note that some of the overloaded methods purposefully don't use Java generics, but rather plain classes, to support languages like Matlab that don't support generics.