

Java Client - Bug #6747

TypeMarshaller.convertTypeFromType can't convert the v2.Log object to v1.Log object

2015-01-07 22:32 - Jing Tao

Status: Closed	Start date: 2015-01-07
Priority: Normal	Due date:
Assignee: Rob Nahf	% Done: 100%
Category: d1_common_java	Estimated time: 0.00 hour
Target version: CLJ-2.0.0	
Story Points:	
Description	
<p>In the getLogRecords method, we use TypeMarshaller.convertTypeFromType to convert a v2.Log object to v1.Log object. However I got an error: metacat 20150107-12:52:58: [ERROR]: class java.lang.ClassCastException: org.dataone.service.types.v2.LogEntry cannot be cast to org.dataone.service.types.v1.LogEntry [edu.ucsb.nceas.metacat.restservice.D1ResourceHandler] java.lang.ClassCastException: org.dataone.service.types.v2.LogEntry cannot be cast to org.dataone.service.types.v1.LogEntry at org.dataone.service.JiBX_bindingMungeAdapter.JiBX_binding_marshall_1_31() at org.dataone.service.JiBX_bindingMungeAdapter.JiBX_binding_marshall_1_32() at org.dataone.service.JiBX_bindingLog_access.marshall() at org.dataone.service.types.v1.Log.marshall(Log.java) at org.jibx.runtime.impl.MarshallingContext.marshallRoot(MarshallingContext.java:1021) at org.jibx.runtime.impl.MarshallingContext.marshallDocument(MarshallingContext.java:1041) at org.dataone.service.util.TypeMarshaller.marshallTypeToOutputStream(TypeMarshaller.java:99) at org.dataone.service.util.TypeMarshaller.marshallTypeToOutputStream(TypeMarshaller.java:78) at edu.ucsb.nceas.metacat.restservice.v1.MNResourceHandler.getLog(MNResourceHandler.java:1056) at edu.ucsb.nceas.metacat.restservice.v1.MNResourceHandler.handle(MNResourceHandler.java:279)</p> <p>I looked the Log classes and found that the v2.Log contains a list of v2.LogEntry objects and v1.log contains a list of v1.LogEntry objects. So I changed the code to convert the v2.LogEntry object to v1.LogEntry object rather than convert the v2.Log to v1.Log directly. Here is the code: for(int i=0; i<log.getCount(); i++) { org.dataone.service.types.v2.LogEntry v2Entry = log.getLogEntry(i); LogEntry v1Entry = TypeMarshaller.convertTypeFromType(v2Entry, LogEntry.class); retLog.addLogEntry(v1Entry); } }</p> <p>However it still doesn't work: Caused by: java.lang.IllegalArgumentException: Cannot invoke org.dataone.service.types.v1.LogEntry.setEvent - argument type mismatch at org.apache.commons.beanutils.PropertyUtilsBean.invokeMethod(PropertyUtilsBean.java:1778) at org.apache.commons.beanutils.PropertyUtilsBean.setSimpleProperty(PropertyUtilsBean.java:1759) at org.apache.commons.beanutils.BeanUtilsBean.copyProperty(BeanUtilsBean.java:447) at org.apache.commons.beanutils.BeanUtilsBean.copyProperties(BeanUtilsBean.java:261) at org.apache.commons.beanutils.BeanUtils.copyProperties(BeanUtils.java:114) at org.dataone.service.util.TypeMarshaller.convertTypeFromType(TypeMarshaller.java:219) at edu.ucsb.nceas.metacat.dataone.v1.MNNodeService.getLogRecords(MNNodeService.java:454) ... 28 more</p> <p>In v2.LogEntry, the parameter of the setEvent is the Event class; in v1.LogEntry class, the parameter of the setEvent is the String class. So they don't match.</p>	
Subtasks:	
Task # 6748: Create a factory / builder class for converting versioned types.	New
Related issues:	
Related to Infrastructure - Bug #6744: Can't access the getLogRecord through ...	Closed 2015-01-07 2015-01-13

Associated revisions

Revision 16256 - 2015-09-05 00:18 - Rob Nahf

fixes #6747. Refactored convertTypeFromType to not produce invalid v1.LogEntries or v1.Log due to incompatible event values.

Revision 16256 - 2015-09-05 00:18 - Rob Nahf

fixes #6747. Refactored convertTypeFromType to not produce invalid v1.LogEntries or v1.Log due to incompatible event values.

History

#1 - 2015-01-07 22:35 - Jing Tao

If we add the new method setEvent with the String parameter, will the issue be fixed?

#2 - 2015-01-07 23:15 - Rob Nahf

- Tracker changed from Task to Bug
- translation missing: en.field_remaining_hours set to 0.0
- Start date deleted (2015-01-07)
- Assignee set to Rob Nahf
- Category set to d1_common_java

The convertTypeFromType method doesn't seem to universally work. (probably only works for cases where one type is derived from another and has extra fields, but not sure).

Propose to make a Factory / Builder class in d1_common_java for specific cases.

#3 - 2015-01-07 23:15 - Rob Nahf

- Target version set to CCI-2.0.0
- Start date set to 2014-12-01
- Due date set to 2015-01-07

#4 - 2015-01-08 17:01 - Ben Leinfelder

- Due date changed from 2015-01-07 to 2015-01-08

Please check our the BeanUtils Converter to see if this can be leveraged before developing a roll-your-own solution.

<http://commons.apache.org/proper/commons-beanutils/apidocs/org/apache/commons/beanutils/Converter.html>

#5 - 2015-01-23 00:28 - Jing Tao

Something need to be clarified : the V2 LogEntry class has the setEvent(String event) method; the v1 LogEntry class has the setEvent(Event event) method.

I tried a wrapper solution - create a setEvent(String event) method in V1LogEntryWrapper class:

```
public static class V1LogEntryWrapper extends org.dataone.service.types.v1.LogEntry{
```

```
    public void setEvent(String event) {
        if(event != null) {
            super.setEvent(Event.convert(event));
        }
    }
}
```

Try to convert the V2 LogEntry object to the V1LogEntryWrapper object first, then cast the V1LogEntryWrapper to the v1 LogEntry:

```
public org.dataone.service.types.v1.LogEntry convert(LogEntry logEntryV2) throws InstantiationException, IllegalAccessException,
InvocationTargetException, JiBXException, IOException {
    org.dataone.service.types.v1.LogEntry logEntryV1 = null;;
    if(logEntryV2 != null) {
```

```
        //convert the v2 LogEntry object to the V1LogEntryWrapper object
        V1LogEntryWrapper wrapper = TypeMarshaller.convertTypeFromType(logEntryV2, V1LogEntryWrapper.class
```

```
);  
    //convert the V2LogEntryWrapper object to the v1 LogEntry object  
    logEntryV1 = (org.dataone.service.types.v1.LogEntry) wrapper;  
    }  
    return logEntryV1;}
```

However it still doesn't work:

Cannot invoke org.dataone.service.types.v1.LogEntry.setEvent - argument type mismatch

I am going to use the getter and setter methods to convert it.

#6 - 2015-01-23 01:51 - Jing Tao

- Status changed from New to In Progress

Using the getter and setter works.

#7 - 2015-02-11 19:47 - Rob Nahf

- Project changed from Infrastructure to Java Client

- Category deleted (d1_common_java)

- Target version changed from CCI-2.0.0 to CLJ-2.0.0

#8 - 2015-08-10 20:21 - Rob Nahf

Looking into the method, the errors have to do with BeanUtils.copyProperties, which assumes that setters with the same name correspond to the same type. We should still be able to use other BeanUtils methods (describe, in combination with copyProperty) and some targeted introspection to avoid type-mismatch exceptions.

Because the deep-copy preparatory step is the step related to Marshalling, this method should be moved to a dedicated builder or factory classes.

A single method for all conversions (as exists now) is seemingly more future-proof with regards to the API, but implementation might be easier with dedicated methods that know the idiosyncrasies of each conversion, such as:

```
public static convertSystemMetadata(Object original, Class targetClass);  
public static convertLog(Object original, Class targetClass);  
public static convertNodeList(Object original, Class targetClass);
```

etc.

Although it's tempting at this point to have version-to-version specific converter methods, it would lead to a multiplication of methods upon v3 and v4, needing v!/2 methods for each evolving type.

The shortcomings of BeanUtils.describe, is that it cannot know the class of the setter's parameter before it is set.

#9 - 2015-09-05 00:27 - Rob Nahf

- *Status changed from In Progress to Testing*
- *% Done changed from 30 to 50*
- *Category set to d1_common_java*

rewrote convertTypeFromType to handle all v2 -v1 conversions. The initial solution left incompatible LogEntries (where the v2.Event string doesn't map to a v1.Event Enum element) in the Log. However, we determined that this was not schema valid, so when serialized, they could not be deserialized.

The solution here is to omit the entire incompatible LogEntry. Added special handling to remove incompatibilities before returning the v1.Log or v1.LogEntry.

#10 - 2015-09-05 01:07 - Rob Nahf

- *% Done changed from 50 to 100*
- *Status changed from Testing to Closed*

Applied in changeset [d1client|r16256](#).