

git_comments:

1. * * A class for marshaling @link{AbstractId} so REST calls can serialize AbstractId to its canonical value.
2. should not unmarshal from String
3. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.

git_commits:

1. **summary:** ACCUMULO-4755 Custom serialization for AbstractId types (#336)
message: ACCUMULO-4755 Custom serialization for AbstractId types (#336) Custom JAXB marshaling with XmlJavaTypeAdapter

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ACCUMULO-4755 Custom serialization for AbstractId types
body: Since the monitor is using the annotation `@Produces(MediaType.APPLICATION_JSON)` it looks like Jersey is taking the JAXB route by default. This was causing my custom Jackson serializer to not work on the front-end. Switched over to customizing the marshaling function with a custom XmlAdapter `JaxbAbstractIdSerializer`

github_pulls_comments:

1. Can one of the admins verify this patch?
2. Squashed and merged. I confirmed that jaxb-api is brought in from Hadoop's class path.

github_pulls_reviews:

1. The version should go in the parent POM's dependencyManagement section. The child POMs should omit the version.
2. Why make this non-final? Given how important this method is for its serialization/internal representation, it's pretty important that it not be overridden.
3. UnsupportedOperationException might be more appropriate here.
4. I don't think this needs to change.
5. I don't think the serialVersionUID needs to change.
6. Should guard against NPE here. Serialization shouldn't break if ID is null.
7. Definitely an oversight on my part, fixed.
8. Ah, good. I was worried that this was required for the serialization to work for some reason.
9. Too bad `@XmlJavaTypeAdapter` is not marked with `@Inherited`. Then, you could just annotate `AbstractId` instead of both `Table.ID` and `Namespace.ID`. It may be possible to register the serializer directly, using a method call in the monitor code instead of annotations. This would also save from adding the compile-time dependency to the core module. But, this is almost certainly not worth it.

jira_issues:

jira_issues_comments: