

**git\_comments:**

1. START OF GENERATED CODE
2. 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.
3. END OF GENERATED CODE
4. `** 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.`
5. `** Generated by camel-package-maven-plugin - do not edit this file!`
6. `** Used for JMS users to allow the JMSType header from the JMS spec to * specify a FQN classname to use to unmarshal to.`
7. `** Adds permissions that controls which Java packages and classes XStream is * allowed to use during unmarshal from xml/json to Java beans. A permission * must be configured either here or globally using a JVM system property. * The permission can be specified in a syntax where a plus sign is allow * and minus sign is deny. Wildcards is supported by using . as prefix. For * example to allow com.foo and all subpackages then specify com.foo.. * Multiple permissions can be configured separated by comma such as * com.foo.-com.foo.bar.MySecretBean. The following default permission is * always included: - java.lang.java.util. unless its overridden by * specifying a JVM system property with they key * org.apache.camel.xstream.permissions.`
8. `** To use custom Jackson modules com.fasterxml.jackson.databind.Module * specified as a String with FQN class names. Multiple classes can be * separated by comma.`
9. `** If you want to marshal a pojo to JSON and the pojo has some fields with * null values. And you want to skip these null values you can set this * option to NOT_NULL`
10. `** Set of features to disable on the Jackson * com.fasterxml.jackson.databind.ObjectMapper. The features should be a * name that matches a enum from * com.fasterxml.jackson.databind.SerializationFeature * com.fasterxml.jackson.databind.DeserializationFeature or * com.fasterxml.jackson.databind.MapperFeature Multiple features can be * separated by comma`
11. `** Camel Fastjson support * * Generated by camel-package-maven-plugin - do not edit this file!`
12. `** Which json library to use.`
13. `** Class name of the java type to use when unarmshalling`
14. `** Refers to a custom collection type to lookup in the registry to use. This * option should rarely be used but allows to use different collection types * than java.util.Collection based as default.`
15. `** When marshalling a POJO to JSON you might want to exclude certain fields * from the JSON output. With Jackson you can use JSON views to accomplish * this. This option is to refer to the class which has JsonView annotations`
16. `** Whether to enable the JAXB annotations module when using jackson. When * enabled then JAXB annotations can be used by Jackson.`
17. `** To enable pretty printing output nicely formatted. Is by default false.`
18. `** 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.
19. \*\* Whether the data format should set the Content-Type header with the type \* from the data format if the data format is capable of doing so. For \* example application/xml for data formats marshalling to XML or \* application/json for data formats marshalling to JSon etc.
  20. \*\* To use custom Jackson modules referred from the Camel registry. Multiple \* modules can be separated by comma.
  21. \*\* If enabled then Jackson is allowed to attempt to use the \* CamelJacksonUnmarshalType header during the unmarshalling. This should \* only be enabled when desired to be used.
  22. \*\* To unarmshal to a List of Map or a List of Pojo.
  23. \*\* Lookup and use the existing ObjectMapper with the given id when using \* Jackson.
  24. \*\* Set of features to enable on the Jackson \* com.fasterxml.jackson.databind.ObjectMapper. The features should be a \* name that matches a enum from \* com.fasterxml.jackson.databind.SerializationFeature \* com.fasterxml.jackson.databind.DeserializationFeature or \* com.fasterxml.jackson.databind.MapperFeature Multiple features can be \* separated by comma

#### git\_commits:

1. **summary:** Regen. This closes #1743  
**message:** Regen. This closes #1743

#### github\_issues:

#### github\_issues\_comments:

#### github\_pulls:

1. **title:** add a new json dataformat for fastjson library  
**body:** add a new json dataformat for fastjson library: <https://github.com/alibaba/fastjson>

#### github\_pulls\_comments:

1. Thanks, do you mind fix checkstyle issues mvn compile -P sourcecheck
2. fixed now. thx. @davsclaus @onders86
3. This looks good anyway +1
4. New data format API is cool contribution. thanks. One last thing may be asking you to squash your commits?
5. I'm a bit concerned about the license first line: <https://github.com/alibaba/fastjson/blob/master/license.txt>
6. Licensed under the Apache License, Version 2.0 (the "License"); @oscerd
7. I was looking at the copyright Copyright 1999-2016 Alibaba Group Holding Ltd. I guess it's not a problem, but maybe @davsclaus can answer better than me.
8. @davsclaus i fixed it
9. @davsclaus does this copyright in ref library matter? thx. @onders86 how to squash commits? can you give me some tips, thx.
10. No the (c) is not a problem, its fine. There are many ASL2 projects that are licensed by a company, for example Hazelcast is also ASL2 license but (c) from that company <https://github.com/hazelcast/hazelcast/blob/master/hazelcast-client/src/main/java/com/hazelcast/client/ClientExtension.java#L2>
11. Its okay about squashing, sadly we don't have a nice way of doing this from the github website, which you can do in some projects. We can merge this as-is. As if you have not do any squash before it can be a bit danger to do, if you don't do it correct. I have a alias for my git client so I can squash X commits back. I found it somewhere on the internet `` squash => !f(){ git reset --soft HEAD~\${1} && git commit --edit -m"\${git log --format=%B --reverse HEAD..HEAD@{1}}"; };f `` @kimmking next contribution, then remember to create a JIRA ticket and refer to the CAMEL-NNN number of the JIRA in the commit messages.
12. @davsclaus thx so much. I have squashed commits now. And thanks a lot to every nice guy here.
13. @kimmking <https://github.com/edx/edx-platform/wiki/How-to-Rebase-a-Pull-Request#squash-your-changes>
14. @wuwen5 thx too much.
15. Thanks for the contribution

### **github\_pulls\_reviews:**

1. More NPE safe as below i would say ``` Message message = null; if (exchange.hasOut()) { message = exchange.getOut(); } else { message = exchange.getIn(); } if (message != null) { message.setHeader(Exchange.CONTENT\_TYPE, "application/json"); message.setHeader(Exchange.CONTENT\_LENGTH, len); } ```
2. Just a note that exchange is never null, also that getIn or getOut also is never null (they lazy create a new empty message)
3. We tend to flip this so != null is last
4. Please set version as variable inside camel-parent/pom.xml
5. fixed it, thx @dmvolod

### **jira\_issues:**

### **jira\_issues\_comments:**