

XML to JSON camel message conversion

Code example attached to this page demonstrates how we can convert xml request message into json response

Step-by-step guide

How to import and run project:

1. Import project through eclipse import maven project functionality
2. Create maven run configuration with "clean install camel:run" goal
3. Run configuration
4. For debug purpose you can create maven debug configuration with goal:
clean install camel:run -Dexec.args="-classpath %classpath -Xdebug
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=9090"
5. You can check the rest api with postman on following rest address: <http://localhost:9090/orderFulfillment/processXmlOrders>
XML body:

```
<Order xmlns="http://www.pluralsight.com/orderfulfillment/Order">
  <OrderType>
    <FirstName>Larry</FirstName>
    <LastName>Horse</LastName>
    <Email>larry@hello.com</Email>
    <OrderNumber>1006</OrderNumber>
    <TimeOrderPlaced>2016-08-16T10:10:28.262+02:00</TimeOrderPlaced>
    <FulfillmentCenter>ABCFulfillmentCenter</FulfillmentCenter>
    <OrderItems>
      <ItemNumber>329918349</ItemNumber>
      <Price>2.50000</Price>
      <Quantity>1</Quantity>
    </OrderItems>
  </OrderType>
</Order>
```

6. Header: Content-Type=application/xml

Route Builder with automatic conversion

```
from("cxfrs://http://localhost:9090?resourceClasses=com.rest.service.Order  
Service&bindingStyle=Default").log("BEFORE (XML) = ${body}")  
    .marshal("xmlToJson").log("BEFORE (JSON) = ${body}");
```

where xmlToJson:

```
@Bean  
public XmlJsonDataFormat xmlToJson() {  
    List<XmlJsonDataFormat.NamespacesPerElementMapping> namespaces = new  
    ArrayList<XmlJsonDataFormat.NamespacesPerElementMapping>();  
    namespaces.add(new XmlJsonDataFormat.NamespacesPerElementMapping("u",  
    "http://www.pluralsight.com/orderfulfillment/Order"));  
    XmlJsonDataFormat xmlJsonDataFormat = new XmlJsonDataFormat();  
    xmlJsonDataFormat.setEncoding("UTF-8");  
    xmlJsonDataFormat.setForceTopLevelObject(true);  
    xmlJsonDataFormat.setTrimSpaces(true);  
    xmlJsonDataFormat.setRootName("Order");  
    xmlJsonDataFormat.setSkipNamespaces(true);  
    xmlJsonDataFormat.setRemoveNamespacePrefixes(true);  
    xmlJsonDataFormat.setExpandableProperties(Arrays.asList("d", "u"));  
    xmlJsonDataFormat.setNamespaceMappings(namespaces);  
  
    return xmlJsonDataFormat;  
}
```

Route Builder with manual conversion

```
@Override  
public void configure() throws Exception {  
  
    from("cxfrs://http://localhost:9090?resourceClasses=com.rest.service.Order  
Service&bindingStyle=SimpleConsumer").beanRef(  
        "fulfillmentCenterOneProcessor",  
        "transformToOrderRequestMessage").log("My HttpResponse = ${body}");  
    }  
  
    where beanid = fulfillmentCenterOneProcessor and method name =  
    transformToOrderRequestMessage
```

GitHub Repository link:

<https://github.com/ionc21/camel-cxfrs-jdbc-rest>