# **From this Lab exercise you be able to understand:**

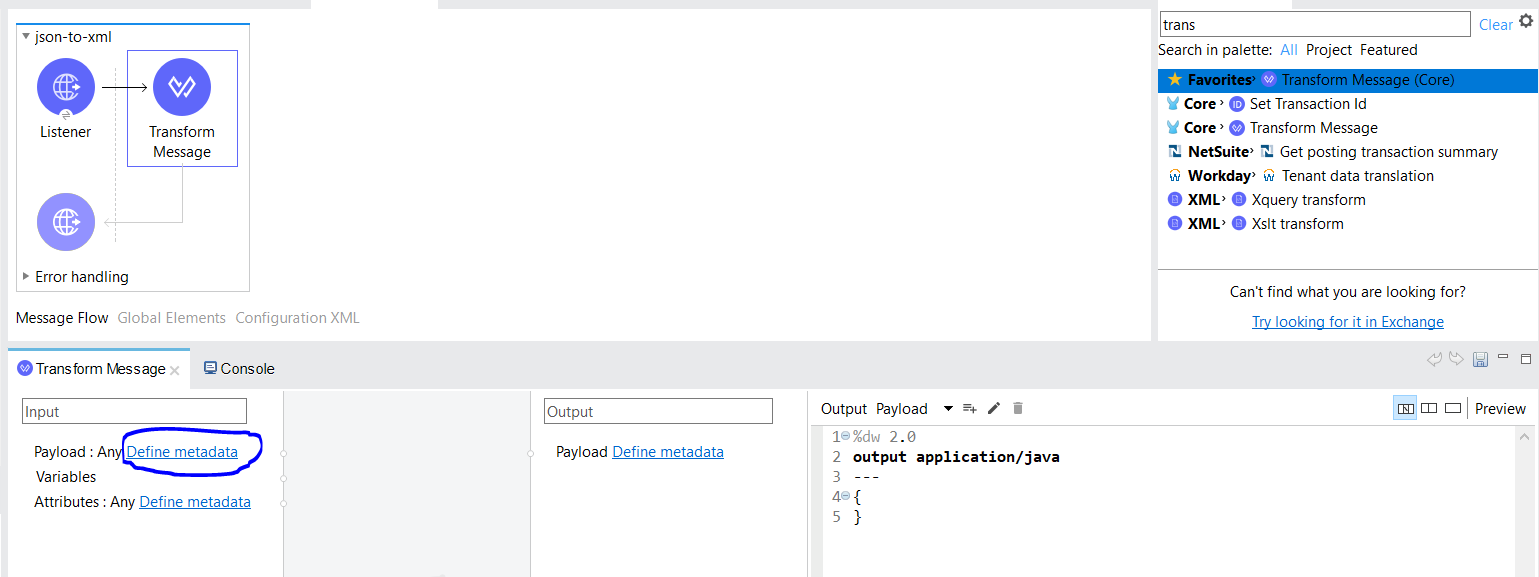
* transform json to xml
* transform xml to json
* transform csv to json
* transform flat file to xml

# Pre-request

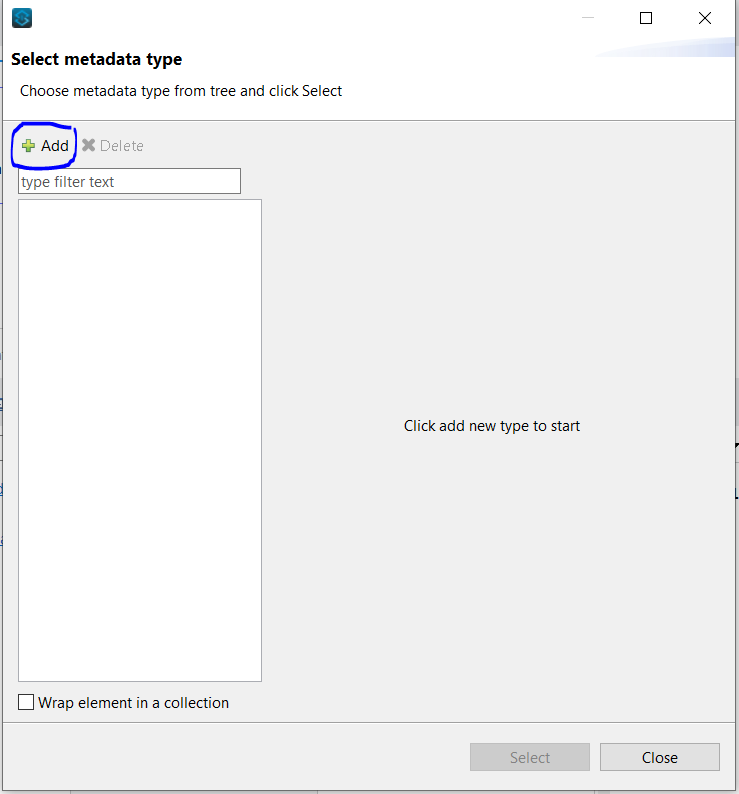
Create sample files of the expected input and output messages. Remember to save these files to disk so MuleSoft can reference them to create scaffoldings of input and output messages in the Transform Message component. Create json, xml, csv and flat file

# Transform json to xml

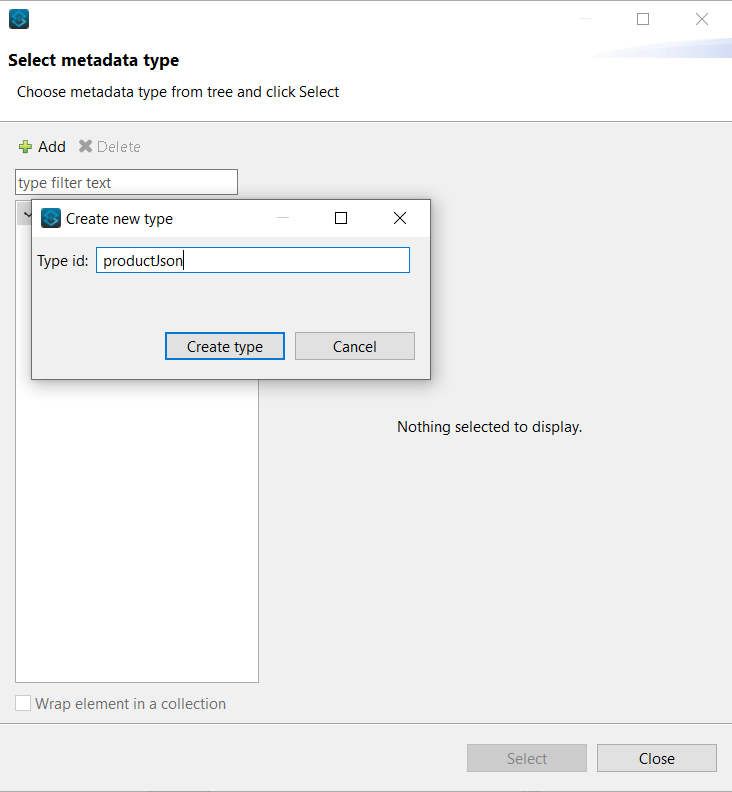
1. Set http listener and transformer from mule palette as shown below
2. Set http connector configuration with localhost:8081
3. Set listener path to /jsonToXml to trigger from postman
4. On the transformer define json metadata
5. Click define metadata link in the Input section of the Transform Message component.



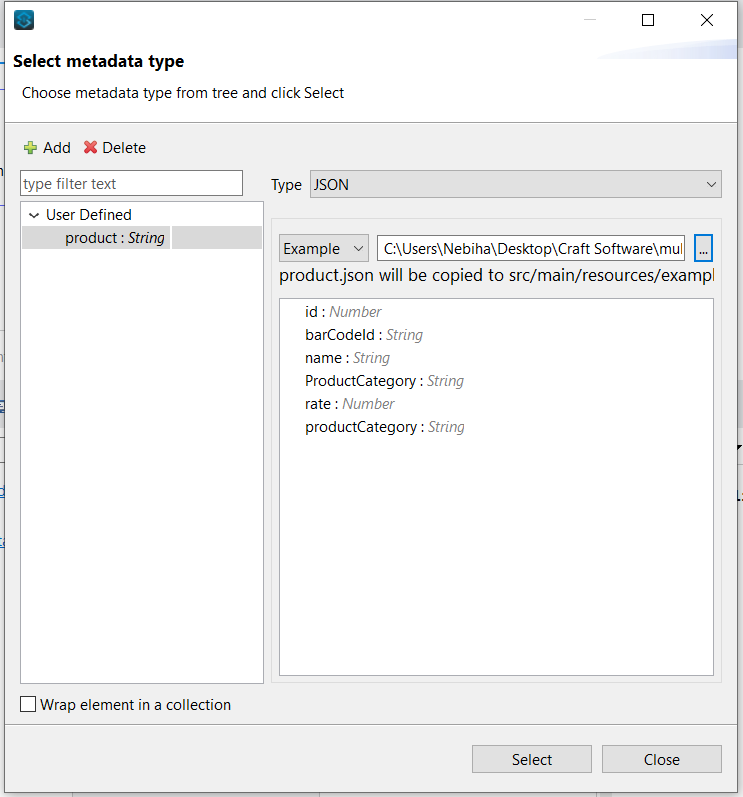
1. New window will pop up so, click Add button on the emerging window



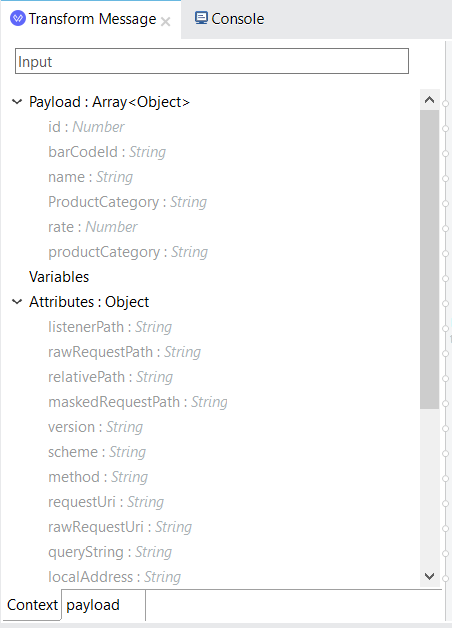
1. Type productJson
2. Click Create type button  This creates a global object definition in the project that can be reused to set metadata for other transformations



1. Select the option json
2. For the schema select example
3. Select the json file from local directory



1. Click Select button
2. The json payload and datatype will appear on the left side bar



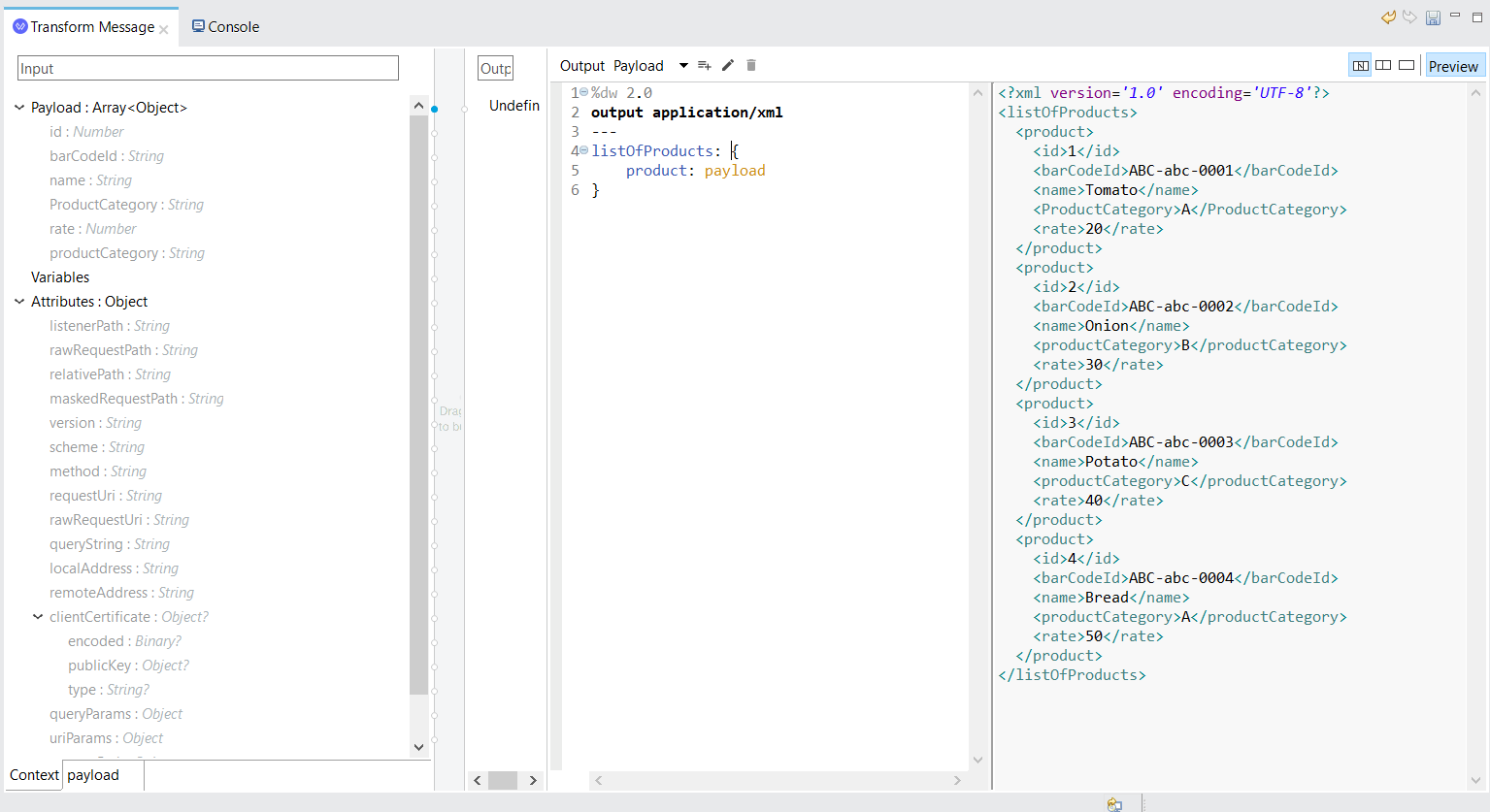
1. Change application/java to application/xml
2. Write script inside the body

*listProduct: {*

*product: payload*

*}*

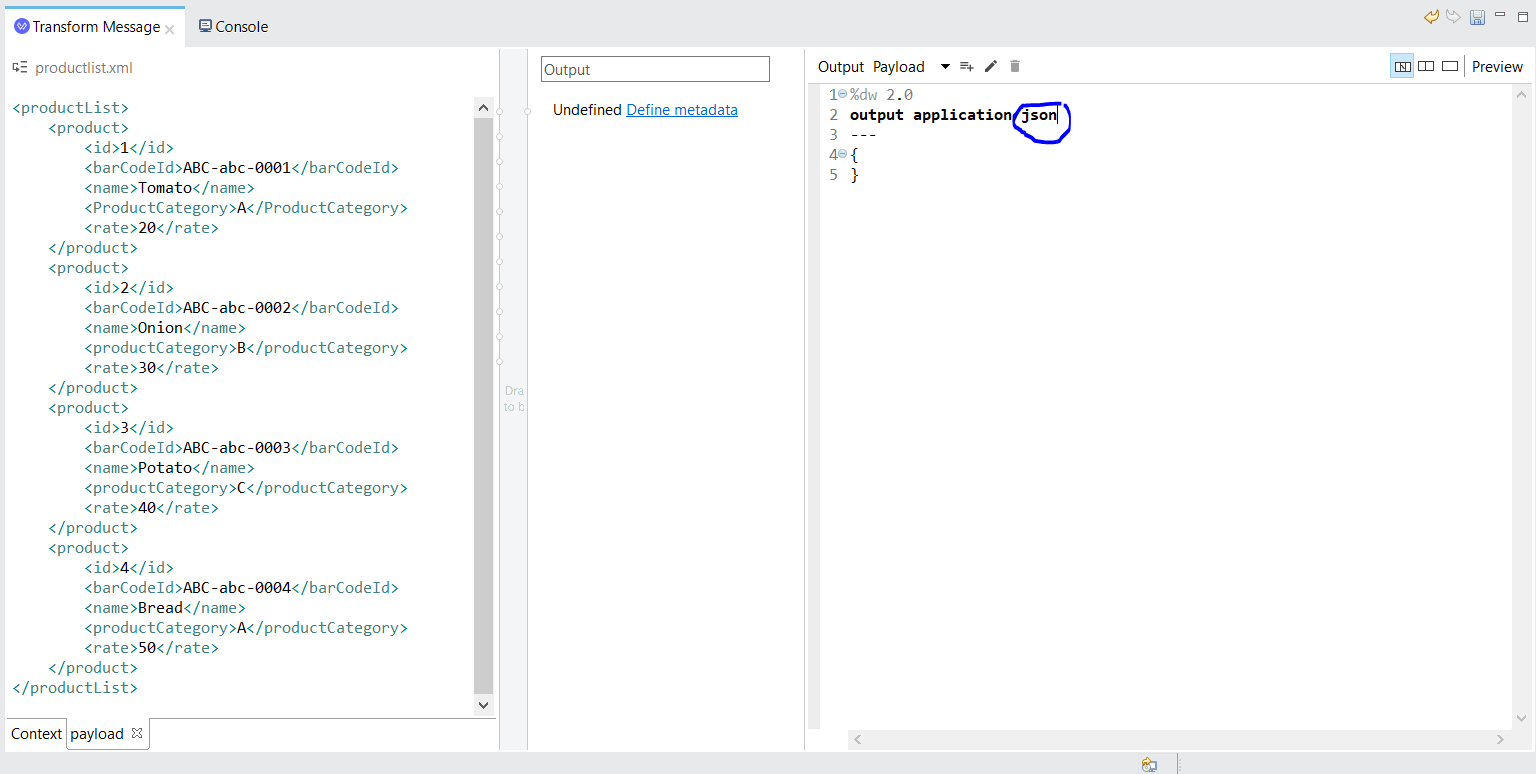
1. Click preview window on the right corner



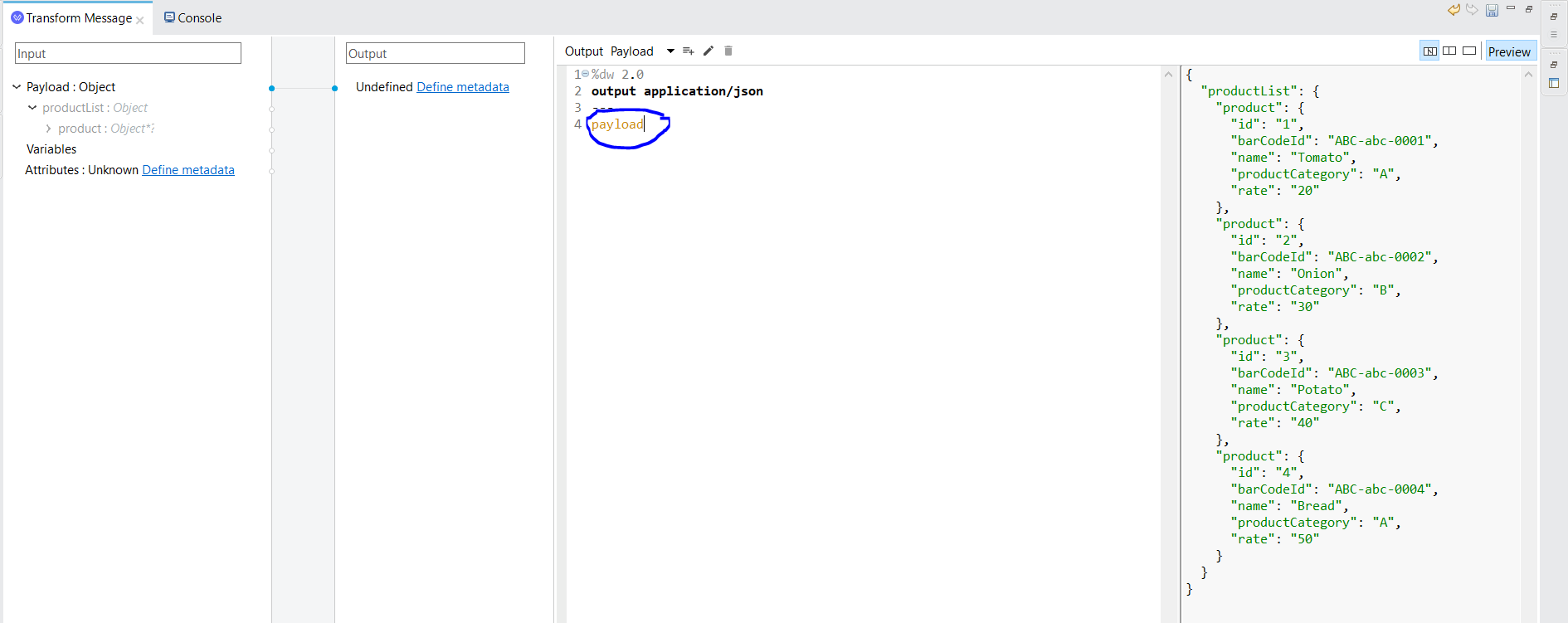
**Note:** The **preview** window will let us to see the output or the transformed message without triggering the flow from external application

**B. xml to json**

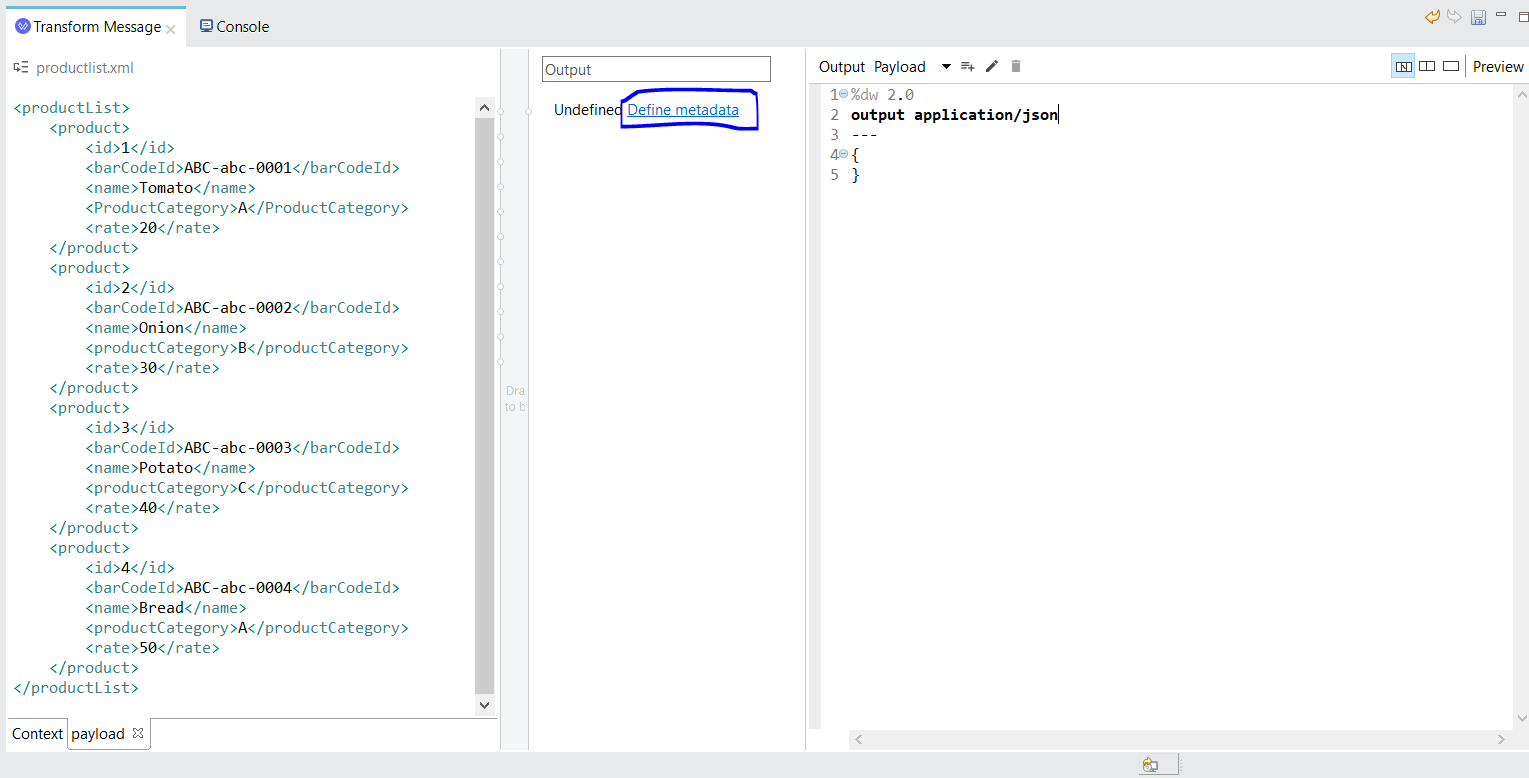
1. Drag transformer from mule palette
2. Define metadata like the previous section, with name productList
3. Change application/java to application/json



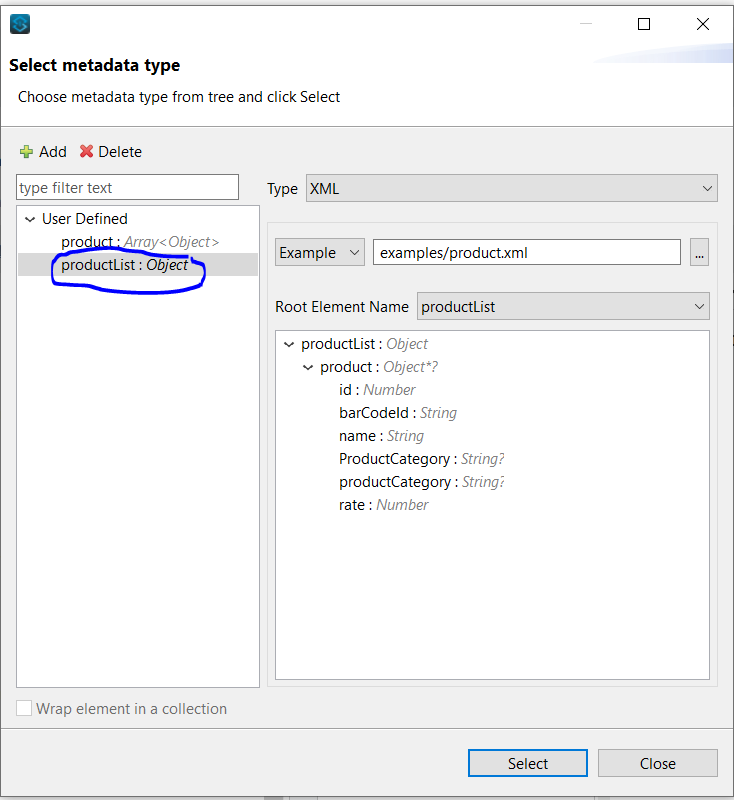
1. Write payload on the body
2. Click Preview

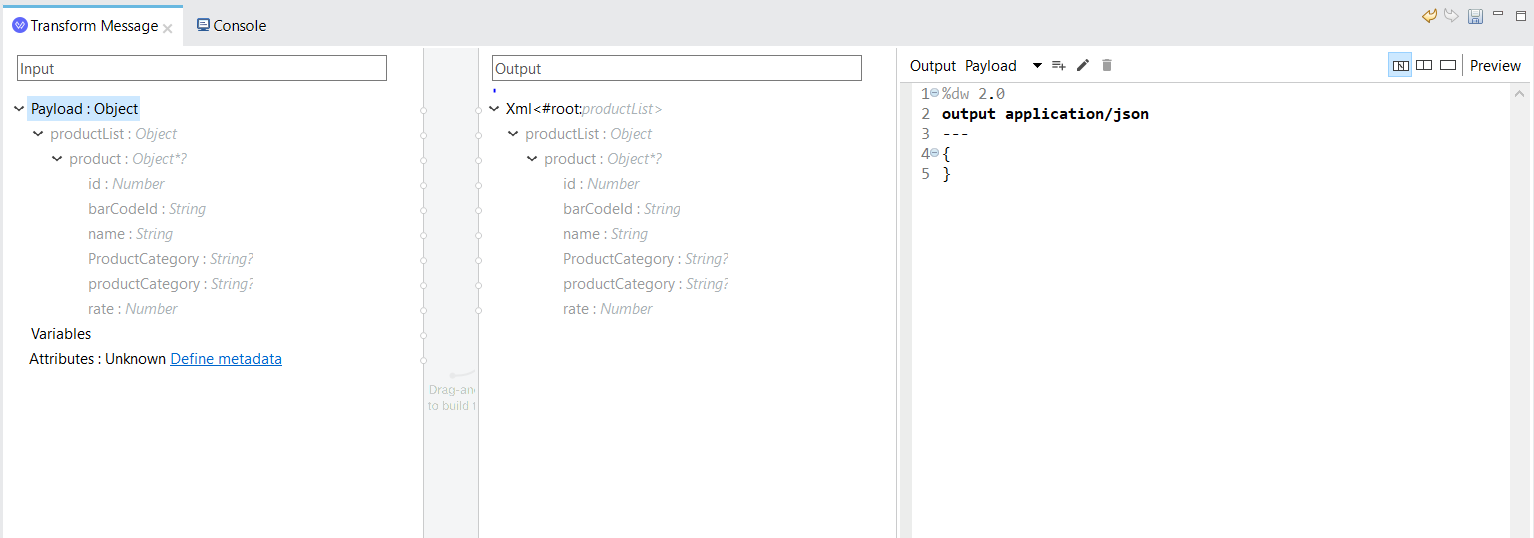


1. Define metadata

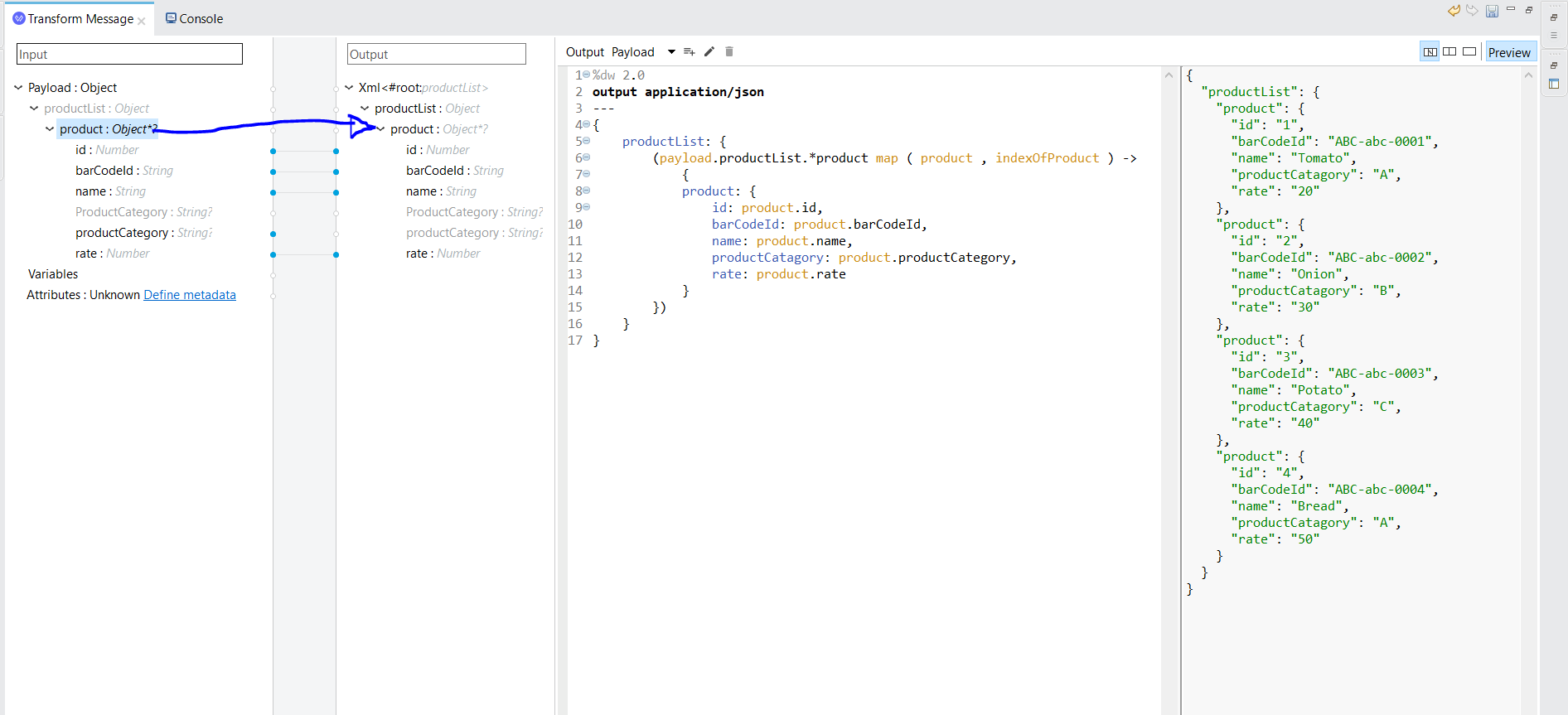


1. Click Select button

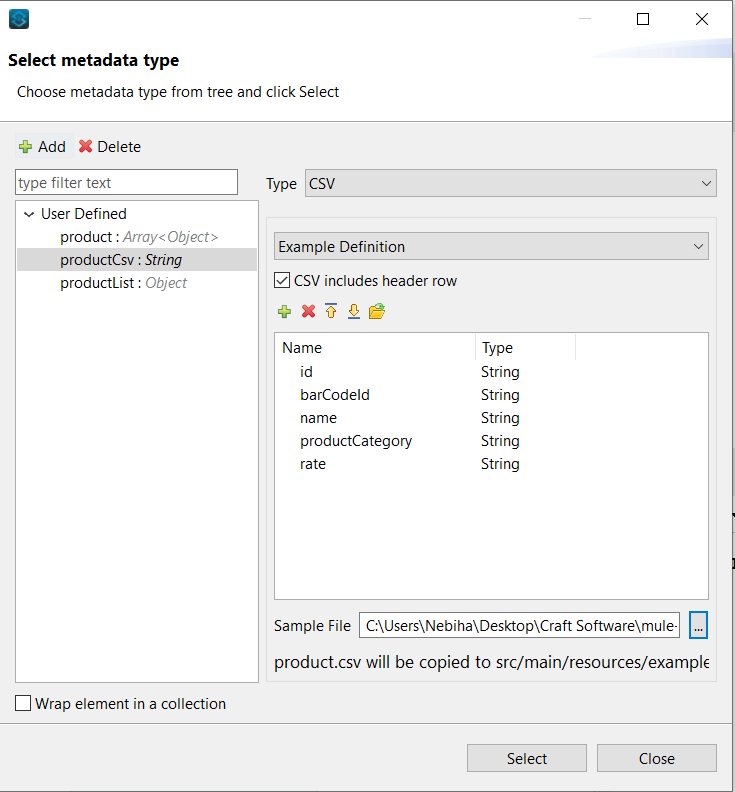




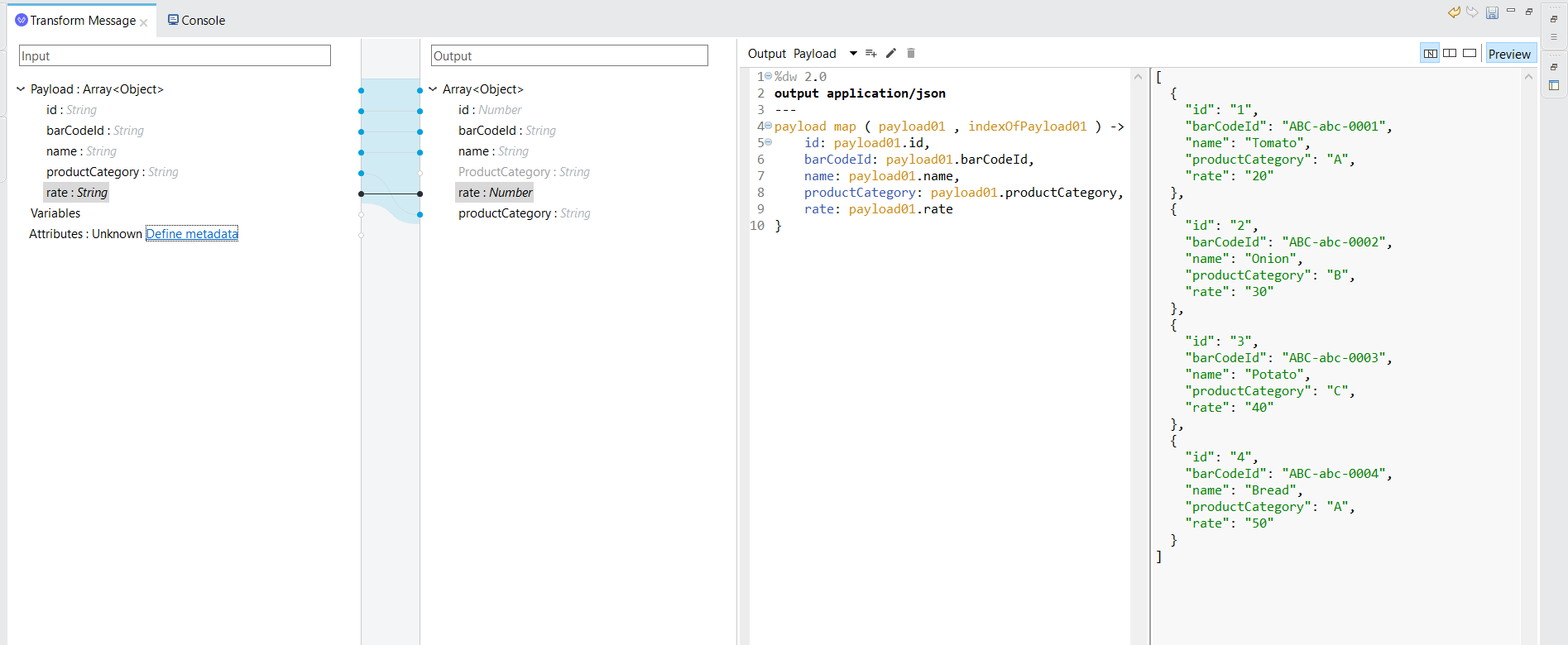
1. Once the source and target definitions have been created, using the drag and drop functionality, map all the fields. So, drag and drop product from payload object to output defined metadata
2. The script will be automatically generated
3. Click **Preview** window to see transformed json data



1. **CSV to JSON**
2. Define product csv



1. Click **Select** button
2. Define metadata
3. Add another object json definition use JSON file defined in the previous section
4. Map all the fields from source CSV format to target JSON format



**D. Flatte File to XML**

1. Define productFlate
2. Define xml metadata for xml transformation
3. Drag and drop metadata