Write a data transformation Java class that transforms given;

1. list of name:value data into JSON Object data

2. JSON formatted string data into JSON Object data

3. JSON data into name:value Map

4. JSON data into XML data

Within your class, make sure to handle/process;

1. Malformed data

2. Data in wrong format

**Solution:**

**Program code**

**Jsonprocessor.java**

|  |
| --- |
| import org.json.\*; import java.util.\*; public class Jsonprocessor {  public static void main(String args[]) throws JSONException {  JSONObject myJson = new JSONObject();  Map < String, String > myMap = new HashMap < String, String > ();  //map is initialized  myMap.put("id", "1");  myMap.put("name", "test\_name");  myMap.put("address", "test\_address");   //convert list of name:value data into JSON Object data  System.out.println("\nList of name:value data into JSON Object data");  try {  myJson = new JSONObject(myMap);  //display the json object  System.out.println(myJson);  } catch (Exception exception) {  System.out.println("Data not in expected format. Error: " + exception);  }   //convert JSON formatted string data into JSON Object data  System.out.println("\nJSON formatted string data into JSON Object data");  String jsonString = "{ 'id':'1','name':'test\_name','address':'test\_address'}";  try {  JSONObject json = new JSONObject(jsonString);  //display the json object  System.out.println(json);  } catch (org.json.JSONException exception) {  System.out.println("Given string is not formated correctly! Error: " + exception);  }  //convert JSON data into name:value Map  System.out.println("\nJSON data into name:value Map");  try {  Map mapFromJson = myJson.toMap();  //display the Map  System.out.println(mapFromJson.toString());  } catch (org.json.JSONException exception) {  System.out.println("Data is not formated correctly! Error: " + exception);  }   //convert JSON data into XML data  System.out.println("\nJSON data into XML data");  try {  String str = XML.toString(myJson);  //display the XML  System.out.println(str);  } catch (org.json.JSONException exception) {  System.out.println("Data is not formated correctly! Error: " + exception);  }  } } |

**Steps to run the program**

1. Download org.json package which is required by the code.
2. Put the jar file in the same directory where the code lies.
3. Run following command to compile the code

javac -cp json-20210307.jar Jsonprocessor.java

1. Finally execute the program

java -cp json-20210307.jar:. Jsonprocessor

**Explanation**

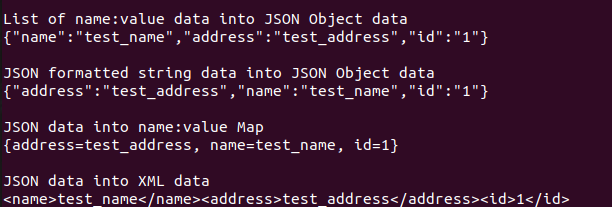
Map variable myMap is initialized with some data which is transformed to JSON object using JSONObject constructor. Similarly JSON formatted string jsonString is also transformed to JSON object.

JSON Object is then transformed to Map using toMap() method of JSONObject class. Finally, the XML class is used to convert JSONObject to XML.

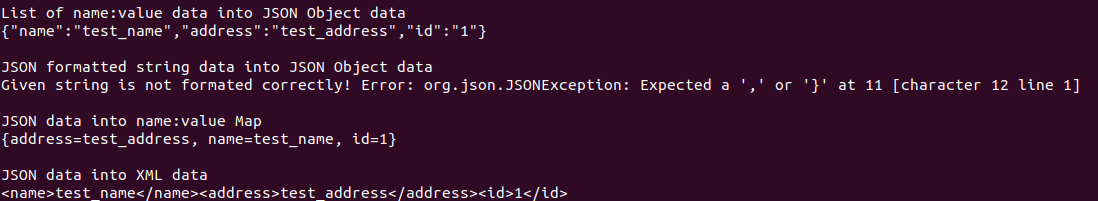
**Outputs**

**Figure 1**

Output of the program

****

**Figure 2**

Exception Handling when wrong formatted string is supplied to transform to JSONObject:****

**References**

Albano, J. (2020, March 30). Introduction to JSON With Java. *Dzone.Com*.

<https://dzone.com/articles/introduction-to-json-with-java>

Java documentation Version 20210307. (2021, March 7). *Jar-Download*.

<https://jar-download.com/artifacts/org.json/json/20210307/documentation>