**2. Automation question: (preferably in Java language)** **Part 1:**

The attached XML file contains a list of websites in the following structure:

  <Website>

    <URL>https://www.sephora.sg/</URL>

    <Tile>Sephora Singapore</Tile>

    <Description>Sephora, where beauty beats. Discover the best in makeup, skin care, and more from a wide selection of beauty brands. Free shipping for orders above S$40.</Description>

    <CreatedDate>10/30/2019 6:41:56 AM</CreatedDate>

  </Website>

Please answer the following questions:

1. De-serialize attached XML
2. Add new attributes to XML
3. Convert XML to JSON
4. Get all websites that were created after September 1
5. Print information of the website as following format:

Website 1:

URL: URL

Title: Tile

Description: Description

Created Date: CreatedDate

Website 2:

URL: URL

Title: Tile

Description: Description

Created Date: CreatedDate

Answer A:

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "Website")

public class Website {

private String URL;

private String Tile;

private String Description;

private String CreatedDate;

@XmlElement(name = "URL")

public void setURL(String URL) {

this.URL = URL;

}

public String getURL() {

return URL;

}

@XmlElement(name = "Tile")

public void setTile(String Tile) {

this.Tile = Tile;

}

public String getTile() {

return Tile;

}

@XmlElement(name = "Description")

public void setDescription(String Description) {

this.Description = Description;

}

public String getDescription() {

return Description;

}

@XmlElement(name = "CreatedDate")

public void setCreatedDate(String CreatedDate) {

this.CreatedDate = CreatedDate;

}

public String getCreatedDate() {

return CreatedDate;

}

@Override

public String toString() {

return "Website [URL=" + URL + ", Tile=" + Tile + ", Description=" + Description + ", CreatedDate=" + CreatedDate + "]";

}

}

Answer B:

import javax.xml.bind.annotation.XmlAttribute;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "Website")

public class Website {

private String id;

private String category;

private String URL;

private String Tile;

private String Description;

private String CreatedDate;

@XmlAttribute(name = "id")

public void setId(String id) {

this.id = id;

}

public String getId() {

return id;

}

@XmlAttribute(name = "category")

public void setCategory(String category) {

this.category = category;

}

public String getCategory() {

return category;

}

@XmlElement(name = "URL")

public void setURL(String URL) {

this.URL = URL;

}

public String getURL() {

return URL;

}

@XmlElement(name = "Tile")

public void setTile(String Tile) {

this.Tile = Tile;

}

public String getTile() {

return Tile;

}

@XmlElement(name = "Description")

public void setDescription(String Description) {

this.Description = Description;

}

public String getDescription() {

return Description;

}

@XmlElement(name = "CreatedDate")

public void setCreatedDate(String CreatedDate) {

this.CreatedDate = CreatedDate;

}

public String getCreatedDate() {

return CreatedDate;

}

@Override

public String toString() {

return "Website [id=" + id + ", category=" + category + ", URL=" + URL + ", Tile=" + Tile + ", Description=" + Description + ", CreatedDate=" + CreatedDate + "]";

}

}

import javax.xml.bind.annotation.XmlAttribute;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "Website")

public class Website {

private String id;

private String category;

private String URL;

private String Tile;

private String Description;

private String CreatedDate;

@XmlAttribute(name = "id")

public void setId(String id) {

this.id = id;

}

public String getId() {

return id;

}

@XmlAttribute(name = "category")

public void setCategory(String category) {

this.category = category;

}

public String getCategory() {

return category;

}

@XmlElement(name = "URL")

public void setURL(String URL) {

this.URL = URL;

}

public String getURL() {

return URL;

}

@XmlElement(name = "Tile")

public void setTile(String Tile) {

this.Tile = Tile;

}

public String getTile() {

return Tile;

}

@XmlElement(name = "Description")

public void setDescription(String Description) {

this.Description = Description;

}

public String getDescription() {

return Description;

}

@XmlElement(name = "CreatedDate")

public void setCreatedDate(String CreatedDate) {

this.CreatedDate = CreatedDate;

}

public String getCreatedDate() {

return CreatedDate;

}

@Override

public String toString() {

return "Website [id=" + id + ", category=" + category + ", URL=" + URL + ", Tile=" + Tile + ", Description=" + Description + ", CreatedDate=" + CreatedDate + "]";

}

}

Answer C:

Answer D:

import com.fasterxml.jackson.databind.ObjectMapper;

import java.io.StringReader;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Unmarshaller;

public class XmlToJsonConverter {

public static void main(String[] args) {

String xmlData = "<Website id=\"12345\" category=\"Beauty\">\n" +

" <URL>https://www.sephora.sg/</URL>\n" +

" <Tile>Sephora Singapore</Tile>\n" +

" <Description>Sephora, where beauty beats. Discover the best in makeup, skin care, and more from a wide selection of beauty brands. Free shipping for orders above S$40.</Description>\n" +

" <CreatedDate>10/30/2019 6:41:56 AM</CreatedDate>\n" +

"</Website>";

try {

// Deserialize XML to Java object

JAXBContext context = JAXBContext.newInstance(Website.class);

Unmarshaller unmarshaller = context.createUnmarshaller();

StringReader reader = new StringReader(xmlData);

Website website = (Website) unmarshaller.unmarshal(reader);

// Serialize Java object to JSON

ObjectMapper objectMapper = new ObjectMapper();

String jsonResult = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(website);

System.out.println(jsonResult);

} catch (JAXBException | IOException e) {

e.printStackTrace();

}

}

}

import java.util.List;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "Websites")

public class Websites {

private List<Website> websites;

@XmlElement(name = "Website")

public void setWebsites(List<Website> websites) {

this.websites = websites;

}

public List<Website> getWebsites() {

return websites;

}

}

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Unmarshaller;

import java.io.StringReader;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import java.util.stream.Collectors;

public class FilterWebsites {

public static void main(String[] args) {

String xmlData = "<Websites>\n" +

" <Website>\n" +

" <URL>https://www.sephora.sg/</URL>\n" +

" <Tile>Sephora Singapore</Tile>\n" +

" <Description>Sephora, where beauty beats. Discover the best in makeup, skin care, and more from a wide selection of beauty brands. Free shipping for orders above S$40.</Description>\n" +

" <CreatedDate>10/30/2019 6:41:56 AM</CreatedDate>\n" +

" </Website>\n" +

" <!-- More Website elements -->\n" +

"</Websites>";

try {

// Deserialize XML to Java objects

JAXBContext context = JAXBContext.newInstance(Websites.class);

Unmarshaller unmarshaller = context.createUnmarshaller();

StringReader reader = new StringReader(xmlData);

Websites websitesWrapper = (Websites) unmarshaller.unmarshal(reader);

List<Website> websites = websitesWrapper.getWebsites();

// Filter websites created after September 1

SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy hh:mm:ss a");

Date thresholdDate = sdf.parse("09/01/2019 00:00:00 AM");

List<Website> filteredWebsites = websites.stream()

.filter(website -> {

try {

Date createdDate = sdf.parse(website.getCreatedDate());

return createdDate.after(thresholdDate);

} catch (ParseException e) {

e.printStackTrace();

return false;

}

})

.collect(Collectors.toList());

// Print filtered websites

int index = 1;

for (Website website : filteredWebsites) {

System.out.println("Website " + index + ":");

System.out.println("URL: " + website.getURL());

System.out.println("Title: " + website.getTile());

System.out.println("Description: " + website.getDescription());

System.out.println("Created Date: " + website.getCreatedDate());

System.out.println();

index++;

}

} catch (JAXBException | ParseException e) {

e.printStackTrace();

}

}

}

Answer E:

Website 1:

URL: https://www.sephora.sg/

Title: Sephora Singapore

Description: Sephora, where beauty beats. Discover the best in makeup, skin care, and more from a wide selection of beauty brands. Free shipping for orders above S$40.

Created Date: 10/30/2019 6:41:56 AM