Homework # 50 - JAXB Parsing + XSD Validation Java class # 30

pdf pom.xml

Please complete the following tasks (5):

01. Create project and GitHub repository "HW_50" or download

```
a. Login into the https://github.com
b. Click on New repository button
c. Enter Repository name "HW_50" (Public / Initialize this repository with a README)
d. Click on "Copy to download" button (Use HTTPS)
e. Click on "Copy to clipboard" button (https://github.com/[your_username]/HW_50.git)
```

02. Delete the following classes:

~~App.java~~

~~AppTest.java~~

03. Update Java version of the project: from [J2SE- 1.5] to [Java SE 8 [1.8.0_221]]

04. At the pom.xml add dependency for okhttp version: 4.1.1

```
<!-- https://mvnrepository.com/artifact/com.squareup.okhttp3/okhttp -->
<dependency>
        <groupId>com.squareup.okhttp3</groupId>
        <artifactId>okhttp</artifactId>
        <version>4.1.1</version>
</dependency>
```

05. Create Java classes (5): Employees; EmployeesList, EmployeesType; EmployeesListType; EmployeesOkHttp3

employee.xml   http://alex.academy/xml/employee.xml

```
<?xml version="1.0" encoding="utf-8"?>
<employees xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="employees.xsd">
      <employee id="385450">
            <firstname>Mark</firstname>
            <lastname>Twain</lastname>
            <title>QA</title>
            <hiredate>April 20, 1910</hiredate>
            <phone>2125551212</phone>
            <email>mark_twain@gmail.com</email>
      </employee>
</employees>
```

employees.xml   http://alex.academy/xml/employees.xml

```
<?xml version="1.0" encoding="utf-8"?>
<employees xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="employees.xsd">
      <employee id="385450">
            <firstname>John</firstname>
            <lastname>Muir</lastname>
            <title>Engineer</title>
            <hiredate>October 15, 2005</hiredate>
            <phone>2121234567</phone>
            <email>john_smith@gmail.com</email>
      </employee>
      <employee id="385730">
            <firstname>Christina</firstname>
            <lastname>Aguilera</lastname>
            <title>Manager</title>
            <hiredate>September 1, 2010</hiredate>
            <phone>4157654321</phone>
            <email>christina_johnson_smith@gmail.com</email>
      </employee>
</employees>
```

EmployeesType.java

```
package core;
import javax.xml.bind.annotation.*;

@XmlRootElement(name = "employees")
public class EmployeesType {

        @XmlElement(name = "employee")
        protected Level2 employee;

        static class Level2 {

                @XmlAttribute(name = "id")      String id;

                @XmlElement(name = "firstname") String firstname;
                @XmlElement(name = "lastname")  String lastname;
                @XmlElement(name = "title")     String title;
                @XmlElement(name = "hiredate")  String hiredate;
                @XmlElement(name = "phone")     String phone;
                @XmlElement(name = "email")     String email;
        }
}
```

Employees.java

```
package core;
import java.io.File;
import javax.xml.bind.*;
import javax.xml.parsers.*;
import org.w3c.dom.Document;

public class Employees {

        public static void main(String[] args) throws Exception {

                String xml = "http://alex.academy/xml/employee.xml";
                DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
                DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
                Document doc = dBuilder.parse(xml);
                doc.getDocumentElement().normalize();
//              File file = new File("employee.xml");
EmployeesType o = (EmployeesType) JAXBContext.newInstance(EmployeesType.class).createUnmarshaller().unmarshal(doc);

                System.out.println("Employee ID: " +   o.employee.id + "; "
                                   + "First name: " + o.employee.firstname + "; "
                                   + "Last Name: " +  o.employee.lastname + "; "
                                   + "Title: " +      o.employee.title + "; "
                                   + "Hire date: " +  o.employee.hiredate + "; "
                                   + "Phone: " +      o.employee.phone + "; "
                                   + "Email: " +      o.employee.email
                                   );
        }
}
```

```
> Employee ID: 385450;
> First name: Mark;
> Last Name: Twain;
> Title: QA;
> Hire date: April 20, 1910;
> Phone: 2125551212;
> Email: mark_twain@gmail.com
```

EmployeesListType.java

```
package core;
import java.util.*;
import javax.xml.bind.annotation.*;

@XmlRootElement(name = "employees")
public class EmployeesListType {
        List<EmployeeType> employees = new ArrayList<EmployeeType>();

        @XmlElement(name = "employee")
        List<EmployeeType> getEmployees() {return employees;}

        public static class EmployeeType {

                List<String> id = new ArrayList<String>();
                @XmlAttribute(name = "id") List<String> getId()              {return id;}

                List<String> firstname = new ArrayList<String>();
                @XmlElement(name = "firstname") List<String> getFirstNames() {return firstname;}

                List<String> lastname = new ArrayList<String>();
                @XmlElement(name = "lastname") List<String> getLastNames()   {return lastname;}

                List<String> title = new ArrayList<String>();
                @XmlElement(name = "title") List<String> getTitle()          {return title;}

                List<String> hiredate = new ArrayList<String>();
                @XmlElement(name = "hiredate") List<String> getHireDate()    {return hiredate;}

                List<String> phone = new ArrayList<String>();
                @XmlElement(name = "phone") List<String> getPhone()          {return phone;}

                List<String> email = new ArrayList<String>();
                @XmlElement(name = "email") List<String> getEmail()          {return email;}
        }
}
```

EmployeesList.java

```
package core;
import java.io.File;
import java.util.*;
import javax.xml.bind.*;
import javax.xml.parsers.*;
import org.w3c.dom.Document;

public class EmployeesList {

        public static void main(String[] args) throws Exception {

                String xml = "http://alex.academy/xml/employees.xml";
                DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
                DocumentBuilder db = dbf.newDocumentBuilder();
                Document doc = db.parse(xml);
                doc.getDocumentElement().normalize();
                File file = new File("employees.xml");
EmployeesListType o = (EmployeesListType) JAXBContext.newInstance(EmployeesListType.class).createUnmarshaller().unmarshal(doc);

                for (EmployeesListType.EmployeeType emp : o.getEmployees()) {
                        List<String> id = new ArrayList<String>(); id.addAll(emp.getId());
                        List<String> fn = new ArrayList<String>(); fn.addAll(emp.getFirstNames());
                        List<String> ln = new ArrayList<String>(); ln.addAll(emp.getLastNames());
                        List<String> t = new ArrayList<String>(); t.addAll(emp.getTitle());
                        List<String> hd = new ArrayList<String>(); hd.addAll(emp.getHireDate());
                        List<String> p = new ArrayList<String>(); p.addAll(emp.getPhone());
                        List<String> e = new ArrayList<String>(); e.addAll(emp.getEmail());

                System.out.println("Employee ID: " + id.toString().replaceAll("\\[|\\]", "") + "; "
                                   + "First name: " + fn.toString().replaceAll("\\[|\\]", "") + "; "
                                   + "Last Name: "  + ln.toString().replaceAll("\\[|\\]", "") + "; "
                                   + "Title: "      + t.toString().replaceAll("\\[|\\]", "")  + "; "
                                   + "Hire date: "  + hd.toString().replaceAll("\\[|\\]", "") + "; "
                                   + "Phone: "      + p.toString().replaceAll("\\[|\\]", "")  + "; "
                                   + "Email: "      + e.toString().replaceAll("\\[|\\]", "")
                                   );
                }
}}
```

```
> Employee ID: 385450; First name: John; Last Name: Muir; Title: Engineer; Hire date: October 15, 2005; Phone: 2121234567; Email: john_smith@gmail.com
> Employee ID: 385730; First name: Christina; Last Name: Moore; Title: Manager; Hire date: October 10, 2015; Phone: 4157654321; Email: christina_moore@gmail.com
```

EmployeesOkHttp3.java

```
package core;
import java.io.StringReader;
import javax.xml.bind.*;
import okhttp3.*;

public class EmployeesOkHttp3 {

        public static void main(String[] args) throws Exception {
                String xml = "http://alex.academy/xml/employee.xml";
                OkHttpClient client = new OkHttpClient();
                Request request = new Request.Builder().url(xml).get().build();
//              MediaType mediaType = MediaType.parse("application/xml");
//              RequestBody body = RequestBody.create(mediaType, "<user> </user>");
//              Request request = new Request.Builder().url(xml).post(body).build();
StringReader sr = new StringReader((String) client.newCall(request).execute().body().string());
//              Response response = client.newCall(request).execute();
//              String doc = response.body().string();
//              StringReader sr = new StringReader(doc);
EmployeesType o = (EmployeesType) JAXBContext.newInstance(EmployeesType.class).createUnmarshaller().unmarshal(sr);
//              JAXBContext jaxbC = JAXBContext.newInstance(EmployeesType.class);
//              Unmarshaller jaxbU = jaxbC.createUnmarshaller();
//              EmployeesType o = (EmployeesType) jaxbU.unmarshal(sr);
                System.out.println( "Employee ID: " +   o.employee.id + "; "
                                  + "First name: " +    o.employee.firstname + "; "
                                  + "Last Name: " +     o.employee.lastname + "; "
                                  + "Title: " +         o.employee.title + "; "
                                  + "Hire date: " +     o.employee.hiredate + "; "
                                  + "Phone: " +         o.employee.phone + "; "
                                  + "Email: " +         o.employee.email
                             );
}}
```

```
> Employee ID: 385450;
> First name: Mark;
> Last Name: Twain;
> Title: QA;
> Hire date: April 20, 1910;
> Phone: 2125551212;
> Email: mark_twain@gmail.com
```

employees.xsd http://alex.academy/xml/employees.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified" elementFormDefault="qualified">
    <xs:element name="employees" type="employeesType"/>
    <xs:complexType name="employeeType">
        <xs:sequence>
            <xs:element name="firstname">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[A-Z]\w*"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="lastname">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[A-Z]\w*"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="title">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="Engineer"/>
                        <xs:enumeration value="Manager"/>
                        <xs:enumeration value="QA"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="hiredate">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                   <xs:pattern value="([ADFJMNOS]\w*\s)([0-9]{1,2},)\s([0-9]{4})"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="phone">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="\d{10}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="email">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value=".+\@.+\..{2,18}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
        <xs:attribute type="xs:string" name="id" use="required"/>
    </xs:complexType>
    <xs:complexType name="employeesType">
        <xs:sequence>
    <xs:element type="employeeType" name="employee" maxOccurs="unbounded" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

employees.dtd http://alex.academy/xml/employees.dtd

```
<!ELEMENT employees (employee)+>
<!ELEMENT employee (firstname , lastname , title , hiredate , phone , email)>
<!ATTLIST employee xmlns CDATA #FIXED "">
<!ATTLIST employee id CDATA #REQUIRED>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

employees.xml http://alex.academy/xml/employees-dtd.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE catalog SYSTEM "employees.dtd">
<employees>
        <employee id="385450">
                <firstname>Mark</firstname>
                <lastname>Twain</lastname>
                <title>QA</title>
                <hiredate>April 20, 1910</hiredate>
                <phone>2125551212</phone>
                <email>mark_twain@gmail.com</email>
        </employee>
</employees>
```

employees.xml http://alex.academy/xml/employeesdtd.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE employees [
        <!ELEMENT employees (employee)+>
        <!ELEMENT employee (firstname , lastname , title , hiredate , phone , email)>
        <!ATTLIST employee xmlns CDATA #FIXED "">
        <!ATTLIST employee id CDATA #REQUIRED>
        <!ELEMENT firstname (#PCDATA)>
        <!ELEMENT lastname (#PCDATA)>
        <!ELEMENT title (#PCDATA)>
        <!ELEMENT hiredate (#PCDATA)>
        <!ELEMENT phone (#PCDATA)>
        <!ELEMENT email (#PCDATA)>
        ]>
<employees xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="emp.xsd">
        <employee xmlns="" id="385450">
                <firstname>John</firstname>
                <lastname>Muir</lastname>
                <title>Engineer</title>
                <hiredate>October 15, 2005</hiredate>
                <phone>2121234567</phone>
                <email>john_smith@gmail.com</email>
        </employee>
        <employee xmlns="" id="385730">
                <firstname>Christina</firstname>
                <lastname>Aguilera</lastname>
                <title>Manager</title>
                <hiredate>September 1, 2010</hiredate>
                <phone>4157654321</phone>
                <email>christina_johnson_smith@gmail.com</email>
        </employee>
</employees>
```

XSD Validation

Please complete the following tasks (5):

01. Modify pom.xml - add the following dependencies (3) TestNG, Hamcrest, OkHttp

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
          <groupId>org.testng</groupId>
          <artifactId>testng</artifactId>
          <version>7.0.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-all -->
<dependency>
          <groupId>org.hamcrest</groupId>
          <artifactId>hamcrest-all</artifactId>
          <version>1.3</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.squareup.okhttp3/okhttp -->
<dependency>
          <groupId>com.squareup.okhttp3</groupId>
          <artifactId>okhttp</artifactId>
          <version>4.1.1</version>
</dependency>
```

Report (1) Surefire

```
<reporting>
    <plugins>
          <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-report-plugin</artifactId>
                <version>2.20.1</version>
          </plugin>
          <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-project-info-reports-plugin</artifactId>
                <version>2.9</version>
          </plugin>
          <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jxr-plugin</artifactId>
                <version>2.5</version>
          </plugin>
    </plugins>
</reporting>
```

02. Create TestNG suite (1): testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="BAT">
    <test name="XSD Validation">
          <classes>
                <class name="core.XSDValidationTest"></class>
          </classes>
    </test>
</suite>
```

03. Create TestNG class (1): XSDValidationTest.java

```
package core;
import org.testng.*;
import org.testng.annotations.*;
public class XSDValidationTest {
      @Test public void test() {assertThat(1, equalTo(1));}}
```

04. Create csv file (1): Workspace/BAT/src/resources/test_data/csv/bat/xsd_validation.csv

```
BAT-01.01 (XSD Validation Type # 1),src/resources/test_data/xsd/bat/bat_t1.xsd,http://alex.academy/xml/bat.xml
BAT-01.02 (XSD Validation Type # 2),src/resources/test_data/xsd/bat/bat_t2.xsd,http://alex.academy/xml/bat.xml
BAT-01.03 (XSD Validation Type # 3),src/resources/test_data/xsd/bat/bat_t3.xsd,http://alex.academy/xml/bat.xml
BAT-01.04 (XSD Validation Type # 1),src/resources/test_data/xsd/bat/bat_t1.xsd,http://alex.academy/xml/bat_e1.xml
BAT-01.05 (XSD Validation Type # 2),src/resources/test_data/xsd/bat/bat_t2.xsd,http://alex.academy/xml/bat_e2.xml
BAT-01.06 (XSD Validation Type # 3),src/resources/test_data/xsd/bat/bat_t3.xsd,http://alex.academy/xml/bat_e3.xml
```

05. Create xsd files (3): Workspace/BAT/src/resources/test_data/xsd/bat/bat_t1.xsd example

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:element name="list">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="poet">
                                        <xs:complexType>
                                                <xs:sequence>
                                                        <xs:element name="name"/>
                                                        <xs:element name="dob"/>
                                                        <xs:element name="age"/>
                                                        <xs:element name="nobel"/>
                                                        <xs:element name="qa" type='xs:string' minOccurs="0"/>
                                                </xs:sequence>
                                                <xs:attribute name="id"/>
                                        </xs:complexType>
                                </xs:element>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

Workspace/BAT/src/resources/test_data/xsd/bat/bat_t2.xsd example

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:element name="list">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="poet">
                                        <xs:complexType>
                                                <xs:sequence>
                                                        <xs:element name="name">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:string">
                                                                     <xs:pattern value="[A-Z][a-z]*\s[A-Z][a-z]*"/>
                                                                        <xs:minLength value="2"/>
                                                                        <xs:maxLength value="20"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:element>
                                                        <xs:element name="dob" type="xs:date"/>
                                                        <xs:element name="age">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:integer">
                                                                                <xs:minInclusive value="0"/>
                                                                                <xs:maxInclusive value="100"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:element>
                                                        <xs:element name="nobel" type="xs:boolean"/>
                                                </xs:sequence>
                                                <xs:attribute name="id" use="required">
                                                        <xs:simpleType>
                                                                <xs:restriction base="xs:positiveInteger">
                                                                        <xs:pattern value="\d{6}"/>
                                                                </xs:restriction>
                                                        </xs:simpleType>
                                                </xs:attribute>
                                        </xs:complexType>
                                </xs:element>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

Workspace/BAT/src/resources/test_data/xsd/bat/bat_t3.xsd example

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:element name="list">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="poet">
                                        <xs:complexType>
                                                <xs:sequence>
                                                        <xs:element name="name">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:string">
                                                                        <xs:pattern value="Boris Pasternak"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:element>
                                                        <xs:element name="dob">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:date">
                                                                        <xs:enumeration value="1890-02-10"/>
                                                                        <xs:enumeration value="1890-01-29"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:element>
                                                        <xs:element name="age">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:integer">
                                                                                <xs:pattern value="70"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:element>
                                                        <xs:element name="nobel">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:boolean">
                                                                                <xs:pattern value="true"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:element>
                                                </xs:sequence>
                                                <xs:attribute name="id"/>
                                        </xs:complexType>
                                </xs:element>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

bat.xml http://alex.academy/xml/bat.xml

```
<?xml version="1.0" encoding="utf-8"?>
<list>
        <poet id="385450">
                <name>Boris Pasternak</name>
                <dob>1890-02-10</dob>
                <age>70</age>
                <nobel>true</nobel>
        </poet>
</list>
```

bat_e1.xml http://alex.academy/xml/bat_e1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<list>
        <poet id="385450">
                <name>Boris Pasternak</name>
                <dob>1890-02-10</dob>
                <age>70</age>
                ~~<nobel>true</nobel>~~
        </poet>
</list>
```

bat_e2.xml

http://alex.academy/xml/bat_e2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<list>
        <poet id="385450">
                <name>Boris Pasternak</name>
                <dob>1890-02-10</dob>
                <age>~~seventy~~</age>
                <nobel>true</nobel>
        </poet>
</list>
```

bat_e3.xml

http://alex.academy/xml/bat_e3.xml

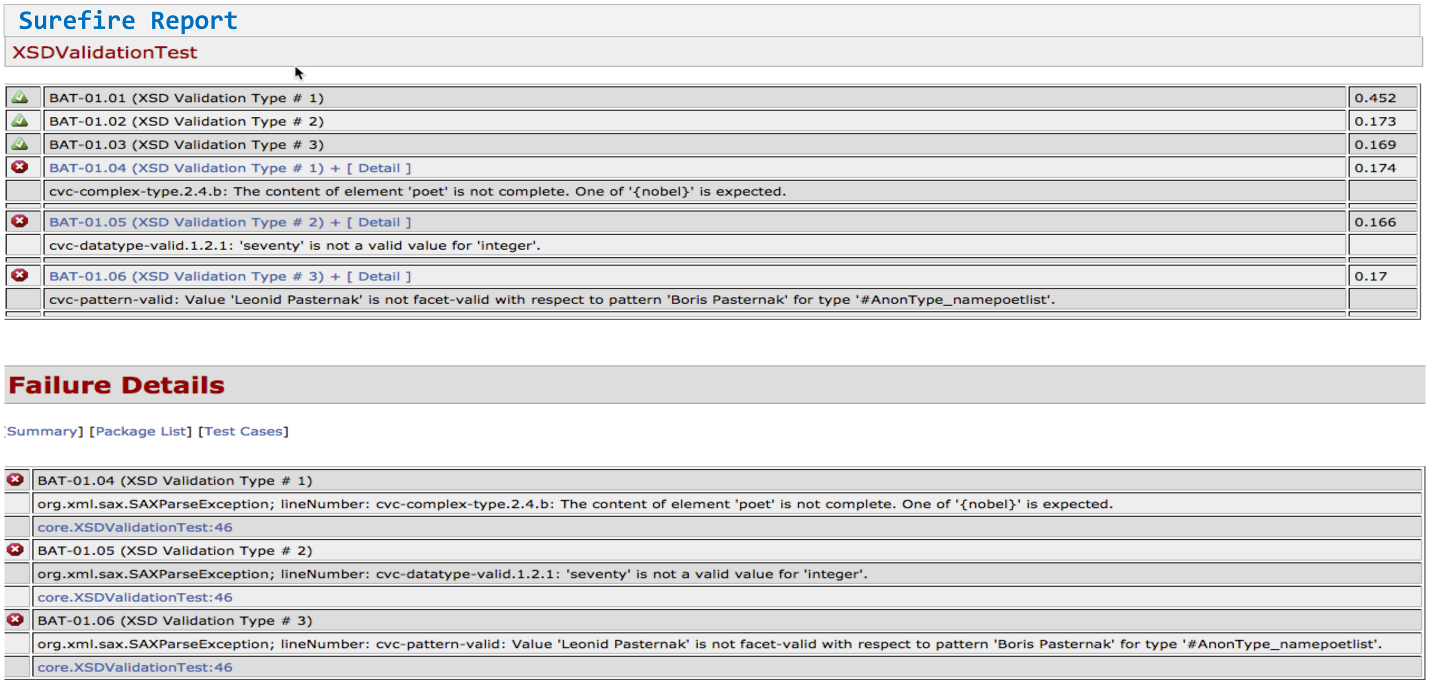
```
<?xml version="1.0" encoding="utf-8"?>
<list>
        <poet id="385450">
                <name>~~Leonid~~ Pasternak</name>
                <dob>1890-02-10</dob>
                <age>70</age>
                <nobel>true</nobel>
        </poet>
</list>
```

XSDValidationTest.java

```
package core;
import static org.hamcrest.CoreMatchers.*;
import static org.hamcrest.MatcherAssert.assertThat;
import java.io.*;
import java.net.*;
import java.util.*;
import org.testng.*;
import org.testng.annotations.*;
import java.lang.reflect.Method;
import javax.xml.XMLConstants;
import javax.xml.transform.stream.StreamSource;
import javax.xml.validation.SchemaFactory;
import org.xml.sax.SAXException;
import okhttp3.*;
import okhttp3.Request.Builder;

public class XSDValidationTest implements ITest{
        String csvFile = "./test_data/csv/bat/xsd_validation.csv";
    // String csvFile = System.getProperty("testcases"); // mvn site -Dtestcases="./input.csv"

        @DataProvider(name = "dp")
        public Iterator<String[]> a2d() throws InterruptedException, IOException {
                String cvsLine = "";
                String[] a = null;
                ArrayList<String[]> al = new ArrayList<>();
                BufferedReader br = new BufferedReader(new FileReader(csvFile));
                while ((cvsLine = br.readLine()) != null) {
                        a = cvsLine.split(",");
                        al.add(a);}
                br.close();
                return al.iterator();
                                                                                                 }

        @Test(dataProvider = "dp")
        public void test(String tc_id, String xsd_file, String xml_url) throws SAXException, IOException{
                System.out.println(tc_id);
```

```java
                assertThat(validateXMLSchema(new File(xsd_file), new URL(xml_url)), equalTo(true));
                }
        public static boolean validateXMLSchema(File xsd_file, URL xml_url) throws SAXException, IOException{
                OkHttpClient client = new OkHttpClient();
                Builder b = new Request.Builder();
                ByteArrayInputStream xml_str = new ByteArrayInputStream(client.newCall(b.url(xml_url)
                            .get().build()).execute().body().string().getBytes());
                SchemaFactory.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI)
                            .newSchema(xsd_file).newValidator().validate(new StreamSource(xml_str));
        return true;
        }
}
```

```
BAT-01.01 (XSD Validation Type # 1)
BAT-01.02 (XSD Validation Type # 2)
BAT-01.03 (XSD Validation Type # 3)
BAT-01.04 (XSD Validation Type # 1)
The content of element 'poet' is not complete. One of '{nobel}' is expected.
BAT-01.05 (XSD Validation Type # 2)
'seventy' is not a valid value for 'integer'.
BAT-01.06 (XSD Validation Type # 3)
Value 'Leonid Pasternak' is not facet-valid with respect to pattern 'Boris Pasternak'
```



Surefire Report

XSDValidationTest

✓	BAT-01.01 (XSD Validation Type # 1)	0.452
✓	BAT-01.02 (XSD Validation Type # 2)	0.173
✓	BAT-01.03 (XSD Validation Type # 3)	0.169
☒	BAT-01.04 (XSD Validation Type # 1) + [Detail]	0.174
	cvc-complex-type.2.4.b: The content of element 'poet' is not complete. One of '{nobel}' is expected.	
☒	BAT-01.05 (XSD Validation Type # 2) + [Detail]	0.166
	cvc-datatype-valid.1.2.1: 'seventy' is not a valid value for 'integer'.	
☒	BAT-01.06 (XSD Validation Type # 3) + [Detail]	0.17
	cvc-pattern-valid: Value 'Leonid Pasternak' is not facet-valid with respect to pattern 'Boris Pasternak' for type '#AnonType_namepoetlist'.	

Failure Details

Summary] [Package List] [Test Cases]

☒	BAT-01.04 (XSD Validation Type # 1)
	org.xml.sax.SAXParseException; lineNumber: cvc-complex-type.2.4.b: The content of element 'poet' is not complete. One of '{nobel}' is expected.
	core.XSDValidationTest:46
☒	BAT-01.05 (XSD Validation Type # 2)
	org.xml.sax.SAXParseException; lineNumber: cvc-datatype-valid.1.2.1: 'seventy' is not a valid value for 'integer'.
	core.XSDValidationTest:46
☒	BAT-01.06 (XSD Validation Type # 3)
	org.xml.sax.SAXParseException; lineNumber: cvc-pattern-valid: Value 'Leonid Pasternak' is not facet-valid with respect to pattern 'Boris Pasternak' for type '#AnonType_namepoetlist'.
	core.XSDValidationTest:46

pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>core</groupId>
        <artifactId>BAT</artifactId>
        <version>1.1</version>
        <packaging>jar</packaging>
        <name>BAT</name>
        <url>http://maven.apache.org</url>
        <properties>
                <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
                <maven.compiler.source>1.8</maven.compiler.source>
                <maven.compiler.target>1.8</maven.compiler.target>
        </properties>
        <dependencies>
                <!-- https://mvnrepository.com/artifact/org.testng/testng -->
                <dependency>
```

```
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>7.0.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-all -->
        <dependency>
            <groupId>org.hamcrest</groupId>
            <artifactId>hamcrest-all</artifactId>
            <version>1.3</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/com.squareup.okhttp3/okhttp -->
        <dependency>
            <groupId>com.squareup.okhttp3</groupId>
            <artifactId>okhttp</artifactId>
            <version>4.1.1</version>
        </dependency>
    </dependencies>
    <reporting>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-report-plugin</artifactId>
                <version>2.20.1</version>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-project-info-reports-plugin</artifactId>
                <version>2.9</version>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jxr-plugin</artifactId>
                <version>2.5</version>
            </plugin>
        </plugins>
    </reporting>
</project>
```

```
[RemoteTestNG] detected TestNG version 7.0.0
BAT-01.01 (XSD Validation Type # 1)
BAT-01.02 (XSD Validation Type # 2)
BAT-01.03 (XSD Validation Type # 3)
BAT-01.04 (XSD Validation Type # 1)
BAT-01.05 (XSD Validation Type # 2)
BAT-01.06 (XSD Validation Type # 3)
PASSED: BAT-01.01 (XSD Validation Type # 1) ("BAT-01.01 (XSD Validation Type # 1)", "src/resources/test_data/xsd/bat/bat_t1.xsd", "http://alex.academy/xml/bat.xml")
PASSED: BAT-01.02 (XSD Validation Type # 2) ("BAT-01.02 (XSD Validation Type # 2)", "src/resources/test_data/xsd/bat/bat_t2.xsd", "http://alex.academy/xml/bat.xml")
PASSED: BAT-01.03 (XSD Validation Type # 3) ("BAT-01.03 (XSD Validation Type # 3)", "src/resources/test_data/xsd/bat/bat_t3.xsd", "http://alex.academy/xml/bat.xml")
FAILED: BAT-01.04 (XSD Validation Type # 1) ("BAT-01.04 (XSD Validation Type # 1)", "src/resources/test_data/xsd/bat/bat_t1.xsd", "http://alex.academy/xml/bat_e1.xml")
org.xml.sax.SAXParseException; lineNumber: 7; columnNumber: 9; cvc-complex-type.2.4.b: The content of element 'poet' is not complete. One of '{nobel}' is expected.
FAILED: BAT-01.05 (XSD Validation Type # 2) ("BAT-01.05 (XSD Validation Type # 2)", "src/resources/test_data/xsd/bat/bat_t2.xsd", "http://alex.academy/xml/bat_e2.xml")
org.xml.sax.SAXParseException; lineNumber: 6; columnNumber: 21; cvc-datatype-valid.1.2.1: 'seventy' is not a valid value for 'integer'.
FAILED: BAT-01.06 (XSD Validation Type # 3) ("BAT-01.06 (XSD Validation Type # 3)", "src/resources/test_data/xsd/bat/bat_t3.xsd", "http://alex.academy/xml/bat_e3.xml")
org.xml.sax.SAXParseException; lineNumber: 4; columnNumber: 32; cvc-pattern-valid: Value 'Leonid Pasternak' is not facet-valid with respect to pattern 'Boris Pasternak'.
===============================================
    Default test
    Tests run: 6, Failures: 3, Skips: 0
===============================================
===============================================
Default suite
Total tests run: 6, Passes: 3, Failures: 3, Skips: 0
===============================================
```