800+ Q&As | Logout | Contact

Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

search here … Go

Home Why? ▾ 300+ Java FAQs ▾ 300+ Big Data FAQs ▾ Courses ▾

Membership ▾ Your Career ▾

Home › bigdata-success.com › Tutorials - Big Data › TUT - File Formats › 03: Convert XML file To an Avro File – writing & reading

03: Convert XML file To an Avro File – writing & reading

Posted on May 5, 2016

This extends the Convert XML file To Sequence File With Hadoop libaries. Avro files are **schema driven** & support **schema evolution**, which means you can add new columns & modify existing columns.

Step 1: The pom.xml file should include the Apache Spark libraries as shown below.

```
<project xmlns="http://maven.apache.org/POM/4.0.0
    xsi:schemaLocation="http://maven.apache.org/P
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mytutorial</groupId>
    <artifactId>sequence-file</artifactId>
```

300+ Java Interview FAQs

300+ Java FAQs 🔥

16+ Java Key Areas Q&As

150+ Java Architect FAQs

80+ Java Code Quality Q&As

150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs 🔥

Tutorials - Big Data

TUT - ⏯ Starting Big Data

TUT - Starting Spark & Scala

```
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>sequence-file</name>
    <url>http://maven.apache.org</url>

    <properties>
        <maven.compiler.source>1.8</maven.compile
        <maven.compiler.target>1.8</maven.compile
        <project.build.sourceEncoding>UTF-8</proj
        <junit.version>4.8.1</junit.version>
        <hadoop.version>2.7.2</hadoop.version>
        <spark-version>1.3.0</spark-version>
    </properties>

    <dependencies>
        <!-- JUnit -->

        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>

        <!-- Hadoop -->
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-hdfs</artifactId>
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</group
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-client</artifactId
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</group
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>

        <!-- Apache Spark -->
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.11</artifact
            <version>${spark-version}</version>
            <exclusions>
                <exclusion>
```

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

800+ Java Interview Q&As

300+ Core Java Q&As

300+ Enterprise Java Q&As

150+ Java Frameworks Q&As

120+ Companion Tech Q&As

Tutorials - Enterprise Java

```
                    <groupId>javax.servlet</groupI
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>

</project>

```

Step 2: The **report.xml** file under “src/main/resources/data”.

```

<?xml version="1.0" encoding="UTF-8"?>
<transactionReports xmlns="http://mytutorial.com/
    <transactionReport>
        <report>
            <reportNumber>9999</reportNumber>
            <createdDatetime>2015-06-15T11:29:52+
            <processedDatetime>2015-06-15T11:29:5
            <reportStatusCode>Active</reportStatu
        </report>
    </transactionReport>
</transactionReports>

```

Step 3: The avro schema file “**trans-report.avsc**” under “src/main/resources/schema”.

```

{"namespace": "mytutorial.com.report",
 "type": "record",
 "name": "ReportAvro",
 "fields": [
     {"name": "reportNumber", "type": "string"},
     {"name": "createdDatetime", "type": "string"
     {"name": "processedDatetime", "type": "strin
     {"name": "reportStatusCode", "type": "string
 ]
}

```

Step 4: The **Report.java** to map XML contents to POJO (Plain Old Java Object).

```

package com.mytutorial.pojo;

public class Report {

    private String reportNumber;
    private String createdDatetime;
    private String processedDatetime;
    private String reportStatusCode;

    public Report(String reportNumber, String cre
        this.reportNumber = reportNumber;
        this.createdDatetime = createdDatetime;
        this.processedDatetime = processedDatetim
        this.reportStatusCode = reportStatusCode;
    }

    public String getReportNumber() {
        return reportNumber;
    }

    public void setReportNumber(String reportNumb
        this.reportNumber = reportNumber;
    }

    public String getCreatedDatetime() {
        return createdDatetime;
    }

    public void setCreatedDatetime(String created
        this.createdDatetime = createdDatetime;
    }

    public String getProcessedDatetime() {
        return processedDatetime;
    }

    public void setProcessedDatetime(String proces
        this.processedDatetime = processedDatetim
    }

    public String getReportStatusCode() {
        return reportStatusCode;
    }

    public void setReportStatusCode(String report
        this.reportStatusCode = reportStatusCode;
    }

    @Override
    public String toString() {
        return "Report [reportNumber=" + reportNu
                + processedDatetime + ", reportSt
    }
}

```

Step 5: Finally, the stand-alone “ConvertXmlToAvroFile.java” to convert an XML to POJO, and then to AVRO “GenericRecord”, and then to an AVRO file “**data/report.avro**“.

```

package com.mytutorial;

import java.io.File;
import java.io.IOException;
import java.io.StringReader;
import java.net.URL;
import java.util.Iterator;

import javax.xml.namespace.NamespaceContext;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.apache.avro.Schema;
import org.apache.avro.file.DataFileReader;
import org.apache.avro.file.DataFileWriter;
import org.apache.avro.generic.GenericData;
import org.apache.avro.generic.GenericDatumReade
import org.apache.avro.generic.GenericDatumWrite
import org.apache.avro.generic.GenericRecord;
import org.apache.avro.io.DatumReader;
import org.apache.avro.io.DatumWriter;
import org.apache.commons.io.FileUtils;
import org.w3c.dom.Node;
import org.xml.sax.InputSource;

import com.mytutorial.pojo.Report;

public class ConvertXmlToAvroFile {

    private static final String FILE_IN_PATH = "
    private static final String FILE_OUT_PATH =
    private static final String AVRO_SCHEMA_FILE

    public static void main(String[] args) throw
        URL resource = ConvertXmlToSequence.clas

        File inputFile = new File(resource.getPa
        File outputFile = new File(resource.getP
        File avroSchemaFile = new File(resource.

        Report report = convertXmlToPojo(inputFi

        write(avroSchemaFile, outputFile, report)

        read(avroSchemaFile, outputFile); // rea
    }

    // write the pojo "Report" to avro file
    public static void write(File avroSchemaFile
        Schema avroSchema = new Schema.Parser().
        GenericRecord myrecord = new GenericData
        myrecord.put("reportNumber", report.getR
        myrecord.put("createdDatetime", report.g
        myrecord.put("processedDatetime", report
        myrecord.put("reportStatusCode", report.

        DatumWriter<GenericRecord> datumWriter =
        DataFileWriter<GenericRecord> writer = n
        writer.create(avroSchema, outputFile);
        writer.append(myrecord);
        writer.close();
    }

    //read an avro file
    private static void read(File avroSchemaFile
        Schema avroSchema = new Schema.Parser().

        DatumReader<GenericRecord> datumReader =
        DataFileReader<GenericRecord> reader = n
        GenericRecord record = reader.next();
        System.out.println(record);
        reader.close();
    }

    // Xpath to read XML & convert it to a pojo
    private static Report convertXmlToPojo(File
        XPath xPath = XPathFactory.newInstance()
        // namespace
        NamespaceContext ctx = new NamespaceCont
            public String getNamespaceURI(String
                return prefix.equals("urn") ? "h
            }

            public Iterator<String> getPrefixes(
                return null;
            }

            public String getPrefix(String uri)
                return null;
            }
        };

        xPath.setNamespaceContext(ctx);
        String str = FileUtils.readFileToString(
        StringReader sr = new StringReader(str);
        InputSource source = new InputSource(sr)

        // get the DOM
        Node root = (Node) xPath.evaluate("/", s

        // use the DOM
        String reportNumber = xPath.evaluate("//
        String createdDatetime = xPath.evaluate(
        String processedDatetime = xPath.evaluat
        String reportStatusCode = xPath.evaluate

        return new Report(reportNumber, createdD
    }
}


```

‹ XML Parsing with JAXB implementation called MOXy

03: Q16 – Q26 Hadoop MapReduce interview questions & answers ›

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. Privacy Policy

© 2022 java-success.com