

# ADVANCED PROGRAMMING

## Lab Sheet 16 – Data Transformation

### INTRODUCTION

---

Although the previous session's exercises gave you some experience of using XML and JSON as data interchange formats, you may not yet have a sense as to just how commonly you might find yourself working with XML and JSON data in industry. This lab task simulates the sort of work you might find yourself doing when writing back-end systems, on some real-world data from a government data feed.

The Department for Infrastructure (Roads), in Northern Ireland publishes a live data feed (details [here](#)) of notable incidents affecting journeys in the country. The data takes the form of an XML-based RSS feed, which browsers and other software can automatically subscribe to and notify users of updates. The (British) Highways Agency used to publish a similar XML-based data feed of incidents (details [here](#)) but that system appears to no longer be maintained.

I have captured the XML produced by the Transport NI system during an evening rush hour, and you are going to *transform* the data, outputting it into a simplified format with less information. Tasks such as this (e.g. take input data, pick out what you need, output resultant data) are very common in industry, particularly when working with *web services*, which you'll learn about later on in Advanced Programming.

### TASK 1: READING THE DATA

---

Download the captured XML data file from moodle, and add it to a fresh eclipse project. I would recommend creating an *input* and *output* folder in your project, to keep things nice and tidy. You will need to use some similar techniques to this week's lab to read the XML data, the format of which is illustrated in Fig. 1., below.

```

<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <title>Trafficwatch NI Incidents</title>
    <link>http://www.trafficwatchni.com/</link>
    <description>A listing of incidents in Northern Ireland.</description>
    <language>en-gb</language>
    <pubDate>Tue, 05 Oct 2021 14:33:00 +0000</pubDate>
    <atom:link href="http://rss.trafficwatchni.com/trafficwatchni_incidents" type="application/rss+xml" />
    <item>
      <title>Belfast: Signals OFF Ormeau Rd / Cromac St / Ormeau Av, , Ormeau</title>
      <description><![CDATA[
        In Belfast: the traffic Signals Ormeau Road at the junction with Cromac St / Ormeau Av, , Ormeau
        Road users are asked to approach this junction with extra care and be prepared for delays.
      ]]></description>
      <pubDate>Tue, 05 Oct 2021 08:13:32 +0000</pubDate>
      <guid isPermaLink="false">TWNIIncident1971831</guid>
    </item>
  </channel>
</rss>

```

Figure 1: Incident Feed XML Data Structure

Your task is to obtain four key pieces of information about each incident in the XML data, and save them to a simpler output format, also XML-based. The four pieces of information required are:

1. Date of Incident
2. Time of Incident
3. Incident Title
4. Incident Description (without CDATA tags)

The Simpler XML structure, to which you should output the data, is depicted in Fig. 2., below.

```

<?xml version="1.0" encoding="UTF-8"?>
<incidents>
  <incident>
    <date>2021-10-05</date>
    <time>05:58</time>
    <title>Something terrible</title>
    <description>Something terrible has happened. Expect long delays</description>
  </incident>
  <incident>
    <date>2021-10-05</date>
    <time>07:12</time>
    <title>Something else</title>
    <description>This is probably not great either</description>
  </incident>
</incidents>

```

Figure 2: Required Output Data Format

Your task is to create a program capable of reading the data from a file in the input format, located in your input directory, and writing data in the output format, to a file located in your output directory.

## TASK 2: CONVERTING TO JSON

---

One particularly common data transformation task is to convert data from an XML data source into a JSON-based format (e.g. to store in MongoDB). For this task, write a second program capable of converting the data to the format depicted in Fig. 3., below, and writing it to a file in your eclipse project's output directory.

```
[
  {
    "date": "2021-10-05",
    "time": "05:58",
    "title": "Something terrible",
    "description": "Something terrible has happened. Expect long delays"
  },
  {
    "date": "2021-10-05",
    "time": "07:12",
    "title": "Something else",
    "description": "This is probably not great either"
  }
]
```

Figure 3: Required JSON Output Format