

Narzędzie wstępnej weryfikacji pliku XML

1 Wstęp

Celem notatki jest opis i prezentacja prostego sposobu na sprawdzenie, czy plik importowy XML jest zgodny ze schematem XSD. Nie jest to oczywiście kompletna weryfikacja poprawności danych wysyłanych do systemu POL-on, która odbywa się w procesie importu. Dotarły do nas jednak sygnały, że istnieje potrzeba stworzenia i opisanie sposobu wstępnej weryfikacji wysyłanego pliku.

Próby wykonania tej operacji za pomocą ogólnie dostępnych narzędzi on-line mogą sprawiać dużo problemów. Większość dostępnych witryn tego typu zakłada, że istnieje jeden plik XSD oraz jeden plik XML, który poddajemy weryfikacji. W naszym przypadku XSD składa się z wielu powiązanych ze sobą plików, nie wystarczy użyć np. pliku *student.xsd*, by zweryfikować poprawność wysyłanego do systemu pliku importowego zawierającego studentów.

2 Weryfikacja

Poniżej prezentujemy bardzo prosty, ale skuteczny sposób weryfikacji plików importowych. Służy do tego program *xsd_valid.jar*, który dostępny jest na stronie pomocy technicznej systemu POL-on. Poniżej krótka instrukcja pokazująca, jak z niego skorzystać.

Załóżmy, że istnieje plik *stud01.xml* zawierający dane studentów, który chcemy zweryfikować za pomocą schematu XSD. W tym celu tworzymy dowolny folder, np. *C:\test*. (Oczywiście lokalizacja i nazwy katalogów mogą być inne).

Wewnątrz *C:\test* tworzymy dwa dodatkowe katalogi: *xml* i *xsd*. W katalogu *xml* umieszczamy plik, który ma zostać poddany weryfikacji, np. *stud01.xml*. Do katalogu *xsd* kopiujemy importowy schemat XSD. W samym katalogu *C:\test* należy umieścić program *xsd_valid.jar*.

Po wykonaniu powyższych operacji powinniśmy mieć następującą strukturę plików i katalogów:

C:\test

C:\test\xsd_valid.jar

C:\test\xml\stud01.xml

C:\test\xsd\plikImp.xsd

C:\test\xsd\element\doktorant.xsd

C:\test\xsd\element\osoba.xsd

C:\test\xsd\element\pracownik.xsd

C:\test\xsd\element\student.xsd

C:\test\xsd\wspolne\wspolne.xsd

W tym momencie można już uruchomić program sprawdzający, czy plik XML jest zgodny ze schematem XSD. Do tego służy polecenie (uruchamiane z poziomu wiersza poleceń):

```
java -jar xsd_valid.jar <plik xsd> <plik xml>
```

czyli w naszym przypadku:

```
java -jar xsd_valid.jar .\xsd\plikImp.xsd .\xml\stud01.xml
```

Jeśli plik jest poprawny (z punktu widzenia schematu XSD), to powinniśmy zobaczyć następujący komunikat:

```
c:\test>java -jar xsd_valid.jar .\xsd\plikImp.xsd .\xml\stud01.xml
```

Plik .\xml\stud01.xml OK

Jeśli w pliku XML jest natomiast błąd, zostaniemy o tym poinformowani:

```
c:\test>java -jar xsd_valid.jar .\xsd\plikImp.xsd .\xml\stud01.xml
```

cvc-complex-type.2.4.a: Invalid content was found starting with element 'cudzoziemiec'. One of '{imie2, prefixNazwiska, nazwisko}' is expected.

W przykładzie brakło wymaganego pola <nazwisko> w danych osobowych studenta.

```
<osoba>
  <imie1>Jan</imie1>
  <cudzoziemiec>N</cudzoziemiec>
  <plec>M</plec>
  <rokUrodzenia>1982</rokUrodzenia>
  <pesel>82050439243</pesel>
</osoba>
```

UWAGA:

xsd_valid.jar to program wymagający zainstalowania środowiska uruchomieniowego Java.

Kod źródłowy xsd_valid.jar

```
package xsdvalid;

import java.io.File;
import java.io.IOException;
import javax.xml.XMLConstants;
import javax.xml.transform.stream.StreamSource;
import javax.xml.validation.Schema;
import javax.xml.validation.SchemaFactory;
import javax.xml.validation.Validator;
import org.xml.sax.SAXException;

public class XsdValid {

    public static void main(String[] args) {
        validate(args[0], args[1]);
    }

    static void validate(final String xsdFile, final String xmlFile) {
        SchemaFactory factory;
        Validator validator;
        Schema schema;
        try {
            factory = SchemaFactory.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI);
            schema = factory.newSchema(new File(xsdFile));
            validator = schema.newValidator();
            validator.validate(new StreamSource(new File(xmlFile)));

            System.out.println("Plik " + xmlFile + " OK");
        } catch (IOException | SAXException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```