

IMJV Xml validatie module

Inleiding

De IMJV xml validatie module bestaat uit een command line programma dat xml bestanden valideert ten opzichte van IMJV [xsd schema](#). De validatie module laat toe om zelf gegenereerde xml bestanden te valideren ten opzichte van een IMJV xsd schema. De validatie module rapporteert alle fouten in de xml bestanden.

Dit document bevat een handleiding hoe we de IMJV validatie module moeten gebruiken.

Voorwaarden

Om met de validatie module te werken zijn er twee zaken vereist: een correct Java 6 installatie en de validatie module zelf. De validatie module is beschikbaar op onderstaande link (gelieve "x.x.x" steeds te vervangen door de meest recente versie van de validator):

<https://www.milieuinfo.be/repository/content/repositories/public/be/milieuinfo/imjv/xmlvalidation/validator/x.x.x/validator-x.x.x.jar>

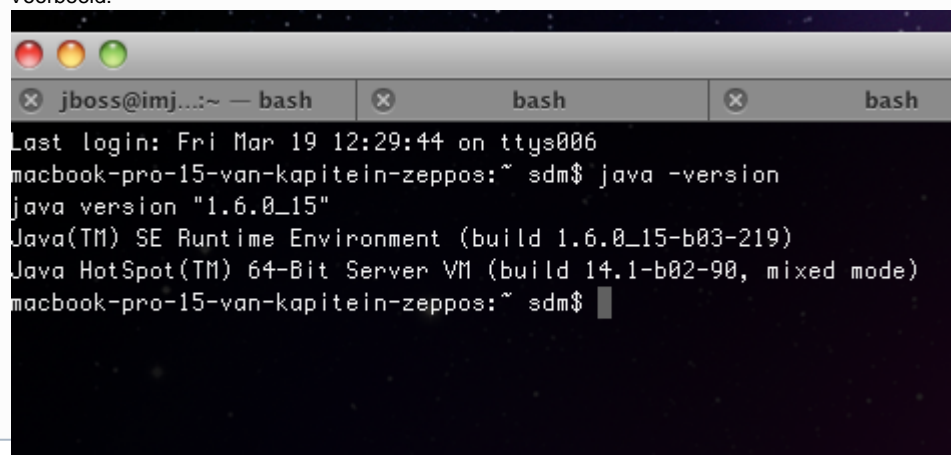


Java installatie

De Java omgeving kan je afhalen op de website <http://www.java.com/en/download/index.jsp>. Volg de stappen op deze website om de Java omgeving correct te installeren.

Open na de installatie de command line en typ "java -version". Bij een correcte installatie moet je versie 1.6 of hoger te zien krijgen.

Voorbeeld:



```
macbook-pro-15-van-kapitein-zeppos:~ sdm$ java -version
java version "1.6.0_15"
Java(TM) SE Runtime Environment (build 1.6.0_15-b03-219)
Java HotSpot(TM) 64-Bit Server VM (build 14.1-b02-90, mixed mode)
macbook-pro-15-van-kapitein-zeppos:~ sdm$
```

Handleiding

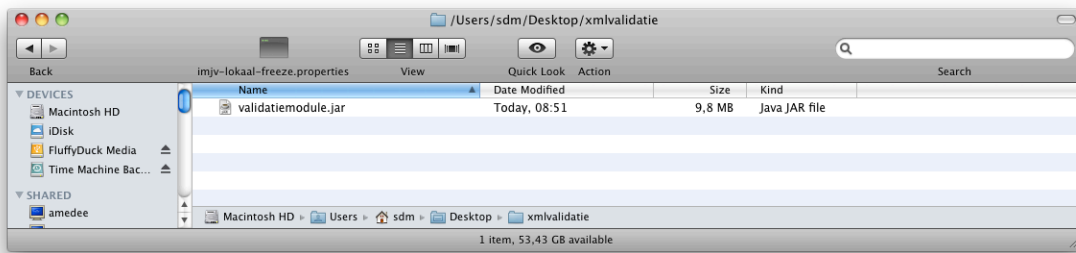
De validatie module is een command line Java module die drie input parameters verwacht:

1. folder met xml bestanden
2. xsd bestand
3. Java klasse

We beschrijven nu de verschillende stappen die nodig zijn om de validatie van xml bestanden uit te voeren met behulp van de validatie module.

Stap 1: Locatie van de validatie module

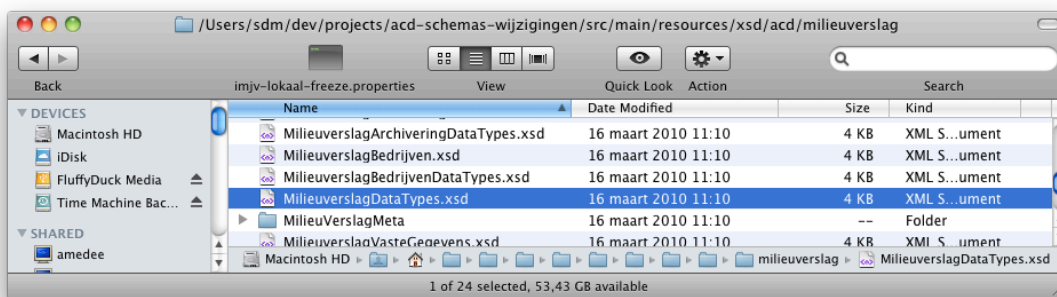
1. Klik op onderstaande link om de validatie module af te halen (gelieve "x.x.x" steeds te vervangen door de meest recente versie van de validator):
<https://www.milieuinfo.be/repository/content/repositories/public/be/milieuinfo/imjv/xmlvalidation/validator/x.x.x/validator-x.x.x.jar>
2. Plaats de validatie module in een zelf gekozen folder.



In dit voorbeeld staat de validatie module in de `/Users/sdm/Desktop/xmlvalidatie` folder.

Stap 2: Locatie van het xsd schema

1. Klik op onderstaande link om de juiste xsd schema's af te halen (gelieve "x.x.x" steeds te vervangen door de meest recente versie van de schemas)
 2. [Download hier de IMJV Schemas](#)
 3. Verander de extentie van het bestand van .jar naar .zip en pak dit bestand uit naar een zelf gekozen folder.
- 1.

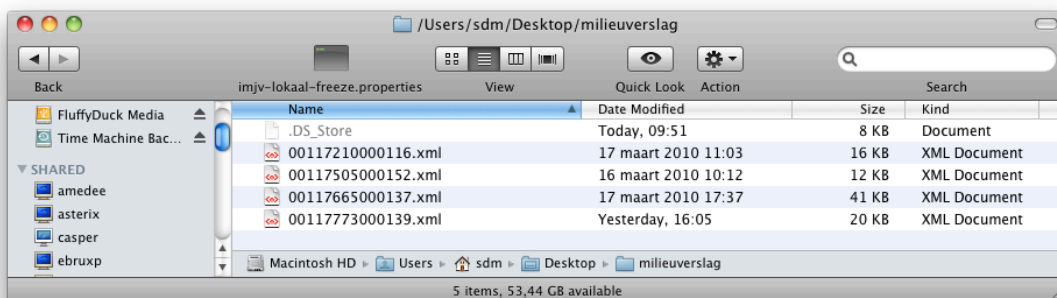


In dit voorbeeld staat het xsd schema in de folder `/Users/sdm/dev/projects/acd-schemas-wijzigingen/src/main/resources/xsd/acd/milieuverslag`

Stap 3: Folder met xml bestanden

Om de validate module te gebruiken hebben we eerst een aantal xml bestanden nodig. Voor deze handleiding beschouwen we dat de xml bestanden gemaakt zijn op basis van de MilieuverslagBedrijven.xsd. Dit xsd schema maakt deel uit van de schema's die we in de vorige stap hebben afgehaald.

De xml bestanden moeten in één specifieke folder staan:



In dit geval staan er 4 xml bestanden in de folder `/Users/sdm/Desktop/milieuverslag`.

Stap 4: Specificatie van de Java Klasse

De IMJV validatiemodule vereist nog een specificatie van een Java klasse als argument. Deze klasse beschrijft de binding tussen de Java omgeving en het xsd bestand.

Het valideren van de xml bestanden voor bedrijven gebeurt ten opzichte van het MilieuverslagBedrijven.xsd bestand. De overeenkomstige Java klasse van dit bestand is: `be.milieuinfo.schemas.milieuverslag.bedrijven.MilieuverslagBedrijvenType`

Stap 5: Uitvoeren van de validatie module

De validatie module kunnen we als volgt uitvoeren:

Uivoeren validatie module

```
java -jar validator-x.x.x.jar /Users/sdm/Desktop/milieuverslag  
/acd-schemas/src/main/resources/xsd/acd/milieuverslag/MilieuverslagBedrijven.xsd  
be.milieuinfo.schemas.milieuverslag.bedrijven.MilieuverslagBedrijvenType
```

De eerste parameter is de folder met de xml bestanden. De tweede parameter bevat de locatie van het xsd bestand. De laatste parameter bevat de java klasse.

Na het uitvoeren van de validatie module krijgen we hetvolgende resultaat:

Resultaat uitvoering

```
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117210000116.xml  
Start unmarshall and validate  
Start unmarshalling  
Finish unmarshalling  
*** Uploading took: 1923 ms  
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117505000152.xml  
Start unmarshall and validate  
Start unmarshalling  
Finish unmarshalling  
*** Uploading took: 1118 ms  
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117665000137.xml  
Start unmarshall and validate  
Start unmarshalling  
Finish unmarshalling  
*** Uploading took: 848 ms  
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117773000139.xml  
Start unmarshall and validate  
Start unmarshalling  
Finish unmarshalling  
*** Uploading took: 796 ms  
Finished validating.
```

In dit voorbeeld zijn alle xml bestanden correct.

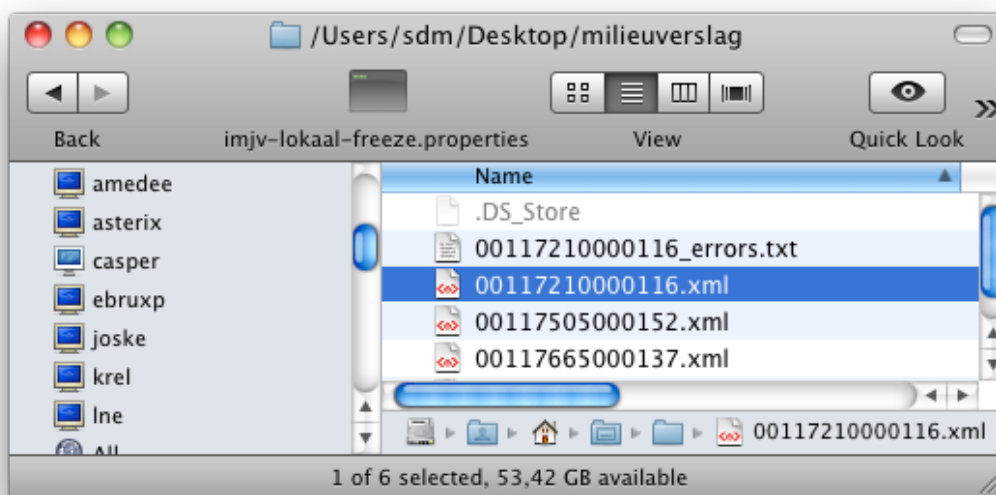
Indien we bestand 00117210000116.xml aanpassen en we vervangen het cbbnummer van de exploitant door een foutief nummer, dan krijgen we:

Output met validatie fouten


```
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117210000116.xml
Start unmarshall and validate
Start unmarshalling
Finish unmarshalling
*** Uploading took: 1674 ms
cvc-type.3.1.3: The value '123546' of element 'CBBExploitantNumber' is not valid.
cvc-pattern-valid: Value '123546' is not facet-valid with respect to pattern
'0[0-9]{13}' for type 'CBBnummerType'.
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117505000152.xml
Start unmarshall and validate
Start unmarshalling
Finish unmarshalling
*** Uploading took: 1094 ms
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117665000137.xml
Start unmarshall and validate
Start unmarshalling
Finish unmarshalling
*** Uploading took: 849 ms
*** Start validating file: /Users/sdm/Desktop/milieuverslag/00117773000139.xml
Start unmarshall and validate
Start unmarshalling
Finish unmarshalling
*** Uploading took: 755 ms
Finished validating.
```

In de output kunnen we reeds vaststellen dat het cbnummer van de exploitant niet correct is.

De IMJV validatiemodule groepeerde de validatiefouten in een error bestand en plaatst dit naast het xml bestand.



In dit voorbeeld staat er een error bestand naast het xml bestand dat niet valideert.

 De IMJV validatiemodule verwijdert zelf de error.txt bestanden bij een nieuwe uitvoering. Manueel verwijderen van deze bestanden is overbodig.

Opmerkingen

1. Tijdens de uitvoering van de IMJV validatiemodule krijgen we volgende warning te zien: *Boot: Warning: Unable to locate main/main.jar in the JAR file validator-x.x.x.jar*. Deze warning wordt veroorzaakt door de gebruikte technologie en mag genegeerd

worden.

2. Indien de IMJV validatiemodule wordt uitgevoerd met Java 5, dan krijgen we volgende error:

```
Caused by: java.lang.LinkageError: JAXB 2.0 API jar is being loaded (from
onejar:javax.xml.bind.annotation.XmlSchema.class), but this RI (from
onejar:com.sun.xml.bind.v2.model.impl.ModelBuilder.class) requires JAXB 2.1 API
jar.
  at com.sun.xml.bind.v2.model.impl.ModelBuilder.<clinit>(ModelBuilder.java:173)
  at
com.sun.xml.bind.v2.runtime.JAXBContextImpl.getTypeInfoSet(JAXBContextImpl.java:431
)
  at com.sun.xml.bind.v2.runtime.JAXBContextImpl.<init>(JAXBContextImpl.java:288)
  at
com.sun.xml.bind.v2.runtime.JAXBContextImpl$JAXBContextBuilder.build(JAXBContextImp
l.java:1111)
  at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:154)
  at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:121)
  at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:202)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
  at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:2
5)
  at java.lang.reflect.Method.invoke(Method.java:592)
  at javax.xml.bind.ContextFinder.newInstance(ContextFinder.java:132)
  at javax.xml.bind.ContextFinder.find(ContextFinder.java:286)
  at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:372)
  at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:337)
  at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:244)
  at
be.milieuinfo.ValidationImportService.setupUnmarshaller(ValidationImportService.jav
a:149)
  at
be.milieuinfo.ValidationImportService.unmarshallAndImport(ValidationImportService.j
ava:41)
  at be.milieuinfo.ValidateXml.upload(ValidateXml.java:32)
  at be.milieuinfo.ValidateXml.main(ValidateXml.java:102)
  ... 6 more
https://www.milieuinfo.be/repository/content/repositories/public/be/milieuinfo/imjv
/xmlvalidation/validator/x.x.x/validator-x.x.x.jar
```