golden-erlsom

Bartosz Pietrzyk Ludwik Ciechański

Akademia Górniczo Hutnicza im. Stanisława Staszica w Krakowie

> bpietrzy@student.agh.edu.pl lciechan@student.agh.edu.pl

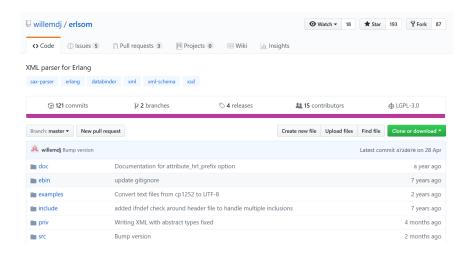
> > 19 czerwca 2018

Plan prezentacji

- erlsom
 - github
 - możliwości
 - dokumentacja
 - instalacja
 - interfejs
- 2 golden-erlsom
 - github i api
 - przykłady i testy
- g podsumowanie

Gdzie znajdę erlsoma?

• https://github.com/willemdj/erlsom



Co oferuje erlsom?

- Erlsom XML Parser for Erlang
- Dostępne są trzy tryby:
 - DOM Mode model obiektowy
 - 2 SAX Mode model sekwencyjny
 - Oata Binder Mode możliwość zmapowania XML-a do erlangowych rekordów
- Projekt jest na bieżąco rozwijany, posiada dobrą dokumentację

Dokumentacja

- tu: http://erlsom.sourceforge.net/erlsom.htm
- lub tu:

https://github.com/willemdj/erlsom/blob/master/doc/reference.md

Data Binder Mode

In this mode, Erlsom parses XML documents that are associated with an XSD (or Schema), It checks whether the XML document conforms to the Schema, and It translates the document to an Erlang structure that is based on the types defined in the Schema. This section tries to explain the relation between the Schema and the Erlang data structure that is produced by Erlsom.

First a quick example using the same XML that was used for the other modes. Before we can parse the document we need to 'compile' the XML Schema (similar to how you might compile a regular expression).

```
10> {ok, Model} = erlsom:compile_xsd_file("minimal.xsd").
```

Now you can use this compiled model:

```
11> {ok, Result, _} = erlsom:scan(Xml, Model).
{ok, {foo type, [], "baz", ["x", "y"]}, "\r\n"}
```

Assuming that you have defined a suitable record #foo_type{} (erlsom:write_xsd_hrl_file() can do it for you), you can use in your program (won't work in the shell):

```
BarValues = Result#foo_type.bar,
AttrValue = Result#foo type.attr,
```

Nice and compact, as you see, but it may need more explanation. I will use a more complex example from the XML Schema Primer (XML Schema Part 0: Primer Second Edition) [Primer]. It can be found here: http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/11. Sections that have been copied from this document are contained in a blue box.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
    <xsd:documentation xml:lang="en">
```

Instalacja

- użyć rebara
- "ręcznie"
 - pobrać repo, zapisać w ROOT/lib/erlsom-1.2.1/src (ROOT to lokalizacja Erlanga, np. C:/Program Files/erl5.6.1)
 - ② odpalić Erlang shell i wykonać: 1> cd('../lib/erlsom-1.2.1/src'). C:/Program Files/erl5.6.1/lib/erlsom-1.2.1/src
 - ok
 2> make:all([outdir, "../ebin"]).

Interfejs

XSD - kompilowanie, rekordy

```
compile_xsd/1
compile_xsd/2
compile_xsd_file/1
compile_xsd_file/2
write_xsd_hrl_file/3
```

XML - parsowanie, zapisywanie

```
scan/3
scan_file/2
write/2
write/3
parse_sax/4
simple_form/1
simple_form/2
```

scan/2

erlsom_lib

```
erlsom_lib:toUnicode/1
erlsom_lib:find_xsd/4
erlsom_lib:detect_encoding/1
```

erlsom_ucs

```
erlsom_ucs:from_utf8/1
erlsom_ucs:from_utf16le
erlsom_ucs:from_utf16be
```

golden-erlsom

- repozytorium: https://github.com/Veyronix/golden-erlsom
- do testowania możliwości erlsoma użyliśmy http://api.nbp.pl/



Przykłady użycia erlsoma

interfejs użytkowy

```
gold_last_30()
gbp_last_10()
n_last_rates_of_currency(N, Currency)
when_currency_was_the_most_valuable_in_n_days(N, Currency)
```

funkcje testowe

```
test_of_scan_file()
test_of_write()
test_of_write_xsd_hrl_file()
test_of_parse_sax()
test_of_simple_form()
test_of_toUnicode()
test_of_detect_encoding()
test_of_from_utf16le()
```

Podsumowanie

- Erlsom skutecznie parsuje XML-a :)
- Jest możliwość wyboru modelu SAX / DOM / Data Binder
- Używając tego ostatniego można przenieść dane z XML-a do erlangowych rekordów
- Kilka drobnych wad: brak walidacji danych, brak wsparcia dla niektórych atrybutów standardu

Dziękujemy za uwagę!

