

# golden-ersom

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

## 1 erlsom

- github
- możliwości
- dokumentacja
- instalacja
- interfejs

## 2 golden-erlsom

- github i api
- przykłady i testy

## 3 podsumowanie

# Gdzie znaleźć erlsoma?

- <https://github.com/willemdj/erlsom>

willemdj / erlsom

Watch

18

Star

193

Fork

87

<> Code

Issues

5

Pull requests

3

Projects

0

Wiki

Insights

XML parser for Erlang

sax-parser

erlang

databinder

xml

xml-schema

xsd

121 commits

2 branches

4 releases

15 contributors

LGPL-3.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

willemdj Bump version

Latest commit 472d070 on 28 Apr

doc	Documentation for attribute_hrl_prefix option	a year ago
ebin	update gitignore	7 years ago
examples	Convert text files from cp1252 to UTF-8	2 years ago
include	added ifndef check around header file to handle multiple inclusions	7 years ago
priv	Writing XML with abstract types fixed	4 months ago
src	Bump version	2 months ago

# Co oferuje erlsom?

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

```
<?xml version="1.0" encoding="UTF-8" ?>
- <employees>
-   <emp id="S001">
      <name>ABC</name>
      <salary>5000</salary>
    </emp>
-   <emp id="S002">
      <name>PQR</name>
      <salary>7000</salary>
    </emp>
-   <emp id="S003">
      <name>XYZ</name>
      <salary>9000</salary>
    </emp>
  </employees>
```

- 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").  
{ok, (model, [{typ...
```

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/><sup>[1]</sup>. 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">
```

- użyć rebara
- "ręcznie"
  - 1 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/2  
scan/3  
scan_file/2  
write/2  
write/3  
parse_sax/4  
simple_form/1  
simple_form/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
```

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

**NBP** Narodowy Bank Polski



## NBP Web API

English version

### Pobieranie kursów walut oraz cen złota w formacie XML oraz JSON

Serwis [api.nbp.pl](http://api.nbp.pl) udostępnia publiczne Web API umożliwiające klientom HTTP wykonywanie zapytań na poniższych zbiorach danych publikowanych przez serwis NBP.PL:

- aktualne oraz archiwalne kursy walut obcych:
  - Tabela A kursów średnich walut obcych,
  - Tabela B kursów średnich walut obcych,
  - Tabela C kursów kupna i sprzedaży walut obcych;
- aktualne oraz archiwalne ceny złota wyliczone w NBP.

Komunikacja z serwisem polega na wysłaniu odpowiednio sparametryzowanego żądania HTTP GET na adres bazowy <http://api.nbp.pl/api/>.

- Informacje ogólne
- Opis funkcji API dotyczących kursów walut
  - Parametry zapytań
  - Zapytania o kompletne tabele
  - Zapytania o pojedynczą walutę
  - Opis parametrów odpowiedzi
- Opis funkcji API dotyczących zapytań o ceny złota
  - Parametry zapytań
  - Zapytania o ceny złota
  - Opis parametrów odpowiedzi
- Przykłady zapytań
- Komunikaty błędów

## Instrukcja użytkownika



# 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()
```

- 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

