

Projekt końcowy z SKiSR

- **Prosty server SOAP - wyszukiwanie w Google (Ruby)**
- **Prosty server SOAP - wyszukiwanie w Google (PHP)**

Oboza Jakub, Ormicki Darek, 3 gr., II rok IS

SOAP (ang. *Simple Object Access Protocol*) jest to protokół zdalnego wywoływania dostępu do obiektów, wykorzystujący XML do kodowania wywołań i najczęściej HTTP do ich przenoszenia. Dokument SOAP składa się z trzech części:

- otoczki (envelope) która określa szkielet opisujący co znajduje się w komunikacie i jak go przetwarzać,
- zbioru reguł kodujących potrzebnych do rozszyfrowania typów danych (również złożonych) zdefiniowanych wewnątrz aplikacji,
- reguł dotyczących wywoływania zdalnych metod i odczytu odpowiedzi.

SOAP jest standardem W3C.

Niemniej SOAP umożliwia dynamiczne rozpoznawanie interfejsów obiektów na serwerze. Jest to dokonywane za pomocą WSDL (ang. *Web Services Description Language*), będącego dokumentem XML, który opisuje typy, metody i mechanizmy dostępu do interfejsów usług sieciowych.

W naszych programach, których zadanie polega na skorzystaniu z wyszukiwarki Google musimy użyć Google Web API.

Dostarcza on funkcję `doGoogleSearch` , której opis i wywołanie znajdują się poniżej.

`doGoogleSearch(key, q, start, maxResult, filter, restrict, safeSearch, lr, ie, oe)`

`key` - klucz developerski

`q` - szukane słowo / fraza

`start` - numer pierwszego zwróconego wyniku

`maxResults` - ilość wszystkich wyników (max. 10)

`filter` - filtr wyszukiwania

`restrict` - ustawia symbol kraju

`safeSearch` - zabezpieczenie przed stronami z treścią pornograficzną

`lr` - wyszukiwanie w danym języku

`ie` i `oe` - kodowanie

1. SOAP / Ruby

źródła z pełnym kodem: *soap.rb*, *developer_key.txt*

wymagana wersja języka: $\geq 1.8.4$

W programie zostaną wykorzystane biblioteki ruby'ego: *wsdlDriver* oraz *cgi*.

Ponieważ Ruby (oraz Google) daje nam możliwość korzystania z dynamicznego rozpoznawania interfejsów obiektów na serwerze użyjemy WSDL.

Program składa się z klasy *Get_result*.

```
Class Get_result
  def initialize
  end
  def read_dev_key
  end
  def soap_inst
  end
  def conn
  end
end
```

Konstruktor inicjalizuje zmienne metody. Metoda *read_dev_key(*args)* otwiera i zwraca klucz developerski zapisany w pliku. Metoda *conn* korzysta z prywatnej metody *soap_inst* w celu utworzenia uchwytu do połączenia z wykorzystaniem WSDL:

```
def soap_inst
  return SOAP::WSDLDriverFactory.new(@@wsdl).create_rpc_driver
end
```

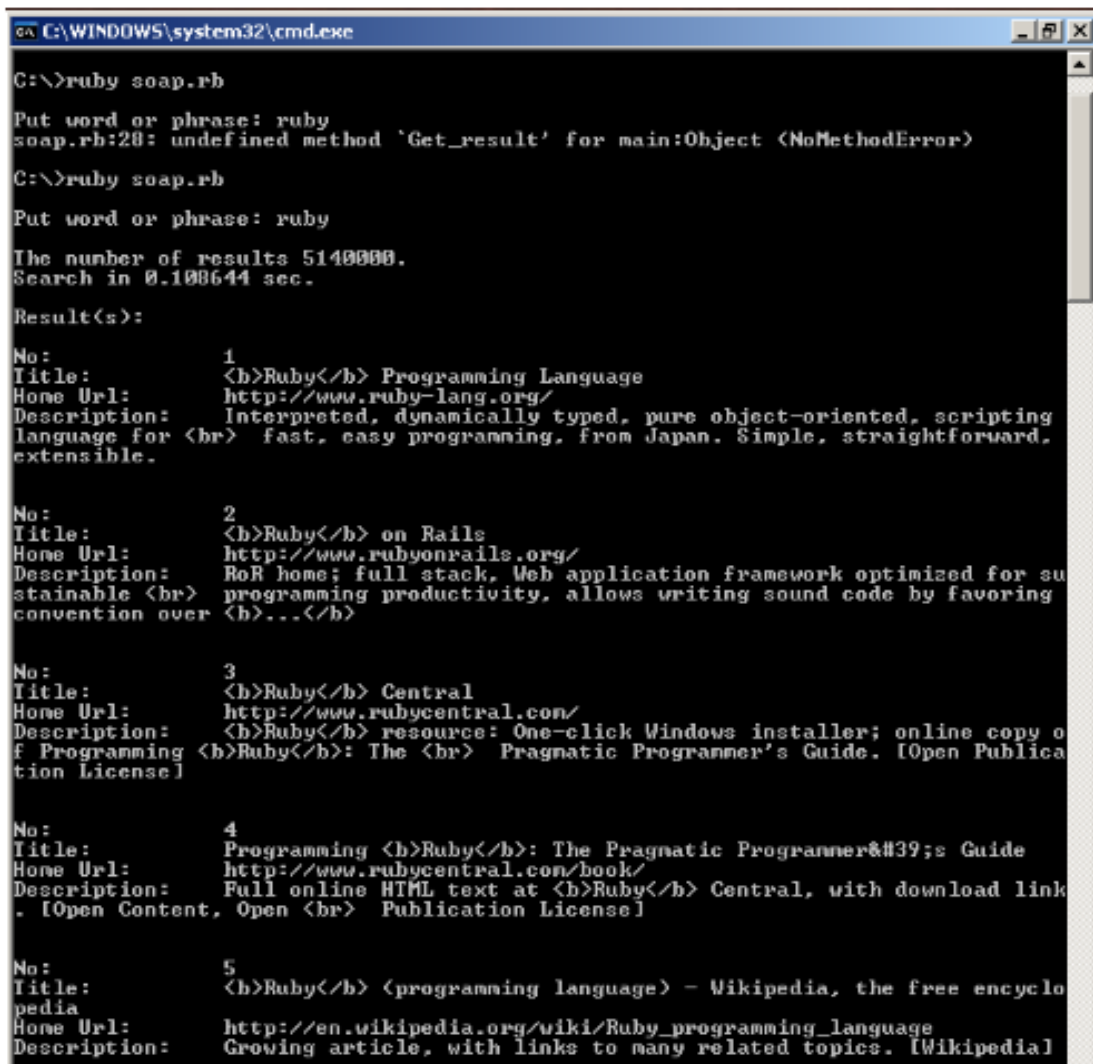
Następnie już bez zbędnego dodawania metody *doGoogleSearch* po prostu ją wykorzystujemy.

```
def conn
  @d = read_dev_key("developer_key.txt")
  return soap_inst.doGoogleSearch(@d, @q, 0, 10, false, "", false, "", "utf-8",
  "utf-8")
end
```

Wyniki zostają zwrócone w postaci tablicy asocjacyjnej i pozostaje je tylko parsować, co robimy wykorzystując iteratory języka Ruby:

```
if result.resultElements.size != 0
  printf "\nResult(s):\n\n"
  count = 0
  result.resultElements.each do
    |var|
    count += 1
    printf "No:\t\t%d\n", count
    puts "Title:\t\t" + var['title']
    puts "Home Url:\t" + var['URL']
    puts "Description:\t" + CGI.unescapeHTML(var['snippet'])
    2.times {puts}
  end
else
  puts "\nNo result."
end
```

Przykład działania:



```
C:\WINDOWS\system32\cmd.exe

C:\>ruby soap.rb

Put word or phrase: ruby
soap.rb:28: undefined method `Get_result' for main:Object (NoMethodError)

C:\>ruby soap.rb

Put word or phrase: ruby

The number of results 5140000.
Search in 0.100644 sec.

Result(s):

No:          1
Title:       <b>Ruby</b> Programming Language
Home Url:    http://www.ruby-lang.org/
Description: Interpreted, dynamically typed, pure object-oriented, scripting
language for <br> fast, easy programming, from Japan. Simple, straightforward,
extensible.

No:          2
Title:       <b>Ruby</b> on Rails
Home Url:    http://www.rubyonrails.org/
Description: RoR home; full stack, Web application framework optimized for su
stainable <br> programming productivity, allows writing sound code by favoring
convention over <b>...</b>

No:          3
Title:       <b>Ruby</b> Central
Home Url:    http://www.rubycentral.com/
Description: <b>Ruby</b> resource: One-click Windows installer; online copy o
f Programming <b>Ruby</b>: The <br> Pragmatic Programmer's Guide. [Open Publica
tion License]

No:          4
Title:       Programming <b>Ruby</b>: The Pragmatic Programmer's Guide
Home Url:    http://www.rubycentral.com/book/
Description: Full online HTML text at <b>Ruby</b> Central, with download link
. [Open Content, Open <br> Publication License]

No:          5
Title:       <b>Ruby</b> <programming language> - Wikipedia, the free encyclo
pedia
Home Url:    http://en.wikipedia.org/wiki/Ruby_programming_language
Description: Growing article, with links to many related topics. [Wikipedial
```

2. SOAP / PHP

źródło z pełnym kodem: *index.php*

wymagana wersja języka: $\geq 4.x$

W programie korzystamy z bibliotek, które udostępnia PEAR.

Program składa się z jednej klasy *SOAPGoogleAPI*.

```
Class SOAPGoogleAPI{
    function SOAPGoogleAPI();
    function search();
    function SOAPCall();
}
```

W konstruktorze ustawiamy zmienne składowe klasy, które następnie będą parametrami wywołań funkcji *doGoogleSearch*.

Metoda `SOAPCall()` jest wywoływana z metody `search()`

```
$this->soap->call( $set_api_function, $params, "urn:GoogleSearch" );
```

`$set_api_function` – wskazanie na funkcję którą będziemy zdalnie wywoływać (w naszym przypadku jest to: `doGoogleSearch`)

`$params` – parametry funkcji `doGoogleSearch`

`urn:GoogleSearch` – parametr przekazywany do biblioteki, który wymusi samodzielne dodanie tej funkcji, po czym wywoła ją.

Dalsza część kodu to obsługa błędów, parsowanie wyników i wyświetlanie ich, np:

```
if( $result ) {
    foreach( $result->resultElements as $value ) {
        echo "<a href=\"\" . $value->URL . \">\" . $value->title . "</a><br />";
        echo $value->snippet . "<br /><br /><hr><br />";
    }
}
```

Przykład działania:

Szukano dla: php

[PHP: Hypertext Preprocessor](#)

PHP is a server-side HTML embedded scripting language. It provides web developers with a full suite of tools for building dynamic websites: native APIs to ...

[PHP: Hypertext Preprocessor](#)

PHP is a widely-used general-purpose scripting language that is especially ...
For users upgrading from **PHP 5.0** and **PHP 5.1** there is an upgrading guide ...

[PHP: Downloads](#)

Regular source and binary snapshots are available from [snaps.php.net](#). ...
For downloadable **PHP** manual packages, go to the [documentation download page](#). ...

[PHP: Documentation](#)

The manual explains and documents everything about **PHP**. Listed are the various formats, such as PDF, HTML and Palm. The manual is also available in about 12 ...

Ponieważ z grudniem ubiegłego roku Google Web API przestało wspierać nowe klucze developerskie w programach posługujemy się starymi kluczami, przy użyciu których także czasami występują nieoczekiwane trudności w ustanawianiu połączenia z Google.

Cały kod programów można zobaczyć w ich plikach źródłowych, gdzie umieszczone zostały także komentarze.

Uruchomienie SOAP / Ruby:

- (w konsoli) **ruby soap.rb**

Uruchomienie SOAP / PHP:

- umieszczony został tymczasowo na http://ruby-lang.pl/soap_php