
Výměnné formáty aplikací -- JSON, YAML. Zpracování XML mimo Javu

Obsah

JSON	1
Základní charakteristika	1
Datové typy	2
Ukázka zprávy ve formátu JSON a odpovídající XML dokument	2
Zpracování formátu JSON	2
Co je YAML	3
Motivace	3
YAML Ain't Markup Language	3
Příklad	3
Možnosti celkově	4
Srovnání	4
Struktura YAML souboru	4
Dokumenty	4
Identifikace uzlů (objektů) a reference	5
Asociativní pole	5
Asociativní pole na řádku	5
Seznamy po řádcích	5
Seznamy na řádku	5
Zpracování YAML dat	6
Pořizování, úprava	6
Kombinace s XML a JSON	6
Nástroje a API v programovacích jazycích	6
XML rozhraní pro C/C++	6
Základní knihovny	6
XML rozhraní pro PHP	7
Koncepce	7
Příklad (1) - DOM	7
Příklad (2) - SAX	7
Příklad - SimpleXML	8
Další zdroje - weby	9
Další zdroje - knihy	9

JSON

Základní charakteristika

- JavaScript Object Notation
- založen na podmnožině jazyka JavaScript
- specifikace RFC 4627
- MIME type: application/json

- přípona souboru .json
- využití - serializace a posílání strukturovaných dat přes síť, např. webové služby, ...
- alternativa k XML
 - snadný převod XML-JSON

Datové typy

- čísla - celá (integer), reálná (real)
- řetězce (String)
 - Unicode znaky
 - ohraničené úvozovkami ""
- boolean - true, false
- pole (Array) - seznam hodnot oddělených čárkou, ohraničený hranatými závorkami
- Objekt - kolekce dvojic klíč:hodnota
- null

Ukázka zprávy ve formátu JSON a odpovídající XML dokument

```
{
  "jméno": "Jan",
  "příjmení": "Novák",
  "adresa":
  {
    "typ": "pracovní",
    "ulice": "Botanická",
    "čísloOrganizační": "68a",
    "město": "Brno"
  }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<osoba>
  <jméno>Jan</jméno>
  <příjmení>Novák</příjmení>
  <adresa typ="pracovní">
    <ulice>Botanická</ulice>
    <čísloOrganizační>68a</čísloOrganizační>
    <město>Brno</město>
  </adresa>
</osoba>
```

Zpracování formátu JSON

- JavaScript

- nativní podpora
- parsing
 - funkce eval - vhodné pouze při práci s daty ze spolehlivého a důvěryhodného zdroje

```
var osoba = eval( "{" + kontakt + "}" );
```
 - metoda JSON.parse() - součástí 4. vydání standardu ECMAScript
 - JSON parser - obsažen např. v moderních prohlížečích (Firefox 3.5, IE 8, Opera 10.5, Google Chrome, Safari, ...)
- PHP - podpora od verze 5.2
- Ostatní jazyky - pomocí knihoven
 - Java - org.json, Json-lib, ...
 - C - JSON_checker, JSON parser, ...
 - C++ - jsoncpp, zoolib, ...
 - další viz např. www.json.org [<http://www.json.org>]

Co je YAML

Motivace

- poptávka po lidsky čitelném, tzn. textovém formátu pro serializaci dat
- méně strojově náročné na zpracování (parsing) než XML
- vhodné i pro ruční zápis (to XML není!)
- menší paměťová režie než XML

YAML Ain't Markup Language

YAML [<http://www.yaml.org>] *není* přímou náhradou XML, není určen pro dokumenty, ale pro serializaci dat

- akronym dříve označoval "Yet Another Markup Language", podobnost s XML ale není taková, aby to bylo vhodné
- první specifikace květen 2001
- nyní (květen 2010) aktuální verze YAML 1.2 (3rd Edition) [<http://www.yaml.org/spec/1.2/spec.html>]

Příklad

Příklad asociativního pole (mapy):

- skalární hodnota (řetězec) -> skalární hodnota (číslo)

- ... a komentáře za #

```
hr: 65      # Home runs
avg: 0.278 # Batting average
rbi: 147    # Runs Batted In
```

Možnosti celkově

YAML nabízí strukturálně více možností než XML:

- snadné zobrazení datových struktur (dokumenty, seznamy, asociativné pole)
- různé možnosti pro zápis literálů (skalárních hodnot) - s nebo bez konci řádků atd.
- snadný mechanismus pro reference a odkazování
- možnost přesného označení typu dat (nebo využití autodetekce)

Srovnání

Blízkými příbuznými co do účelu použití jsou

- **JSON** (ten má navíc přímou vazbu na konkrétní pg. jazyk - JavaScript, což může být výhoda i nedostatek)
- formát e-mailových zpráv (**RFC 2822**)
- **XML**
- literálové **zápisy datových struktur** v řadě pg. jazyků: Perl, PHP, C

Struktura YAML souboru

Dokumenty

- YAML dovoluje do jednoho proudu dat umístit i více dokumentů (souborů)
- Oddělujeme je symbolem "tři znaky minus": ---
- Dokument končí buďto začátkem dalšího nebo symbolem "tři tečky": ...

```
# Ranking of 1998 home runs
---
- Mark McGwire
- Sammy Sosa
- Ken Griffey

# Team ranking
---
- Chicago Cubs
- St Louis Cardinals

---
time: 20:03:20
player: Sammy Sosa
```

```
action: strike (miss)
...
---
time: 20:03:47
player: Sammy Sosa
action: grand slam
...
```

Identifikace uzlů (objektů) a reference

- Symbol & slouží k označení a symbolickému pojmenování uzlu,
- na nějž se dále odkazuje pomocí *

```
---
hr:
  - Mark McGwire
  # Following node labeled SS
  - &SS Sammy Sosa
rbi:
  - *SS # Subsequent occurrence
  - Ken Griffey
```

Asociativní pole

Mohou mapovat jak mezi skalárními, tak *strukturovanými objekty*, pomocí dvojice ? :

```
? - Detroit Tigers
  - Chicago cubs
:
  - 2001-07-23

? [ New York Yankees,
  Atlanta Braves ]
: [ 2001-07-02, 2001-08-12,
  2001-08-14 ]
```

Asociativní pole na řádku

V jednodušších případech je úspornější zapsat asociativní pole na řádek

```
{name: John Smith, age: 33}
```

Seznamy po řádcích

Prvky seznamu mohou být na jednotlivých řádcích, uvozené znaky - a mezera

```
- item1 continuing
- item2 another item
```

Seznamy na řádku

Prvky seznamu mohou být na jednom řádku celé

```
[item1, item 2, item3 still item3]
```

Zpracování YAML dat

Pořizování, úprava

YAML je založen na prostém textovém formátu, přináší proto řadu výhod:

- nemá přísně hierarchickou strukturu (tedy žádný kořenový element jako v XML)
- prostým spojením dvou YAML dokumentů vznikne opět YAML
- dokument v YAML neobsahuje na rozdíl od JSON žádné příkazy, interpretace tedy nepřináší žádná bezpečnostní rizika

Kombinace s XML a JSON

Integrace XML fragmentů do YAML je snadná:

```
---
example: >
  HTML goes into YAML without modification
message: |
  <font name='times' size=10>
  <p><i>"Three is always greater than
    two, even for large values of two"</i>
  </p><p>  --Author Unknown  </p></font>
date: 2007-06-01
```


Nástroje a API v programovacích jazycích


Běžné programovací jazyky nabízejí knihovny pro práci s YAML:

C/C++ yaml-cpp [<http://code.google.com/p/yaml-cpp/>] (pro YAML 1.2)
Java jyaml [<http://jyaml.sourceforge.net/>]
.NET/C# Yaml Library for .NET (C#) [<http://yaml-net-parser.sourceforge.net/>]
PHP Spyc [<http://code.google.com/p/spyc/>]

XML rozhraní pro C/C++

Základní knihovny

Expat autor J. Clark, klasický parser pro zpracování řízené
událostmi (call-back), koncepčně podobné SAX, velmi rychlé,
část knihovny libexpat.so [[http://www.google.com/search?](http://www.google.com/search?q=libexpat.so)
 <http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=libexpat.so>] The Free Encyclopedia [pro Linux
MSXML knihovna pro systémy Windows, použitelná z různých programovacích jazyků

libxml2 je to knihovna pro systémy Linux/UNIX, použitelná např. z C/C+, část projektu Gnome, ale nevyžaduje jej; zvládá parsing, zápis, vyhodnocování XPath, XSLT transformace (separátně v libxslt [http://www.google.com/search?q=libxslt]  [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=libxslt])

Xerces-C++ port parseru Xerces pro C++

XML rozhraní pro PHP

Koncepce

V zásadě shodná s přístupem v Javě, existují rozhraní:

stromově orientovaná	DOM [http://php.net/manual/en/book.dom.php] pro PHP - plná škála možností, na něž jsme z DOM zvyklí (čtení, validace, zápis vč. prettyprinting, přímé programové vytváření dokumentu, jeho elementů, atd.)
proudové (typu pull)	SimpleXML [http://php.net/manual/en/book.simplexml.php] - velmi jednoduché a hojně používané, umožňuje iteraci po prvcích XML, přímé vyhodnocování XPath výrazů atd.
událostmi řízené	SAX [http://php.net/manual/en/book.xml.php] - obdobně jako v Javě, princip stejný, obsaženo ve většině PHP kompilací

Příklad (1) - DOM

Následující kód načte (analyzuje, "parsuje") XML dokument a zapíše jej (serializuje) do souboru

```
$dom = new DOMDocument();

// konfigurace pro načtení
$dom->preserveWhiteSpace = FALSE;
$dom->load('input.xml');

// konfigurace pro uložení
$dom->formatOutput = TRUE;
$dom->encoding = 'utf-8';
$dom->save('output.xml');
```

Příklad (2) - SAX

Následující kód načte (analyzuje, "parsuje") XML dokument s knihami a informace o nich vypíše (převzato z Reading and writing the XML DOM with PHP Using the DOM library, SAX parser and regular expressions, Jack Herrington, IBM 2005)

```
<?php
    $g_books = array();
    $g_elem = null;

    function startElement( $parser, $name, $attrs )
```

```
{
global $g_books, $g_elem;
if ( $name == 'BOOK' ) $g_books []= array();
$g_elem = $name;
}

function endElement( $parser, $name )
{
global $g_elem;
$g_elem = null;
}

function textData( $parser, $text )
{
global $g_books, $g_elem;
if ( $g_elem == 'AUTHOR' ||
$g_elem == 'PUBLISHER' ||
$g_elem == 'TITLE' )
{
$g_books[ count( $g_books ) - 1 ][ $g_elem ] = $text;
}
}

$parser = xml_parser_create();

xml_set_element_handler( $parser, "startElement", "endElement" );
xml_set_character_data_handler( $parser, "textData" );

$f = fopen( 'books.xml', 'r' );

while( $data = fread( $f, 4096 ) )
{
xml_parse( $parser, $data );
}

xml_parser_free( $parser );

foreach( $g_books as $book )
{
echo $book['TITLE'] . " - " . $book['AUTHOR'] . " - ";
echo $book['PUBLISHER'] . "\n";
}
?>
```

Příklad - SimpleXML

Převzato z *SimpleXML processing with PHP A markup-specific library for XML processing in PHP* by
Elliotte Rusty Harold, IBM Developerworks, 2006

```
<html xml:lang="en" lang="en">
<head>
  <title>XPath Example</title>
</head>
<body>
```



```
<?php
$rss = simplexml_load_file('http://partners.userland.com/nytRss/nytHomepage.xml')
foreach ($rss->xpath('//title') as $title) {
    echo "<h2>" . $title . "</h2>";
}
?>

</body>
</html>
```

Další zdroje - weby

- DOM Výborný úvodní článek ke XML v PHP na IBM Developerworks: Reading and writing the XML DOM with PHP [<http://www.ibm.com/developerworks/library/os-xmldomphp/>]
- SimpleXML Eliotte Rusty Harold: SimpleXML processing with PHP A markup-specific library for XML processing in PHP [<http://www.ibm.com/developerworks/library/x-simplexml.html>]

Další zdroje - knihy

- Jiří Kosek: PHP a XML Grada Publishing, 2010 - výborný, dobře čitelný, obsažný přehled jak základů XML, tak možností zpracování v PHP, XML Schema, Relax NG, XSLT, webové služby