Výměnné formáty aplikací -- JSON, YAML. Metadata, sémantický web.

Obsah

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",
      "číslo Organizační":"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 neni!)
- 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 mechanizmus 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 s*trukturovanými 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?q=libexpat.so] WIKIPEDIA [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=libexpat.so] 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] WIKIPEDIA [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 — Elliotte 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

Rámce pro metadata popisující XML a jiné datové zdroje

Rámec RDF

RDF Model a RDf Schema jsou doporučeními W3C

Specifikace a další informace pracovní skupiny - http://www.w3.org/RDF

RDF Model

RDF je obecný mechanizmus pro specifikaci metadat

je použitelný s libovolnými (i ne-digitálními) zdroji

základem modelu jsou trojice:

- zdroj (resource) - např. `http://www.fi.muni.cz/~tomp/xml`
- vlastnost (property) - např. `popis`
- hodnota (value) - např. `Domovská stránka předmětu P138 na FI MU`

Trojice je možné znázornit

- graficky,
- jako trojice `(r, p, v)` nebo
- XML syntaxí

Blíže viz

- Dobrý úvodní článek na `xml.com`: What is RDF? [http://www.xml.com/pub/a/2001/01/24/rdf.html]
- RDF Tutoriál - Zvon RDF Tutorial [http://www.zvon.org/xxl/RDFTutorial/General/book.html]
- RDF Tutoriál - W3Schools RDF Tutoriál [http://www.w3schools.com/rdf/default.asp]
- RDF Tutorial http://www710.univ-lyon1.fr/~champin/rdf-tutorial/node1.html
- Další RDF Tutorial (.ppt) [http://www.aifb.uni-karlsruhe.de/WBS/sst/Teaching/Intelligente%20System%20im%20WWW%20SS%202000/RDF-Tutorial.pdf]

RDF Schema

- Specifikuje omezení na množiny vlastností, jejich definičních oborů a oborů hodnot
- Modeluje se opět v RDF

RDF reprezentace užívaných metadatových schémat - Z39.50, Dublin Core atd.

- RDF je obecný rámec pro modelování metadat, pro konkrétní použití je obvykle nutné definovat *schéma* přípustných *vlastností*, jejich *domén* a množin (přípustných) *hodnot*.

- Tím se vytvoří RDF reprezentace daného metadatového schématu.
- Reprezentace může mít podobu *RDF Schematu*.

Dublin Core - příklad konkrétního metadatového schématu

Co je Dublin Core?

- je generické metadatové schéma s univerzální použitelností
- vznikl původně jako iniciativa knihovníků pro popis bibliografických informací
- dnes univerzálně používán - např. pro metadatový popis informací ve veřejné správě (*e-Government*)
- tvoří jej 15 základních elementů s rámcově definovanou sémantikou
- elementy je možné rozšiřovat - rozkladem na (obvykle disjunktní) podmnožiny (vždy to musí být podmnožiny některého z původních elementů)

Jednoduchý (Simple) Dublin Core

"Jednoduchý" nebo "základní" Dublin Core (angl. Simple Dublin Core nebo Unqualified Dublin Core, dále jen "jednoduchý DC") představuje základní soubor patnácti prvků, který vyvinula a podporuje

- *Iniciativa pro metadata Dublin Core* (Dublin Core Metadata Initiative, DCMI, http://dublincore.org).
- Momentálně je aktuální verzí Dublin Core 1.1.
- je přijat konsorciem IETF [http://ietf.org] jako tzv. *dokument RFC (Request For Comment) 2431*rovněž od 2003 jako *ISO Standard 15836-2003*

Dublin Core - elementy

```
Název Jméno dané zdroji Tvůrce Entita primárně odpovědná za
      vytvoření obsahu zdroje Předmět a klíčová slova Téma obsahu zdroje Popis
      Vysvětlení obsahu zdroje Vydavatel Entita odpovědná za zpřístupnění
      zdroje Přispěvatel Entita, která přispěla k vytvoření obsahu zdroje
      Datum Datum spojené s určitou událostí během existence zdroje Typ zdroje
      Povaha nebo druh obsahu zdroje Formát Fyzická nebo digitální
      reprezentace zdroje Identifikátor zdroje Jednoznačný odkaz na zdroj v
      rámci daného kontextu Zdroj Odkaz na zdroj, z něhož je popisovaný zdroj
      odvozen Jazyk Jazyk intelektuálního obsahu zdroje Vztah Odkaz na
      příbuzný zdroj Pokrytí Rozsah nebo záběr obsahu zdroje Správa autorských
      práv Informace o právech vztahujících se k popisovanému zdroji
```

DC - příklad metadatového popisu

```
Název Zelená kniha o elektronickém obchodu Tvůrce Úřad pro
      veřejné informační systémy, Úřad vlády Předmět Elektronický obchod,
      elektronický podpis, bezpečnost, správa Popis Vládní návrh podpory
      elektronického obchodu v České republice Datum vytvoření 2001-09-20
      Datum zveřejnění 2001-10-17 Identifikátor ISBN:?????
```

Kvalifikovaný Dublin Core

- (Qualified Dublin Core) obsahuje stejný soubor prvků jako jednoduchý DC a doporučuje další upřesnění a omezení každého prvku.
- Typicky se tak děje na základě formálního nebo de-facto mezinárodního standardu, např. může požadovat, aby prvek "jazyk" byl vyplněn v souladu se seznamem ISO pro jazyky (ISO 639).

Kódování DC v XML

DTD - http://dublincore.org/documents/2001/11/28/dcmes-xml/dcmes-xml-dtd.dtd [http://dublincore.org/documents/2001/11/28/dcmes-xml/dcmes-xml-dtd.dtd]

XML Schema - http://dublincore.org/documents/2001/11/28/dcmes-xml/dcmes-xml-xsd.xsd [http://dublincore.org/documents/2001/11/28/dcmes-xml/dcmes-xml-xsd.xsd]

RDF Schema - rdf/dc-rdf-schema-cz.rdf [/~tomp/xml/rdf/dc-rdf-schema-cz.rdf]

RDF Schema pro slovník typů (Type Vocabulary) - /~tomp/xml/rdf/dc-tv-rdf-schema-cz.rdf [/~tomp/xml/rdf/dc-tv-rdf-schema-cz.rdf]

Nástroje pro práci s RDF

Jena Java RDF API and toolkit [http://www.hpl.hp.com/semweb/]

The ICS-FORTH RDFSuite [http://139.91.183.30:9090/RDF/]

DC Creator na University of Bath [http://www.ukoln.ac.uk/cgi-bin/dcdot.pl]

další viz http://www.w3.org/RDF [http://www.w3.org/RDF/]

Příklady praktického použití metadat - veřejná správa

Rámec pro metadata ISVS ČR

Kroky budování

- Přijmout doporučení **Dublin Core** a osvojit jej jako **Národní metadatový standard** (NMS).
- Rozšířit tento standard tak, aby vyhovoval potřebám veřejné správy jak pro snadné vyhledávání informací, tak pro správu informačních zdrojů.
- Vyvinout **Aplikační profil NMS**, který bude obsahovat předepsaná kódovací schémata a závazný výklad jednotlivých metadatových prvků.
- Připravit **Tezaurus veřejné správy**.

Adaptace Dublin Core pro potřeby veřejné správy

pro potřeby veřejné správy v zemích Evropské Unie, Austrálie, Kanady a Nového Zélandu je rozpracováván specifický *aplikační profil* Dublin Core.

Cílem MIReG je vytvořit metadatový rámec (metadata framework), příslušné referenční softwarové nástroje a soubor osvědčených postupů (best practice) pro implementaci rámce v jednotlivých zemích a sektorech. Přitom spolupracuje také s evropskou standardizační autoritou CEN, což dává předpoklad celoevropského respektování vzniklého doporučení.

- proces zahájen na sérii pracovních seminářů **Managing information resources for e-government** (MIReG) a stal se součástí programu *Interchange of Data between Administrations (IDA)* Evropské Unie.
- Dalším partnerem při vytváření evropského metadatového rámce je též projekt **ParlML**, zaměřený na zpřístupňování informací Evropského parlamentu.
- Příslušná pracovní skupina připravuje doporučení **DC-Gov Application Profile**

Aplikační profil NMS

zahrnuje:

- **Upřesnění** (zjemnění, kvalifikaci, specializaci angl. element refinement) metadatových prvků, které přesněji určuje sémantiku daného prvku a tím jej rozděluje na jemněji (přesněji) určené podprvky - např. obecné datum lze kvalifikací rozdělit na menší části, a místo "datum" uvádět přesněji např. *"datum vytvoření", "datum zveřejnění", "datum platnosti", "nástupnické datum"*.
- Kvalifikovaný prvek lze však i nadále zpracovávat nástroji, které příslušné kvalifikaci "nerozumějí" - tyto nástroje potom chápou prvek jako by zůstal nekvalifikovaný (všeobecnější), tj. "datum zveřejnění" mohou chápat jako prosté "datum", čímž je sice část sémantiky ztracena, ale prvek může být stále užitečný např. pro vyhledávání.
- **Kódovací schémata** (též kvalifikace hodnoty, angl. encoding scheme nebo value qualification) specifikující formát, ve kterém bude uložena hodnota pro příslušný metadatový prvek, např. "datum" vždy bude uváděno ve formátu *rrrr-mm-dd* (rok-měsíc-den), což definuje standard ISO 8601.
- Kromě formátu může být kvalifikací hodnoty též např. specifikace *měrné jednotky*, v níž bude hodnota uváděna.

Ontologie

Co jsou ontologie?

prostředek jak popisovat znalosti

množina pojmů a konstruktů, jak je odvozovat, spojovat atd.

základní kategorie ontologií jsou

- **Classes** (general things) in the many domains of interest
- The **relationships** that can exist among things
- The **properties** (or **attributes**) those things may have

používá metadatové rámce (např. RDF), ale je

bohatší s přesnější sémantikou

jsou vybudovány obecné rámce pro tvorbu ontologií pro specifické domény

Aplikace ontologií (Use Cases)

- Webové portály, integrace dat na webu
- Multimediální kolekce
- Správa velkých webů
- Dokumentace návrhu
- Inteligentní agenti
- "Všudypřítomné počítání"

Pracovní skupina při W3C [http://www.w3.org/2001/sw/WebOnt/]

XML Topic Maps

Další návrh pracovní skupině WebOnt - http://www.topicmaps.org/xtm/1.0 [http://www.topicmaps.org/xtm/1.0/]