Technické standardy vytváření datových schémat pro datové sady na stupni otevřenosti 3

Na stupni otevřenosti 3 by měly mít distribuce datových sad přiřazeno datové schéma, které popisuje požadovanou syntaktickou strukturu distribucí. Pro vyjádření datového schématu je nutno zvolit vhodný jazyk pro jeho vyjádření. Jazyk závisí na datovém formátu, který byl zvolen pro vyjádření distribuce.

- V případě formátu CSV je nutno schéma vyjádřit v jazyku Metadata Vocabulary for Tabular Data (https://www.w3.org/TR/tabular-metadata/) či JSON Table Schema (https://specs.frictionlessdata.io/table-schema/).
- V případě formátu XML je nutno schéma vyjádřit v jazyku XML Schema (http://www.w3.org/XML/Schema).
- V případě formátu JSON je nutno schéma vyjádřit v jazyku JSON Schema (http://json-schema.org/).

Při návrhu datových schémat dodržujte následující pravidla:

- Pro primitivní datové typy (řetězec, celé číslo, datum, …) používejte datové typy jazyka XML Schema (i v případě CSV a JSON souborů).
- Schéma každé distribuce musí být volně dostupné ke stažení v síti WWW.
- Distribuce se stejnou strukturou mají společné schéma.
- Pokud je schéma částečně pokryto, tj. některé položky odpovídají významem, některým z již existujících standardů ve vzorových publikačních plánech, převezměte tyto položky. Nově definujte pouze ty položky, jejichž význam v žádném z existujících standardů pokryt není.

Definice vlastního schématu pro data v CSV

Při práci s formátem CSV se nejprve seznamte s nejčastějšími chybami při použití formátu CSV.

Prvním krokem k tvorbě schematu pro CSV data je určení toho, jak se budou jmenovat jednotlivé sloupce CSV souboru, jaké budou mít datové typy a jaký budou mít význam. Pro dosažení maximální míry interoperability postupujte v následujících krocích:

1. Podívejte se na již existující standardy pro datové sady.
    1. Pokud se některá datová sada shoduje s daty, které chcete publikovat, použijte její předpřipravené schéma. Existující schéma nemusíte daty pokrývat celé, všechny položky jsou volitelné.
    2. Pokud se žádná existující datová sada neshoduje s daty, které chcete publikovat, vytvořte nové schéma.
2. Pokud některá datová sada pokrývá data která chcete publikovat pouze částečně, použijte v novém schématu pro pokryté položky názvy sloupců, datové typy a popisky z existující datové sady.
3. Dosud nepokryté sloupce, tj. ty, jejichž význam neodpovídá žádnému sloupci v žádné existující datové sadě ze vzorových publikačních plánů, pojmenujte dle stejných jmenných konvencí. Tj.

    1. název v češtině
    2. všechna písmena malá (lower case)
    3. žádná diakritika
    4. víceslovné názvy spojené podtržítkem _
    5. hierarchickou vazbu reprezentujte také podtržítkem _, např. `pokutovany_ic`, `pokutovany_nazev`
4. Pro nově definované sloupce použijte vhodný datový typ jazyka XML Schema.

Jako jazyk pro definici schématu pro data v CSV můžete použít buďto standard Metadata Vocabulary for Tabular Data z rodiny standardů W3C CSV on the Web (CSVW), nebo definici Table Schema. Oba tyto jazyky říkají, jak má být CSV soubor publikovaný na webu popsán pomocí přídavného JSON souboru, který je publikován spolu s CSV souborem, a liší se pouze syntaxí a expresivitou tohoto JSON souboru. Doporučujeme však použít standard W3C.

Metadata Vocabulary for Tabular Data (CSV on the Web, CSVW)

Použití CSVW schématu si ilustrujeme na zjednodušeném příkladu pro následující dvousloupcové CSV:

```
"idhod","hodnota"
"747627675","14.91"
"747628556","14.96"
```

Jednoduchý CSVW deskriptor (JSON soubor) pro tento CSV soubor může vypadat například takto:

```
{
  "@context": ["http://www.w3.org/ns/csvw",{"@language": "cs"}],
  "url": "012052-17data091517.csv",
  "tableSchema": {
    "columns": [{
      "name": "idhod",
      "titles": "idhod",
      "dc:description":"unikátní identifikátor údaje Veřejné databáze ČSÚ",
      "required": true,
      "datatype": "string"
    }, {
      "name": "hodnota",
      "titles": "hodnota",
      "dc:description":"zjištěná hodnota",
      "required": true,
      "datatype": "number"
    }],
    "primaryKey": "idhod"
  }
}
```

Jednotlivé položky v JSON deskriptoru mají následující význam:

- Položka `@context` musí obsahovat minimálně URL `http://www.w3.org/ns/csvm`, v tomto případě obsahuje ještě specifikaci češtiny jakožto výchozího jazyka textových položek

schématu.
- Položka `url` musí obsahovat (relativní či absolutní) URL popisovaného CSV souboru. Každý CSV soubor má tedy vlastní JSON deskriptor.
- Položka `tableSchema` musí obsahovat buďto URL jiného JSONu se samotným schématem, což je použitelné pro sdílení jednoho schématu více CSV soubory a jejich JSON deskriptory, nebo přímo schéma samotné.
    - Položka `columns` obsahuje pole s popisky jednotlivých sloupců
    - Položka `primaryKey` obsahuje identifikaci primárního klíče v CSV tabulce. To může být buďto jeden sloupec, nebo pole sloupců.
- Položka `name` specifikuje *identifikátor* sloupce v CSV souboru jakožto objektu. Nejedná se o název sloupce v souboru, ten je popsán dále jako jedna z jeho vlastností.
- Položka `titles` obsahuje jeden či více (jako pole) názvů sloupců v CSV. Lze tedy použít jedno schéma pro více CSV souborů, které mají dokonce různé názvy sloupců, případně v hlavičce používají různé jazyky.
- Položka `dc:description` obsahuje textový popis významu sloupce.
- Položka `required` specifikuje, zda je hodnota v tomto sloupci povinná či nikoliv
- Pro datové typy v položce `datatype` lze použít hodnoty založené na datových typech XML Schema.

Vystavený JSON deskriptor dle CSVW by pak měl být poskytován s HTTP hlavičkou `Content-Type: application/csvm+json; charset=utf-8`.

Toto je jen minimalistický příklad toho, co lze popsat pomocí CSVW. Pro využití všech možností je třeba postupovat dle specifikace.

Sdílené schéma pro více CSV souborů

Deskriptor CSV souboru lze oddělit od schématu, pokud schéma chceme použít pro více CSV souborů. Pak budeme mít JSON soubor `schema.json` se schématem samotným:

```
{
  "@context": ["http://www.w3.org/ns/csvw",{"@language": "cs"}],
  "columns": [{
    "name": "idhod",
    "titles": "idhod",
    "dc:description":"unikátní identifikátor údaje Veřejné databáze ČSÚ",
    "required": true,
    "datatype": "string"
  }, {
    "name": "hodnota",
    "titles": "hodnota",
    "dc:description":"zjištěná hodnota",
    "required": true,
    "datatype": "number"
  }],
  "primaryKey": "idhod"
}
```

A deskriptor každého CSV souboru se na něj bude odkazovat (nejspíše však pomocí plného, absolutního URL):

```
{
  "@context": ["http://www.w3.org/ns/csvw",{"@language": "cs"}],
  "url": "012052-17data091517.csv",
  "tableSchema": "schema.json"
}
```

Validace CSVW

CSV popsané pomocí CSVW lze validovat například pomocí knihovny csvlint.rb, kterou stačí spustit a nasměrovat na JSON deskriptor, který pak ukazuje na CSV data, případně schéma. Tedy `csvlint -s schema.json` pro lokální schéma v souboru v souborovém systému, nebo s plným URL deskriptoru, např. `csvlint -s https://mvcr1.opendata.cz/czechpoint/2007.json`.

Table Schema

Použití Table Schema si opět ilustrujeme na zjednodušeném příkladu pro následující dvousloupcové CSV:

```
"idhod","hodnota"
"747627675","14.91"
"747628556","14.96"
```

Jednoduchý Table Schema deskriptor (JSON soubor) pro tento CSV soubor může vypadat například takto:

```
{
    "fields": [{
      "name": "idhod",
      "description":"unikátní identifikátor údaje Veřejné databáze ČSÚ",
      "required": true,
      "type": "default"
    }, {
      "name": "hodnota",
      "description":"zjištěná hodnota",
      "required": true,
      "type": "number"
    }
    ],
    "primaryKey": "idhod"
}
```

Jednotlivé položky mají stejný význam jako u schématu dle CSVW. Rozídly jsou následující:

- Pole se sloupci se místo `columns` jmenuje `fields`
- Místo `datatype` je `type`
- Místo `dc:description` je `description`
- Chybí povinný `@context`

Validace

Pro validaci lze použít online nástroj csvlint.io. Také lze použít knihovnu csvlint.rb. Jelikož ze schématu nevede link na samotná data, je třeba obojí poskytnout jako parametr, např. `csvlint data.csv -s schema.json`

Definice vlastního schématu pro data v XML

Pro popis XML schématu se používá jazyk XML Schema.

Prvním krokem k tvorbě schematu pro XML data je určení toho, jak se budou jmenovat a jak budou zanořeny jednotlivé elementy v XML souboru, jaké budou mít datové typy a jaký budou mít význam. Pro dosažení maximální míry interoperability postupujte v následujících krocích:

1. Podívejte se na již existující standardy pro datové sady.
    1. Pokud se některá datová sada shoduje s daty, které chcete publikovat, použijte její předpřipravené schéma. Existující schéma nemusíte daty pokrývat celé, všechny položky jsou volitelné.
    2. Pokud se žádná existující datová sada neshoduje s daty, které chcete publikovat, vytvořte nové schéma.
2. Pokud některá datová sada pokrývá data která chcete publikovat pouze částečně, použijte v novém schématu pro pokryté položky XML elementy s datovými typy z existujícího XML schématu.
3. Dosud nepokryté elementy, tj. ty, jejichž význam neodpovídá žádnému datovému typu v žádné existující datové sadě ze vzorových publikačních plánů, pojmenujte dle stejných jmenných konvencí. Tj.
    1. vytvořte si vlastní XML namespace
    2. názvy elementů či atributů v češtině
    3. všechna písmena malá (lower case)
    4. žádná diakritika
    5. víceslovné názvy spojené podtržítkem _
    6. hierarchickou vazbu reprezentujte vnořeným XML elementem
4. Pro nově definované elementy použijte vhodný datový typ jazyka XML Schema.