

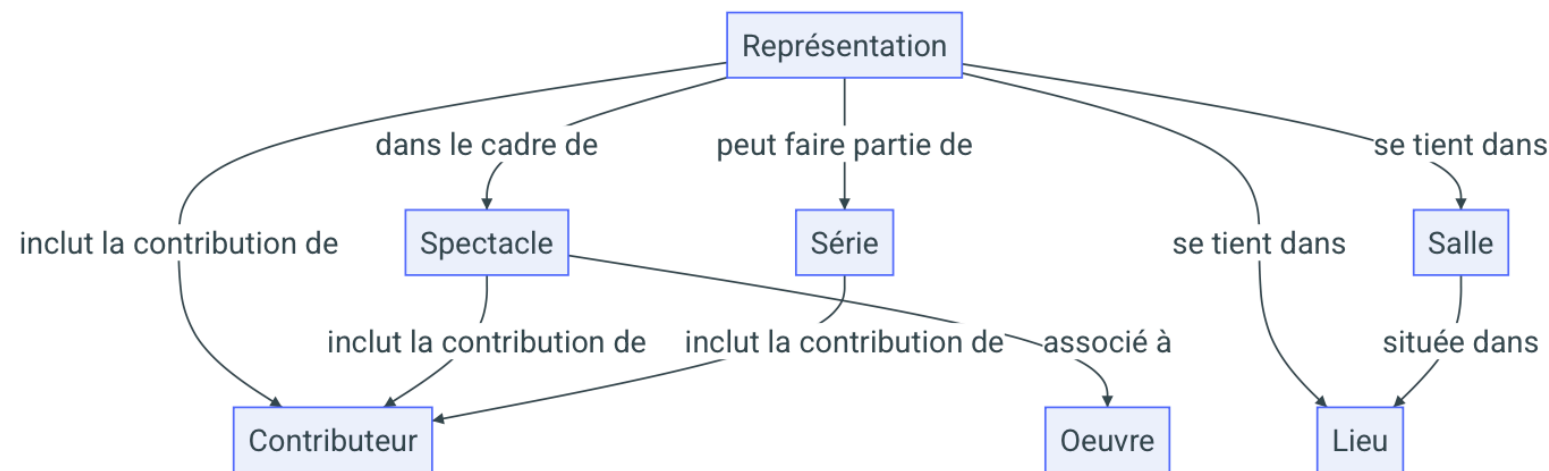
Datascene S rialisation

Rencontre 1

Contents

- Définitions
- Question: JSON-LD ou JSON
- Question: JSON Schema ou SHACL
- Am I going in the right direction?

Definitions



- La **serialization** d'un **modèle de données conceptuel** consiste à convertir sa structure abstraite en un format lisible par une machine, comme JSON, XML ou RDF, afin de le stocker, l'échanger ou l'exploiter dans des systèmes informatiques.
- Cela permet de préserver la logique et les relations du modèle tout en le rendant utilisable par des applications et des bases de données et de s'assurer l'interopérabilité entre différentes applications.

Question

JSON ou JSON-LD

JSON est un simple format de structuration des données.

JSON-LD est une extension de JSON qui permet d'intégrer des données liées (Linked Data) en ajoutant un contexte sémantique aux informations pour faciliter leur interconnexion et leur compréhension par les machines.

Interopérabilité

- JSON est principalement utilisé pour des échanges de données classiques
- JSON-LD est conçu pour être compris par des machines et intégré au Web sémantique

Utilisation

- JSON est couramment employé pour les API et le stockage de donnée
- JSON-LD est souvent utilisé pour le SEO (données structurées), le Web sémantique et l'open data

Exemple de Lieu

Grand Théâtre de Québec

JSON

```
{
  "type": "Lieu",
  "nom": "Grand Théâtre de Québec",
  "identifiant": [
    {
      "type": "uri",
      "value": "http://kg.artsdata.ca/resource/K5-180"
    },
    {
      "type": "uri",
      "value": "http://www.wikidata.org/entity/Q3114610"
    }
  ],
  "description": [
    {
      "langue": "fr",
      "texte": "Un important centre culturel de Québec, accueillant des spectacles variés."
    },
    {
      "langue": "en",
      "texte": "A major cultural center in Quebec City, hosting a variety of performances."
    }
  ],
  "lieuVirtuel": false,
  "salle": [
    {
      "nom": "Salle Louis-Fréchette",
      "description": "Une grande salle de spectacle avec une acoustique exceptionnelle."
    },
    {
      "nom": "Salle Octave-Crémazie",
      "description": "Une salle plus intime pour des performances variées."
    }
  ]
}
```

JSON-LD

```
{
  "@context": "http://datascience.ca/context.jsonld",
  "type": "Lieu",
  "nom": "Grand Théâtre de Québec",
  "identifiant": [
    {
      "type": "uri",
      "value": "http://kg.artsdata.ca/resource/K5-180"
    },
    {
      "type": "uri",
      "value": "http://www.wikidata.org/entity/Q3114610"
    }
  ],
  "description": [
    {
      "langue": "fr",
      "texte": "Un important centre culturel de Québec, accueillant des spectacles variés."
    },
    {
      "langue": "en",
      "texte": "A major cultural center in Quebec City, hosting a variety of performances."
    }
  ],
  "lieuVirtuel": false,
  "salle": [
    {
      "nom": "Salle Louis-Fréchette",
      "description": "Une grande salle de spectacle avec une acoustique exceptionnelle."
    },
    {
      "nom": "Salle Octave-Crémazie",
      "description": "Une salle plus intime pour des performances variées."
    }
  ]
}
```

Noms et URIs

Exemple **https://datascene.ca/modele#Lieu**

Protocol:

- **https**
- ~~http~~

Domaine:

- **datascene.ca**

Path:

- **modele**
- ~~modele/classes_principales,~~
~~modele/classes_utilitaires~~

Fragment:

- **#**
- ~~none~~

Nom de classe:

- **camel case** avec **majuscule** (Lieu)
- ~~minuscules (lieu)~~
- **singulière**
- ~~plurielles~~

Nom de propriétés:

- ~~snake case (lieu_virtuel)~~
- **Camel case** avec **minuscule** (lieuVirtuel)
- **singulière**
- ~~plurielles~~

Prefix

Raccourci

Exemple **datascene:Lieu**

Common usage according to <https://prefix.cc>

- ds: is already commonly mapped to <http://purl.org/ctic/dcat#>
- **datascene:** is not used
- dscene: is also not used
- d: is not recommended (2 character minimum)

Étapes

JSON Schema

1. Définition des JSON Schemas

Chaque classe identifiée (classes principales et classes utilitaires) doit être traduite en un JSON Schema distinct.

Un JSON Schema définit la structure attendue des données, les types de données, les contraintes et les relations entre les propriétés et les classes. Il contient aussi des descriptions courtes de chaque classe et propriété.

2. Validation

Utiliser des outils de validation pour s'assurer que les schémas respectent les standards JSON Schema et qu'ils décrivent correctement les structures de données attendues.

En suivant ces étapes, le modèle conceptuel de Datascene sera efficacement sérialisé en JSON Schema, offrant une documentation structurée et conforme aux standards, facilitant ainsi l'interopérabilité et la compréhension des données liées aux spectacles.

JSON Schema

Classe "Lieu"

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://datascene.ca/lieu.schema.json",
  "title": "Lieu",
  "description": "sert à décrire un lieu, typiquement associé à une représentation. La
  classe est générique et les lieux peuvent correspondre à des édifices ou à des lieux
  extérieurs. Il s'agit du lieu associé à une adresse, ou à des indications
  géographiques précises, qui serait présenté à un consommateur comme l'endroit où se
  présenter pour assister à une représentation.",
  "required": [
    "identifiant",
    "nom",
    "lieuVirtuel"
  ],
  "type": "object",
  "properties": {
    "identifiant": {
      "description": "classe permettant d'énumérer des identifiants uniques associés à
      un objet donné, dans plusieurs systèmes d'information à la fois.",
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "type": {
            "description": "Type d'identifiant",
            "type": "string",
            "minLength": 1,
            "maxLength": 100
          },
          "value": {
            "description": "Valeur de l'identifiant",
            "type": "string",
            "minLength": 1,
            "maxLength": 100
          }
        }
      },
      "required": [
        "type",
        "value"
      ]
    },
    "nom": {
      "type": "string",
      "minLength": 1
    }
  },
  "minItems": 1,
  "uniqueItems": true
}
```

```

  "description": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "langue": {
          "type": "string",
          "minLength": 2,
          "maxLength": 2
        },
        "texte": {
          "type": "string"
        }
      }
    },
    "required": [
      "Langue",
      "Texte"
    ]
  },
  "minItems": 1
},
"lieuVirtuel": {
  "type": "boolean"
},
"salle": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "Nom": {
        "type": "string",
        "minLength": 1
      },
      "Description": {
        "type": "string"
      },
      "Capacite": {
        "type": "integer",
        "minimum": 0
      }
    }
  },
  "required": [
    "Nom",
    "Capacite"
  ]
},
"minItems": 0
}
}
}
```

Validateur

Newtonsoft

PASS

Select schema: Custom

```

1 {
2   "$schema": "https://json-schema.org/draft/2020-12/schema",
3   "$id": "https://datascene.ca/lieu.schema.json",
4   "title": "Lieu",
5   "description": "sert à décrire un lieu, typiquement associé à une
6     représentation. La classe est générique et les lieux peuvent correspondre
7     à des édifices ou à des lieux extérieurs. Il s'agit du lieu associé à une
8     adresse, ou à des indications géographiques précises, qui serait présenté
9     à un consommateur comme l'endroit où se présenter pour assister à une
10    représentation.",
11   "required": [
12     "identifiant",
13     "nom",
14     "lieuVirtuel"
15   ],
16   "type": "object",
17   "properties": {
18     "identifiant": {
19       "description": "classe permettant d'énumérer des identifiants uniques
20         associés à un objet donné, dans plusieurs systèmes d'information à la
21         fois.",
22       "type": "array",
23       "items": {
24         "type": "object",
25         "properties": {
26           "description": "Type d'identifiant",
27           "type": "string",

```

Input JSON:

```

1 {
2   "nom": "Grand Théâtre de Québec",
3   "identifiant": [
4     {
5       "type": "uri",
6       "value": "http://kg.artsdata.ca/resource/K5-180"
7     },
8     {
9       "type": "uri",
10      "value": "http://www.wikidata.org/entity/Q3114610"
11    }
12  ],
13  "description": [
14    {
15      "Langue": "fr",
16      "Texte": "Un important centre culturel de Québec, accueillant des
17        spectacles variés."
18    },
19    {
20      "Langue": "en",
21      "Texte": "A major cultural center in Quebec City, hosting a variety of
22        performances."
23    }
24  ],
25  "lieuVirtuel": false,
26  "salle": [
27    {
28      "Nom": "Salle Louis-Frédette",

```

✓ No errors found. JSON validates against the schema

```

11 },
12 "type": "object",
13 "properties": {
14   "@type": {
15     "type": "string"
16   },
17   "identifiant": {
18     "description": "classe permettant d'énumérer des identifiants uniques
19       associés à un objet donné, dans plusieurs systèmes d'information à la
20       fois.",
21     "type": "array",
22     "items": {
23       "type": "object",

```

```

16 "langue": "fr",
17 "texte": "Un important centre culturel de Québec, accueillant des
18   spectacles variés."
19 },
20 {
21   "langue": "en",
22   "texte": "A major cultural center in Quebec City, hosting a variety of
23     performances."
24 }
25 ],
26 "lieuVirtuel": false,
27 "salle": [
28   {

```

✗ Found 1 error(s)

Message: **Required properties are missing from object: nom.**
 Schema path: **https://datascene.ca/lieu.schema.json#/required**

FAIL

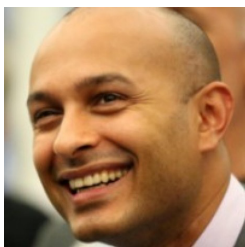
JSON-LD

Syntaxe rigide ou sémantique?

```
{
  "@context": "http://datascience.ca/context.jsonld",
  "type": "Lieu",
  "nom": "Grand Théâtre de Québec",
  "description": [
    {
      "langue": "fr",
      "texte": "Un important centre culturel de Québec, accueillant des spectacles variés."
    },
    {
      "langue": "en",
      "texte": "A major cultural center in Quebec City, hosting a variety of performances."
    }
  ]
}
```

=

```
{
  "@context": "http://datascience.ca/context2.jsonld",
  "type": "Lieu",
  "nom": "Grand Théâtre de Québec",
  "description": {
    "fr": "Un important centre culturel de Québec, accueillant des spectacles variés.",
    "en": "A major cultural center in Quebec City, hosting a variety of performances."
  }
}
```



Manu Sporny :

JSON-LD and JSON Schema were designed to complement each other. Each of the technologies do fundamentally different things, even though it might seem like a few features overlap.

JSON Schema is used to enforce a certain structure and syntax in a JSON document. JSON Schema answers the question: Is the input document syntactically valid for my application?

JSON-LD is used to annotate and interpret a JSON document in a globally unambiguous way. JSON-LD answers the question: How should my application interpret the information in the input document?

<https://github.com/w3c/vc-json-schema/issues/122>

Autre Approche?

SHACL

- Dans un modèle conceptuel, les entités ont souvent des relations complexes (ex. : un spectacle est lié à plusieurs représentations, des contributeurs, des lieux...)
- JSON Schema fonctionne bien pour des structures arborescentes, mais il devient limité lorsqu'il s'agit de gérer des relations croisées ou des références multiples
- SHACL, grâce à RDF et les triplets sujet-prédicat-objet, permet de représenter et de valider naturellement ces liens complexes.
- SHACL est préférable à JSON Schema si votre modèle conceptuel est lié au Web sémantique, nécessite des relations complexes et doit être facilement évolutif.

Validateur 2

Zazuko SHACL Playground

PASS

Shapes Graph

```
1 @prefix sh: <http://www.w3.org/ns/shacl#> .
2 @prefix schema: <http://schema.org/> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4 @prefix datascene: <http://datascence.ca/> .
5 @prefix : <http://datascence.ca/> .
6
7 :Lieu
8   sh:targetClass :Lieu ;
9   sh:description "sert à décrire un lieu, typiquement associé à un événement" ;
10  sh:property nomSh, :identifiantSh, :lieuVirtuel .
11
12 :nomSh
13   sh:path :nom ;
14   sh:message "Nom est obligatoire.";
15   sh:datatype xsd:string ;
16   sh:minCount 1 ;
```

Data Graph

```
1 {
2   "@context": { "@vocab": "http://datascence.ca/" },
3   "@type": "Lieu",
4   "nom": "Grand Théâtre de Québec",
5   "identifiant": [
6     {
7       "type": "uri",
8       "value": "http://kg.artsdata.ca/resource/K5-180"
9     },
10    {
11      "type": "uri",
12      "value": "http://www.wikidata.org/entity/Q3114610"
13    }
14  ],
15  "description": [
16    {
```

Validation Report

Success

Yes

```
12 :nomSh
13   sh:path :nom ;
14   sh:message "Nom est obligatoire.";
15   sh:datatype xsd:string ;
16   sh:minCount 1 ;
```

```
12   }
13   ],
14   "description": [
15     {
16       "langue": "fr".
```

Validation Report

Success

No

Errors found

- b1016:
 - http://datascence.ca/nom:
 - Nom est obligatoire.

FAIL

Question

JSON Schema ou SHACL

	JSON Schema	SHACL
Latest Version	<u>2022-06-16</u> (Draft 2020-12)	<u>2017-07-20</u> (W3C recommendation)
First Version	<u>2016-10-13</u>	<u>2015-10-08</u>
Data	JSON ou JSON-LD	JSON-LD et tous les format RDF
Format	JSON	Turtle
MkDocs plugin	<u>MkDocs JSON Schema</u>	none found but other documentation generators exist
PROS	accessible to a large community of JSON developers and widely supported in web and API development	highly expressive for semantic constraints, relationships, and ontologies
CONS	limitations in semantic reasoning	Specialized skills related to RDF technologies SPARQL and Turtle

Voir aussi mon document: JSON Schema/SHACL: Technical Comparison