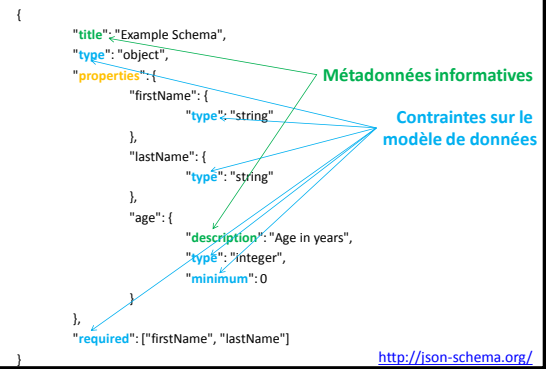
JSON SCHEMA

JSON Schema in a nutschell

- Langage permettant de définir le modèle d'une BD JSON
- JSON Schema est à JSON
  ce que DTD ou XML Schema sont pour XML
- http://json-schema.org/

JSON Schema in a nutschell



Exemple d'un objet JSON…

```
{
  "id": 1,
  "name": "A green door",
  "price": 12.50,
  "tags": ["home", "green"]
}
```

… et de son schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Product",
  "description": "A product from Acme's catalog",
  "type": "object",
  "properties": {
    "id": { … },
    "name": { … },
    "price": { … },
    "tags": { … },
      "minItems": 1,
      "uniqueItems": true
    }
  },
  "required": ["id", "name", "price"]
}
```

… et de son schema

```
…
"properties": {
  "id": {
    "description": "The unique identifier for a product",
    "type": "integer"
  },
  "name": {
    "description": "Name of the product",
    "type": "string"
  },
…
```

… et de son schema

```
...
"properties": {
  ...
  "price": {
    "type": "number",
    "minimum": 0,
    "exclusiveMinimum": true
  },
  "tags": {
    "type": "array",
    "items": {
      "type": "string"
    },
    ...
```

Type de valeur ou liste de valeurs possibles

- **type**
  - string
  - number
  - integer
  - array
  - boolean
  - null
  - object
- **enum**
  - tableau des valeurs possibles

http://json-schema.org/latest/json-schema-validation.html

Validation des valeurs numériques et chaines de caractères

- Validation des nombres
  maximum, minimum
  exclusiveMaximum, exclusiveMinimum (par défaut false)
  multipleOf
- Validation des chaines de caractères
  maxLength, minLength
  pattern (expression régulière)

http://json-schema.org/latest/json-schema-validation.html

Validation des tableaux

- maxItems, minItems (integer)
- uniqueItems (boolean)
- additionalItems, items :
  si la valeur associée à "additionalItems" est false et
  si la valeur associée à "items" est un tableau, alors la donnée tableau est valide si sa taille ne dépasse pas celle de "items".

Extrait de schéma JSON:

```
{
  "items": [ {}, {}, {} ],
  "additionalItems": false
}
```

Exemples de tableaux valides:
[] , [1, 2, 3] ,
[[1, 2, 3, 4], [5, 6, 7, 8]]
Exemples de tableaux non valides:
[1, 2, 3, 4] ,
[null, { "a": "b" }, true, 31.000002020013]

http://json-schema.org/latest/json-schema-validation.html

Validation des objets

- maxProperties, minProperties (integer)
- required (tableau)
  un objet est valide si son ensemble de propriétés contient tous les éléments du tableau
- properties, additionalProperties et patternProperties
- dependencies

http://json-schema.org/latest/json-schema-validation.html

Référence à d'autres JSON Schemas

- allOf
- anyOf
- oneOf
- Not
- definitions

http://json-schema.org/latest/json-schema-validation.html

Ressources web sur JSON Schema

- Tutoriel JSON Schema
  http://spacetelescope.github.io/understanding-json-schema/
- Validateurs JSON Schema en ligne
  e.g. http://www.jsonschemavalidator.net/
- APIs de validation
  e.g. https://pypi.python.org/pypi/jsonschema
  e.g. https://github.com/fge/json-schema-validator