

NoSQL avec MongoDB

Cours 5 - Schéma JSON de Validation

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Documents Impurs
- 2 Validation des documents
- 3 Création du schéma JSON
- 4 Insertion de documents invalides

Documents impurs

Une règle de pouce de MongoDB est la **simplicité**.

Il est important par conception (design) que la base de données soit simple, rapide et efficace, peu importe en termes d'utilisation ou de performances.



Figure: La simplicité et rapidité de traitement des données.

Documents Impurs

Ceci dit, des désavantages peuvent survenir en lien avec l'intégrité du corpus des données.

Il faut se rappeler que, par défaut, tous les documents peuvent être sauvegardés dans la base de données **peu importe leur composition** vis-à-vis les données actuellement présentes.

Documents Impurs

Cette grande liberté peut causer des problèmes majeurs avec une base de données qui doit maintenir une structure pure à tout coût.

La base de données relationnelle (MySQL, Postgres, etc.) le fait par conception (design), mais elles sont plus fastidieuses à utiliser pour les opérations courantes et simples, ainsi que le développement logiciel (plus de temps à initialiser les aspects divers).

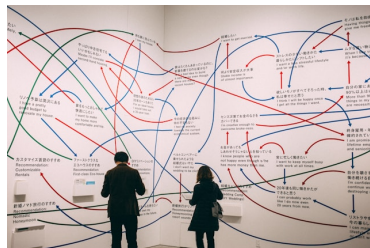
Validation des documents

Pour avoir une certaine assurance contre les documents irréguliers et inconforts au contenu de la base de données, il est possible d'intégrer des règles de validations (valideur) à la collection touchée.

Avec un niveau de validation et d'action complètement personnalisable, les manipulations quelconques sont gérées comme il le faut.

Création du schéma JSON

Le schéma JSON est créé avec des contraintes sur les **propriétés des documents (attributs et leurs contraintes spécifiques)**.



Création du schéma JSON

Listing: <https://labex.io/tutorials/mongodb-how-to-use-json-schema-validation-436478>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  vegetable_schema = {"$jsonSchema": {
5      "title": "Record of a Vegetable",
6      "description":
7          "This document records the details of a Vegetable",
8      "bsonType": 'object',
9      "required": ["name"],
10     "properties": {
11         "name": {
12             "type": "string",
13             "minLength": 2,
14         }
15     }
16 }
17 client = MongoClient()
18 db = client.VegetableJSONSchema
19 db.create_collection("VegetablesValidated", validator=vegetable_schema,
20                     validationLevel='strict', validationAction='error')
21 collection = db.VegetablesValidated
```


Insertion de documents invalides



Lorsqu'un document invalide est inséré (modifié), un message d'erreur étant à même titre qu'une erreur d'exécution est déclenché.

Insertion de documents invalides

Listing: Le document 2 sur 3 est invalide puisqu'il ne respecte pas la règle de validation du schéma JSON

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.VegetableJSONSchema
6  collection = db.VegetablesValidated
7
8  collection.insert_one({"name": "Valid"}) # Valid document.
9  collection.insert_one({"name": "T"}) # Error here.
10 collection.insert_one({"name": "ABC"}) # Not reached.
11
12 """
13 pymongo.errors.WriteError: Document failed validation, full error:
14 ...
15 [{ 'operatorName': 'minLength', 'specifiedAs': {'minLength': 2}, 'reason':
16 'specified string length was not satisfied', 'consideredValue': 'T' } ] ] ] ] } } }
17
18 Process finished with exit code 1
19 """
```

Insertion de documents invalides

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS (1)' sidebar lists the database 'Aliments' and its collections: 'VegetableJSONSchema', 'VegetablesValidated', 'admin', 'Test', 'Vegetables', 'config', and 'local'. The main panel displays the 'Documents' tab for the 'VegetablesValidated' collection, showing 97 documents. The first document is highlighted with a red box, showing its _id and name. The second document is shown below it, but its insertion is blocked by an error.

Documents 97 Aggregations Schema Indexes 1 Validation

Search connections

Aliments

- VegetableJSONSchema
- VegetablesValidated**
- admin
- Test
- Vegetables
- config
- local

ADD DATA EXPORT DATA UPDATE DELETE

`_id: ObjectId('67f0b346a5a6727fb20bd2d4')`
`name: "Valid"`

`_id: ObjectId('67f0b42cbf3c7b5abfccfb2')`
`name: "Beet greens, raw"`
`energy (kcal/kJ): "22/92"`
`water (g): 91.2`
`protein (g): 2.2`
`total fat (g): 0.13`
`carbohydrates (g): 3.7`
`fiber (g): 4.22`

Figure: Seulement le 1er document est dans la base de données, puisque le 2e document est invalide, et l'erreur produite empêche l'insertion du 3e document.

Bibliographie

- <https://labex.io/tutorials/mongodb-how-to-use-json-schema-validation-436478>