



Bài 4

Schema Validation in MongoDB

- › Giới thiệu về schema validation
- › Validating String Fields
- › Validating Number Fields
- › Validating Array Fields
- › Validating Embedded Documents

- Một khía cạnh quan trọng của cơ sở dữ liệu quan hệ (RDBMS) - lưu trữ cơ sở dữ liệu trong các bảng được tạo thành từ các hàng và cột - đó là chúng hoạt động trên các lược đồ cố định, cứng nhắc với các trường kiểu dữ liệu đã biết.
- Cơ sở dữ liệu hướng tài liệu như MongoDB linh hoạt hơn về mặt này, vì chúng cho phép bạn định hình lại cấu trúc tài liệu của mình nếu cần.
- Khi cần các tài liệu dữ liệu tuân theo một cấu trúc cụ thể hoặc đáp ứng các yêu cầu nhất định. MongoDB có một tính năng gọi là xác thực lược đồ (**schema validation**) cho phép áp dụng các ràng buộc trên cấu trúc tài liệu của mình.

Để gán tài liệu trình xác thực lược đồ JSON (**JSON_Schema_document**) cho bộ sưu tập các định bạn đã tạo ở bước trước, bạn có thể chạy lệnh sau:

Lưu ý:

- **Collection_name:** tên collection muốn áp dụng xác thực dữ liệu
- **JSON_Schema_document:** là một cấu trúc json khai báo quy tắc xác thực dữ liệu

```
// Khi đã có collection
db.runCommand({
  "collMod": "collection_name",
  "validator": {
    $jsonSchema: {JSON_Schema_document}
  }
})

// Khi tạo mới collection
db.createCollection(
  "collection_name", {
    "validator": {
      $jsonSchema: {JSON_Schema_document}
    }
  })
```

- Ví dụ sau đây xác thực trường **name** của collection **users** không rỗng, phải là một chuỗi

```
db.runCommand({
  "collMod": "users",
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "description": "Dữ liệu tài khoản người dùng",
      "required": ["name"],
      "properties": {
        "name": {
          "bsonType": "string",
          "description": "Tên không rỗng, phải là chuỗi"
        }
      }
    }
  }
})
```

- Sau khi áp dụng lệnh trên, màn hình command line có dạng

```
> db.runCommand({
...   "collMod": "users",
...   "validator": {
...     $jsonSchema: {
...       "bsonType": "object",
...       "description": "Dữ liệu tài khoản người dùng",
...       "required": ["name"],
...       "properties": {
...         "name": {
...           "bsonType": "string",
...           "description": "Tên không rỗng, phải là chuỗi"
...         }
...       }
...     }
...   }
... })
{ "ok" : 1 }
```

- Khi cố tình thêm mới bản ghi không đúng với xác thực
- Không truyền dữ liệu cho name

```
// Khi không nhập dữ liệu cho trường name
db.users.insert({email:"myemail123@gmail.com",password:"123456"})
// Sẽ có báo lỗi
```

```
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 121,
    "errmsg" : "Document failed validation",
    "errInfo" : {
      "failingDocumentId" : ObjectId("62d0d2e87a496592c1491e03"),
      "details" : {
        .....
      }
    }
  }
})
```

- Khi cố tình thêm mới bản ghi không đúng với xác thực

Truyền dữ liệu cho name không phải chuỗi

```
db.users.insert({name:123,  
email:"demo1@gmail.com",password:"123456"})  
// Sẽ có báo lỗi
```

```
WriteResult({  
  "nInserted" : 0,  
  "writeError" : {  
    "code" : 121,  
    "errmsg" : "Document failed validation",  
    "errInfo" : {  
      "failingDocumentId" : ObjectId("62d0d61b7a496592c1491e04"),  
      "details" : {  
        // .....  
      }  
    }  
  }  
})
```


- Ví dụ sau đây xác thực trường **age** của collection **users** không rỗng, phải là số từ 18 đến 30

```
db.runCommand({
  "collMod": "users",
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "description": "Dữ liệu tài khoản người dùng",
      "required": ["name","age"],
      "properties": {
        "name": { //..... },
        "age": {
          "bsonType": "number",
          "description": "TUổi không rỗng, phải là số từ 18, 30",
          "minimum": 18,
          "maximum": 30
        }
      }
    }
  }
})
```

- Ví dụ sau đây xác thực trường **roles** của collection **users**, phải là mảng ít nhất một phần tử, không trùng lặp và giá trị mỗi phần tử là chuỗi

```
db.runCommand({
  "collMod": "users",
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "description": "Dữ liệu tài khoản người dùng",
      "required": ["name", "age", "roles"],
      "properties": {
        // name, age validate here
        "roles": {
          "bsonType": "array",
          "description": "Roles phải là mảng, ít nhất 1 phần tử, không trùng",
          "minItems": 1,
          "uniqueItems": true,
          "items": {
            "bsonType": "string"
          }
        }
      }
    }
  }
})
```

- Đối với kiểu xác thực này, MongoDB cho phép xác thực một object khi nó được nhúng vào tài liệu
- Chúng ta áp dụng lại quy tắc như các kiểu trên cho object được nhúng này

```
db.runCommand({
  "collMod": "users",
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "description": "Dữ liệu tài khoản người dùng",
      "required": ["name", "age", "roles"],
      "properties": {
        // name, age, roles validate here
        "marks": {
          "bsonType": "object",
          "description": "Điểm số",
          "required": ["toan", "ly"],
          "properties": {
            "toan": {
              "bsonType": "number",
              "description": "Điểm toán từ 0 - 10",
              "minimum": 0,
              "maximum": 10
            }
          }
        }
      }
    }
  }
})
```

Validating Embedded Documents

THANK FOR WATCH !



TRẢI NGHIỆM THỰC HÀNH

HỆ THỐNG ĐÀO TẠO CNTT QUỐC TẾ BACHKHOA - APTECH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn