

05

Criando novo documento

Transcrição

Continuando com as operações básicas do banco de dados, podemos inserir um novo documento, atualizá-lo e também removê-lo. Já vimos como se faz a leitura, e que tudo que precisamos é de um cliente Mongo, a partir do qual selecionamos o banco e manipulamos as coleções. Nelas, realizamos todas as operações: `insertOne` , `insertMany` , `remove` , `update` e várias outras.

Nosso próximo passo é inserir o aluno João em nossa coleção de alunos. Precisaremos criar um novo documento para isso! No Mongo, os documentos são pares de chave e valor. Cada chave geralmente é uma `string` , e o valor pode ser qualquer coisa, inclusive outros documentos. A classe `Document` pode receber em seu construtor o primeiro par de chave e valor, e os demais pares são adicionados ao documento utilizando-se o método `append` do objeto desta classe. Assim:

```
Document novoAluno = new Document("nome", "Joao")
.append("data_nascimento", new Date(2003, 10, 10))
.append("curso", new Document("nome", "Historia"))
.append("notas", Arrays.asList(10, 9, 8))
.append("habilidades", Arrays.asList(
    new Document().append("nome", "Ingles").append("nível", "Básico"),
    new Document().append("nome", "Espanhol").append("nível", "Básico")));
```

Note que, no construtor, passamos o nome do aluno e os demais campos são preenchidos com o método `append` , sempre utilizando-se chave e valor. Então criaremos o novo documento com a mesma estrutura dos alunos do [curso de MongoDB](https://cursos.alura.com.br/course/mongodb) (<https://cursos.alura.com.br/course/mongodb>).

A estrutura do aluno tem os seguintes campos: nome (sendo uma string), data de nascimento (de tipo `Date`), curso, que é outro documento com nome do curso, um campo "nota", que é um `array` ou lista de notas, e o campo "habilidades", documentos em um `array` ou lista em que estão descritos o nome e o nível da habilidade.

Após a criação deste novo aluno, precisaremos utilizar o método `insertOne` do objeto `alunos` para inseri-lo na coleção de alunos.

```
alunos.insertOne(novoAluno);
```

Podemos executar a classe `Principal` novamente, e consultar o resultado da inserção no console no Mongo.

```
{
  "_id" : ObjectId("5991c69c161a4b2483549774"),
  "nome" : "Joao",
  "data_nascimento" : ISODate("3903-11-10T03:00:00Z"),
  "curso" : {
    "nome" : "Historia"
  },
  "notas" : [
    10,
```

```
9,  
8  
],  
"habilidades" : [  
  {  
    "nome" : "Ingles",  
    "nível" : "Básico"  
  },  
  {  
    "nome" : "Espanhol",  
    "nível" : "Básico"  
  }  
]  
}
```

O código da classe `Principal` se encontra assim:

```
public class Principal {  
  public static void main(String[] args){  
    MongoClient cliente = new MongoClient();  
    MongoDatabase bancoDeDados = cliente.getDatabase("test");  
    MongoCollection<Document> alunos = bancoDeDados.getCollection("alunos");  
    Document aluno = alunos.find().first();  
    Document novoAluno = new Document("nome", "Joao")  
      .append("data_nascimento", new Date(2003, 10, 10))  
      .append("curso", new Document("nome", "Historia"))  
      .append("notas", Arrays.asList(10, 9, 8))  
      .append("habilidades", Arrays.asList(  
        new Document().append("nome", "Ingles").append("nível", "Básico"),  
        new Document().append("nome", "Espanhol").append("nível", "Básico")));  
  
    alunos.insertOne(novoAluno);  
    System.out.println(aluno);  
    cliente.close();  
  }  
}
```

O uso da classe `Date` está depreciado, porém, por motivos de demonstração nos exemplos do curso ela funciona e é bem útil.