Достъп до MongoDB с Java – инсталации и търсене на съдържание

Необходимо програмно осигуряване:

1. MongoDB Java драйвер

Трябва да имате предвид, че има множество версии на MongoDB Java драйвера. Използвайте по възможност последната стабилна версия на драйвера. Използвайте хранилището MVNrepository.com, за да свалите драйвера:

https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver

Формата, която се визуализира при избор на версия 3.12.8 на драйвера, е следната:

MongoDB Java Driver » 3.12.8

The MongoDB Java Driver uber-artifact, containing the legacy driver, the mongodb-driver, mongodb-driver-core, and bson

License	Apache 2.0
Categories	MongoDB Clients
HomePage	http://www.mongodb.org
Date	(Feb 18, 2021)
Files	jar (2.2 MB) View All
Repositories	Central

Щракнете върху препратката “jar”, за да свалите драйвера. Запишете драйвера в папка, например “MongoDBLibraries”.

2. JSON coder/decoder

За да може да обработвате данните, които MongoDB сървъра връща като отговор на вашите заявки, ще ви трябва библиотека за работа с JSON обекти. Тъй като такава библиотека не е интегрирана в Java 8, ще използваме библиотека, която може да свалите от следния адрес:

https://mvnrepository.com/artifact/org.json/json

Изберете последната налична версия, например 20210307. Щракнете върху препратката “bundle”, за да свалите библиотеката. Запишете файла в папка “MongoDBLibraries”.

JSON In Java » 20210307

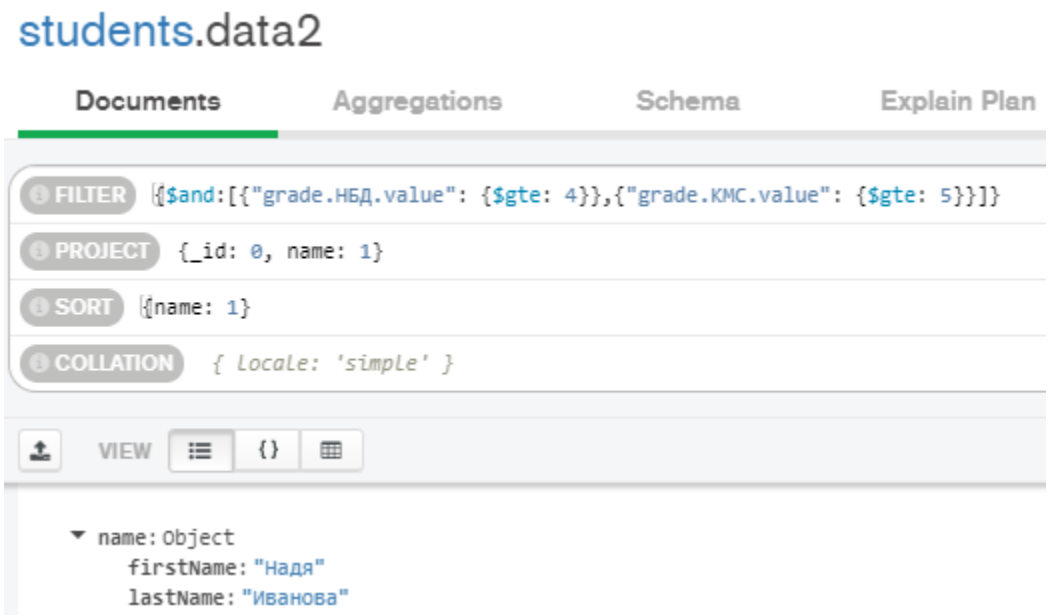
License	JSON
Categories	JSON Libraries
HomePage	https://github.com/douglascrockford/JSON-java
Date	(Mar 09, 2021)
Files	bundle (68 KB) View All

Задача 1: *Напишете Java конзолно приложение, което връща сортиран списък на всички студенти, които имат оценки над зададена стойност по зададени две дисциплини. Работете с база данни „students”, колекция „data2”.*

Колекция „data1” съдържа документи, които описват студентите и техните оценки като името на дисциплините се използват като ключ към оценките:

```
{
    "id": "21705001",
    "name": {
        "firstName": "Георги",
        "lastName": "Котев"
    },
    "grade": {
        "НБД": {
            "value": 3
        },
        "КМС": {
            "value": 6
        },
        "ММС": {
            "value": 2
        },
        "ДСП": {
            "value": 5
        }
    }
}
```

Стартирайте MongoDB Compass, за да симулираме и тестваме необходимата заявка:



Filter: При конкретната заявка се проверява кои студенти имат по дисциплината НБД оценка по-голяма или равна на 4 и оценка по дисциплината КМС по-голяма или равна на 5.

Project: Задаваме отговорът на сървъра да съдържа само името на студента.

Soft: Сортиране на имената на студентите от А -> Я.

Натиснете бутон [...] и експортирайте заявката до код на Java:

```
Bson filter = new Document("$and", Arrays.asList(
        new Document("grade.НБД.value",
        new Document("$gte", 4L)),
        new Document("grade.КМС.value",
        new Document("$gte", 5L))));
Bson project = new Document("_id", 0L)
    .append("name", 1L);
Bson sort = new Document("name", 1L);

MongoClient mongoClient = new MongoClient(
    new MongoClientURI(

"mongodb://localhost:27017/?readPreference=primary&appname=MongoDB%20Compass&ssl=f
alse"
    )
);
MongoDatabase database = mongoClient.getDatabase("students");
MongoCollection<Document> collection = database.getCollection("data2");
FindIterable<Document> result = collection
    .find(filter)
    .projection(project)
    .sort(sort);
```

Създайте NetBeans проект и реализирайте Задача 1.

1. Задайте следните константи:

```
final static int PORT = 27017;
final static int TIMEOUT = 3000;
final static String DATABASE = "students";
final static String COLLECTION = "data2";
final static String HOST_NAME = "localhost";

final static String SUBJECT_1 = "НБД";
final static String SUBJECT_2 = "КМС";
final static long SUBJECT_1_VAL = 4;
final static long SUBJECT_2_VAL = 5;
```

2. Забранете Java MongoDB драйвера да извежда на конзолата информация, свързана с неговата работа:

```
Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
mongoLogger.setLevel(Level.SEVERE);
```

3. Проверете дали MongoDB сървъра е активен:

```
InetAddress ip;
try {
    ip = InetAddress.getByName(HOST_NAME);
    Socket socket = new Socket();
    socket.connect(new InetSocketAddress(ip, PORT), TIMEOUT);
} catch (UnknownHostException uhe) {
    System.out.println("Unknown host name!");
    return;
} catch (IOException iox) {
    System.out.println("MongoDB server is not working!");
    return;
}
```

4. Дефинирайте необходимите обекти за filter, project и sort:

```
Bson filter = and(Arrays.asList(
        gte("grade." + SUBJECT_1 + ".value", SUBJECT_1_VAL),
        gte("grade." + SUBJECT_2 + ".value", SUBJECT_2_VAL)));
Bson project = and(eq("name", 1L), eq("_id", 0L));
Bson sort = eq("name", 1L);
```

5. Създайте необходимите обекти за достъп до желаната колекция с документи:

```
MongoClient mongoClient = new MongoClient(
        new MongoClientURI(
                "mongodb://" + HOST_NAME + ":" + PORT
        )
);
MongoDatabase database = mongoClient.getDatabase(DATABASE);
MongoCollection<Document> collection = database.getCollection(COLLECTION);
```

6. Генерирайте желаната find заявка:

```
FindIterable<Document> result = collection
        .find(filter)
        .projection(project)
        .sort(sort);
```

7. Разпечатайте желаната информация, след като я извлечете от обект result:

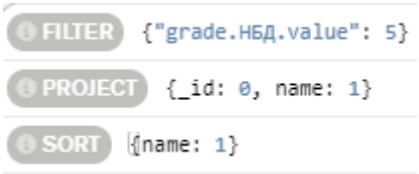
```
for (Document document : result) {
    String json = document.toJson();
    JSONObject obj = new JSONObject(json);
    JSONObject nameObj = obj.getJSONObject("name");
    String firstName = nameObj.getString("firstName");
    String lastName = nameObj.getString("lastName");
    System.out.printf("%s %s\n", firstName, lastName);
}
```

8. Тествайте приложението:

```
run:
Надя Иванова
BUILD SUCCESSFUL (total time: 1 second)
```

Задача 2: *Напишете Java конзолно приложение, което връща сортиран списък на всички студенти, които имат зададена оценка по зададена дисциплина. Името на дисциплината и търсената оценка да се въвеждат от клавиатурата. Работете с база данни „students”, колекция „data2”.*

Тествайте синтаксиса на желаната заявка чрез MongoDB Compass:



Натиснете бутон [**...**] и експортирайте заявката до код на Java.

Създайте NetBeans проект и реализирайте Задача 2.

1. Задайте следните константи:

```
final static int PORT = 27017;
final static int TIMEOUT = 3000;
final static String DATABASE = "students";
final static String COLLECTION = "data2";
final static String HOST_NAME = "localhost";

final static int GRADE_MIN_VAL = 2;
final static int GRADE_MAX_VAL = 6;
```

2. Забранете Java MongoDB драйвера да извежда на конзолата информация, свързана с неговата работа.

3. Проверете дали MongoDB сървъра е активен.

4. Въведете от клавиатурата името на дисциплината и оценката:

При въвеждане на данни от клавиатурата се използва кодова таблица ISO/IEC 8859-1 (Latin-1). За да можете да въвеждате текст на кирилица е необходимо да конвертирате данните до кодова таблица Cp1251:

```
Scanner input = null;
try {
    input = new Scanner(new InputStreamReader(System.in, "Cp1251"));
} catch (UnsupportedEncodingException e) {
}
System.out.print("Please enter the name of the subject: ");
long grade = 0;
String subject = input.nextLine();
do {
    System.out.print("Please enter the grade: ");
    try {
        grade = input.nextLong();
    } catch (InputMismatchException e) {
        System.out.println("The grade must be a number between 2 and 6.");
        input.next();
    }
} while (grade < GRADE_MIN_VAL | grade > GRADE_MAX_VAL);
input.close();
```

5. Дефинирайте необходимите обекти за filter, project и sort:

```
Bson filter = eq("grade."+ subject +".value", grade);
Bson project = and(eq("name", 1L), eq("_id", 0L));
Bson sort = eq("name", 1L);
```

6. Създайте необходимите обекти за достъп до желаната колекция с документи:

```
MongoClient mongoClient = new MongoClient(
        new MongoClientURI(
                "mongodb://" + HOST_NAME + ":" + PORT
        )
);
MongoDatabase database = mongoClient.getDatabase(DATABASE);
MongoCollection<Document> collection = database.getCollection(COLLECTION);
```

7. Генерирайте желаната find заявка:

```
FindIterable<Document> result = collection
        .find(filter)
        .projection(project)
        .sort(sort);
```

8. Разпечатайте желаната информация, след като я извлечете от обект result:

```
for (Document document : result) {
    String json = document.toJson();
    JSONObject obj = new JSONObject(json);
    JSONObject nameObj = obj.getJSONObject("name");
    String firstName = nameObj.getString("firstName");
    String lastName = nameObj.getString("lastName");
    System.out.printf("%s %s\n", firstName, lastName);
}
```

9. Тествайте приложението:

```
run:
Please enter the name of the subject: НБД
Please enter the grade: пет
The grade must be a number between 2 and 6.
Please enter the grade: 5
Весела Иванова
```