

Exercice : requêtes sur la base des films.

Préparation

```
mongoimport -d test -c movies --file movies.json --jsonArray
```

Copiez *movies.json* dans le dossier **data/files/** et mettez-y votre curseur (CMDOS)

Sur votre base **movies**, affichez :

1. Tous les titres;
2. Tous les titres des films parus après 2000;
3. Le résumé de Spider-Man;
4. Qui est le metteur en scène de Gladiator ?
5. Titre des films avec Kirsten Dunst;
6. Quels films ont un résumé?
7. Les films qui ne sont ni des drames ni des comédies.
8. Affichez les titres des films et les noms des acteurs.
9. Dans quels films Clint Eastwood est-il acteur mais pas réalisateur (aide: utilisez l'opérateur de comparaison \$ne).

Solution

```
C:\Program Files\MongoDB\Server\4.0\data\files>mongoimport -d test -c movies --file movies.json --jsonArray
2019-05-13T18:45:42.812+0200    connected to: localhost
2019-05-13T18:45:42.889+0200    imported 65 documents
```

1.

```
db.movies.find({}, {"title": 1})
```

2.

```
db.movies.find({"year": {$gt: 2000}}, {"title": 1, "year": 1})
```

3.

```
db.movies.find({"title": "Spider-Man"}, {"summary": 1})
```

4.

```
db.movies.find({"title": "Gladiator"}, {"director": 1})
```

5.

```
db.movies.find({"actors.last_name": "Dunst"}, {"title": 1})
```

6.

```
db.movies.find({"summary": {$exists: true}}, {"title": 1})
```

NB: cette fonction regarde si le champ existe, pas s'il est vide ou non. Dans la base, il existe des films avec un résumé ayant pour valeur null.

Afin de ne récupérer que les films ayant réellement un résumé, on peut ajouter: `$ne:null`

```
db.movies.find({"summary": {$exists: true, $ne:null}}, {"title": 1})
```

donne les films dont le champ résumé existe et dont la valeur du champ est différente de null.

7.

```
db.movies.find({"genre": {$nin: ["drama", "Comédie"]}}, {"title": 1, "genre": 1})
```

8.

```
db.movies.find({}, {"title": 1, "actors.first_name": 1, "actors.last_name": 1})
```

9.

```
db.movies.find({"actors.last_name": "Eastwood", "director.last_name": {$ne: "Eastwood"}}, {"title": 1})
```

Pour éviter le problème d'avoir « Clint » dans un élément de l'array et « Eastwood » dans un autre <https://stackoverflow.com/questions/14040562/how-to-search-in-array-of-object-in-mongodb>