MongoDB

04

Presented by
Subhash EP

Subhash

mongoDB

{ name: mongo, type: DB }

Subhash

Querying Documents

- Read operations
- Write operations
- Write concerns
- Bulk writing documents

Query through find() interface

- find can accept
  - queries as criteria and
  - projections as parameters.
- This will result in a cursor

Query through find() interface

- Cursors have methods
  - that can be used as modifiers of the executed query,
  - such as limit, map, skip, and sort
- Example:
  - 0117.txt
  - 0118.txt

parameters

- use JSON documents in find parameters.
- We can use find in the following way:

```
db.collection.find(
  {criteria},
  {projection}
)
```

criteria and projection

- criteria is a JSON document that will specify the criteria for the selection of documents inside a collection by using some operators
- projection is a JSON document that will specify which document's fields in a collection will be returned as the query result
- Example: 0119.txt

find all

- in the find interface, both the criteria and projection parameters are optional.
- To use the find interface without any parameters means selecting all the documents in a collection.
- Example: 0120.txt

Selecting documents using criteria

- selecting all the documents in a collection can turn out to be a bad idea due to a given collection's length.
- it is essential to create a criterion in order to select only the documents we want.

- Example: 0121.txt

operators on the criteria

- to select all documents where the price is greater than 10:

- Example: 0122.txt

Comparison operators

- With comparison operators, we can compare BSON type values.
- The $gte operator is for equal or greater than the value
- Example:
  - db.products.find({price: {$gte: 20}})

Comparison Operators

- $gt
- $gte
- $lt
- $lte

$in comparison

- to search any document where the value of a field equals a value that is specified in the requested array in the query
- Example: 0123.txt

$nin

- Search for not included
- Example: 0124.txt

$ne operator

- will match any values that are not equal to the specified value in the query.
- Example:
  - db.products.find({name: {$ne: "Product 1"}})

That's all

End of Session

©

subhash.ep@gmail.com
linkedin.com/in/subhashep