

MongoDB Functions Overview



mongo db functions



MongoDB provides a wide variety of functions to interact with its NoSQL database. Here are some common functions used in MongoDB, categorized by their purpose:

1. CRUD Operations (Create, Read, Update, Delete)

Create:

- `insertOne()`: Inserts a single document into a collection.

```
js
```

```
db.collection.insertOne({ name: "John", age: 30 })
```

- `insertMany()`: Inserts multiple documents at once.

```
js
```

```
db.collection.insertMany([ { name: "Alice", age: 25 }, { name: "Bob", age: 28 } ])
```

Read:

- `find()`: Fetches documents based on a query. By default, returns all documents.

```
js
```

```
db.collection.find({ age: { $gt: 20 } })
```

- `findOne()`: Fetches a single document that matches the query.

```
js
```

```
db.collection.findOne({ name: "John" })
```

Update:

- `updateOne()`: Updates the first document that matches the query.

```
js
```

```
db.collection.updateOne({ name: "John" }, { $set: { age: 31 } })
```

- `updateMany()`: Updates multiple documents that match the query.

```
js
```

```
db.collection.updateMany({ age: { $lt: 30 } }, { $set: { status: "young" } })
```

- `replaceOne()`: Replaces an entire document.

```
js
```

```
db.collection.replaceOne({ name: "John" }, { name: "Jonathan", age: 31 })
```

Delete:

- `deleteOne()`: Deletes a single document that matches the query.

```
js
```

```
db.collection.deleteOne({ name: "John" })
```

- `deleteMany()`: Deletes all documents that match the query.

```
js
```

```
db.collection.deleteMany({ age: { $lt: 30 } })
```

2. Aggregation:

- `aggregate()`: Performs advanced data aggregation, like filtering, sorting, and transforming data.

```
js
```

```
db.collection.aggregate([
  { $match: { status: "active" } },
  { $group: { _id: "$age", total: { $sum: 1 } } }
])
```

3. Indexes:

- ``createIndex()``: Creates an index on a field to improve query performance.

```
js

db.collection.createIndex({ name: 1 })
```

- ``dropIndex()``: Removes an index.

```
js

db.collection.dropIndex({ name: 1 })
```

4. Other Common Functions:

- ``countDocuments()``: Returns the count of documents matching a query.

```
js

db.collection.countDocuments({ age: { $gt: 20 } })
```

- ``distinct()``: Returns distinct values for a given field.

```
js

db.collection.distinct("age")
```

- ``bulkWrite()``: Allows multiple write operations in a single call (e.g., insert, update, delete).

```
js

db.collection.bulkWrite([
  { insertOne: { document: { name: "Sam", age: 22 } } },
  { updateOne: { filter: { name: "Alice" }, update: { $set: { age: 26 } } } },
  { deleteOne: { filter: { name: "Bob" } } }
])
```

5. Transaction Functions (for MongoDB replica sets or sharded clusters):

- ``startSession()``: Starts a session for transactions.

```
js

const session = db.getMongo().startSession()
session.startTransaction()
```

- ``commitTransaction()``: Commits a transaction.

js

```
session.commitTransaction()
```

- `abortTransaction()`: Aborts a transaction.

js

```
session.abortTransaction()
```

These are some of the key functions you would commonly use when working with MongoDB. Let me know if you need more detailed examples or have a specific use case in mind!

ChatGPT can make mistakes. Check important info.