

Utilisation des opérateurs de mise à jour

Utilisation des opérateurs de mise à jour les plus courants en MongoDB, en supposant que nous travaillons avec une collection hypothétique appelée `documents` :

1. \$set

Utilisé pour définir une nouvelle valeur pour un champ spécifié ou pour ajouter un nouveau champ à un document.

Mettre à jour le champ **status** d'un document avec un **id** spécifique.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $set: { status: "complete" } }  
);
```

2. \$unset

Supprime un champ spécifié d'un document.

Supprimer le champ **age** d'un document.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $unset: { age: "" } }  
);
```

3. \$inc

Incrémente la valeur d'un champ par une valeur spécifiée. Si le champ n'existe pas, il sera défini à la valeur spécifiée.

Augmenter le **score** de 10 pour un document.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $inc: { score: 10 } }  
);
```

4. \$mul

Multiplie la valeur d'un champ par une valeur spécifiée.

Doubler la valeur du champ **quantity** pour un document.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $mul: { quantity: 2 } }  
);
```

5. \$push

Ajoute un élément à un tableau. Si le champ n'est pas un tableau, la commande échoue.

Ajouter un nouvel élément à un tableau **tags**.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $push: { tags: "mongodb" } }  
);
```

6. \$pull

Supprime tous les éléments d'un tableau qui correspondent à une condition spécifiée.

Supprimer tous les éléments avec la valeur "mongodb" du tableau **tags**.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $pull: { tags: "mongodb" } }  
);
```

7. \$addToSet

Ajoute une valeur à un tableau uniquement si elle n'est pas déjà présente (pour éviter les doublons).

Ajouter "unique-tag" au tableau **tags** seulement s'il n'est pas déjà présent.

```
db.documents.updateOne(  
  { _id: ObjectId("507f191e810c19729de860ea") },  
  { $addToSet: { tags: "unique-tag" } }  
);
```

Utilisation des options

Lorsque vous utilisez la commande update dans MongoDB, vous avez plusieurs options à votre disposition qui peuvent modifier le comportement de l'opération de mise à jour. Voici les options les plus courantes :

1. upsert

Type : Boolean

Description : Si true, crée un nouveau document lorsque aucun document ne correspond au critère de recherche. Par défaut, sa valeur est false.

Usage : Utile si vous voulez insérer un nouveau document si aucun document existant ne correspond à votre critère de mise à jour.

2. multi

Type : Boolean

Description : Si true, permet la mise à jour de plusieurs documents qui correspondent au critère de recherche. Par défaut, sa valeur est false pour updateOne() et true pour updateMany().

Usage : Utilisé pour déterminer si un ou plusieurs documents doivent être mis à jour.

Exemple d'Utilisation avec Options

Voici un exemple d'utilisation de update avec l'option upsert :

```
db.collection.update(  
  { key: "value" }, // Critère de recherche  
  { $set: { field: "newValue" } }, // Mise à jour  
  { upsert: true } // Option  
);
```