

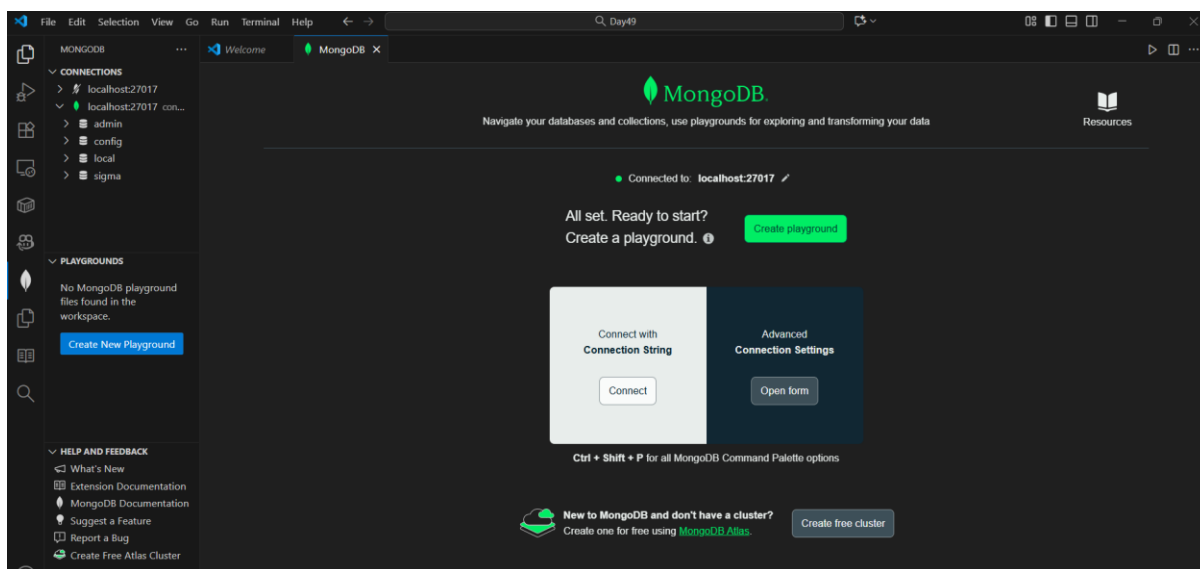
Day 49

“Web Development + Security”

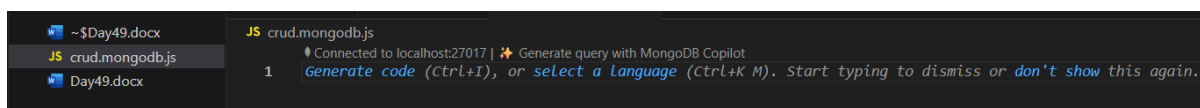
CRUD operations in MongoDB:

CRUD operations (Create, Read, Update, Delete).

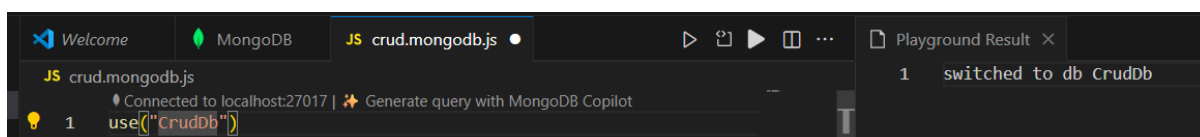
First connect the MongoDB to the VScode:



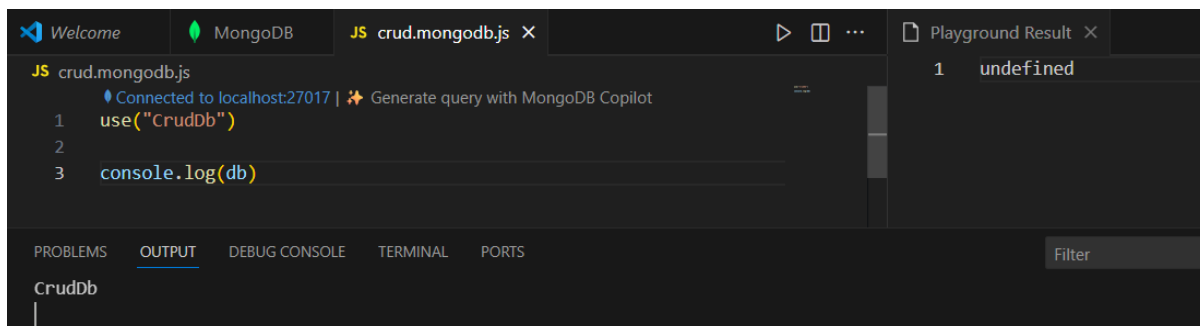
Now, we are creating our own playground: file name be crud.mongodb.js



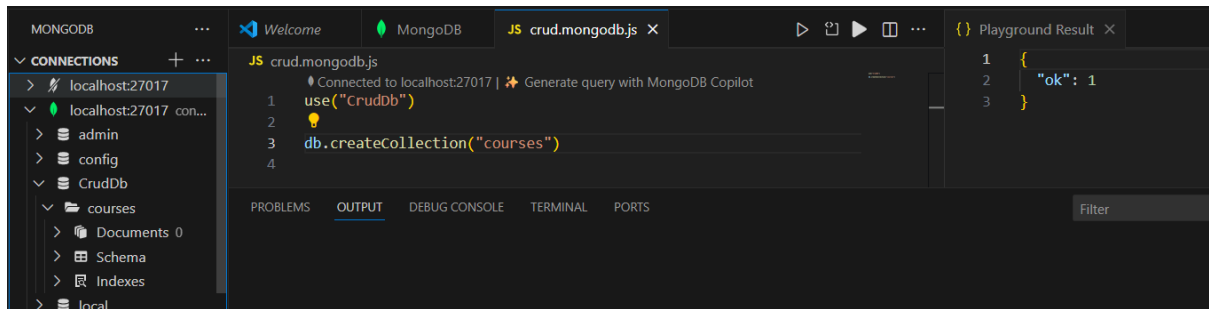
Now, suppose we want to use the Db named “CrudDb” (it may not exists), then we will use the following code: clearly, it switched to CrudDb Db.



Now, to see which Db we are using, we can use console.log(db), as shown below: clearly it says CrudDb is used.



Now, in order to create a new collection in the CrudDb, we will use `createCollection`, as shown below: clearly it worked as expected.

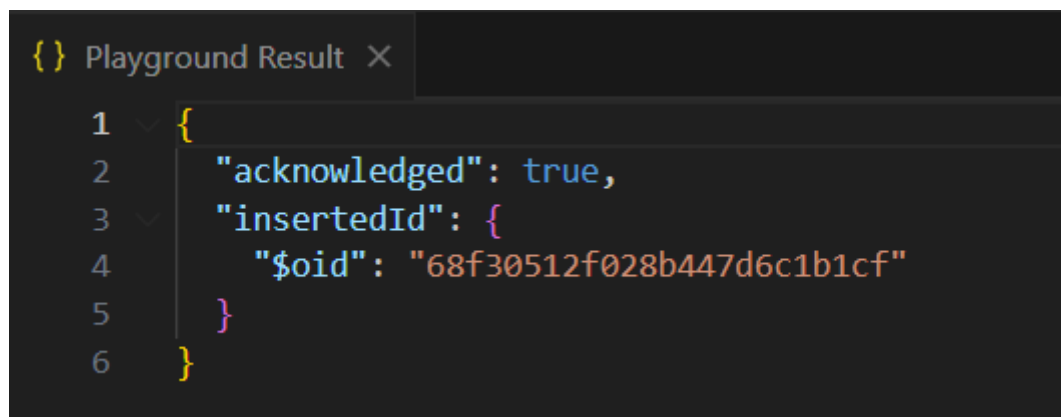


Now, in order to insert a document in the collection we will use the `.insertOne`, as shown below:

Code:



Playground result:



DB json:



Now, similarly, insert the next document: make sure to comment out earlier one otherwise it will get repeated.

```
JS crud.mongodb.js > ...
  Connected to localhost:27017 | Generate query with MongoDB Copilot
1  use("CrudDb")
2
3  db.createCollection("courses")
4
5  // db.courses.insertOne({
6  //   name: "Aditya course",
7  //   price: 0,
8  //   assignment: 12,
9  //   projects: 46
10 // })
11
12 db.courses.insertOne({
13   name: "Utsav course",
14   price: 0,
15   assignment: 14,
16   projects: 36
17 })
```

Now, suppose we want to find those courses where the price is 0, then we will use `.find()` as shown below:

```
19 let a = db.courses.find({price:0})
20 console.log(a)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

{
  cursorHasMore: false,
  documents: [
    {
      _id: ObjectId('68f30512f028b447d6c1b1cf'),
      name: 'Aditya course',
      price: 0,
      assignment: 12,
      projects: 46
    },
    {
      _id: ObjectId('68f30603412afd80a18311c1'),
      name: 'Utsav course',
      price: 0,
      assignment: 14,
      projects: 36
    }
  ]
}
```

Now, suppose we want to count the number of results returned by above “a”, since it is a cursor, we can count using .count(), as shown below: so basically in the Db we have 2 documents whose price is 0.

```
22 let a = db.courses.find({price:0})
23 console.log(a.count())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

2

Now, in order to get all the Documents in the form of array we will use: .toArray()

```
22 let a = db.courses.find({price:0})
23 console.log(a.toArray())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[
  {
    _id: ObjectId('68f30512f028b447d6c1b1cf'),
    name: 'Aditya course',
    price: 0,
    assignment: 12,
    projects: 46
  },
  {
    _id: ObjectId('68f30603412afd80a18311c1'),
    name: 'Utsav course',
    price: 0,
    assignment: 14,
    projects: 36
  }
]
```

Now, in case we want to get the first result whose price is 0, we will use .findOne() as shown below:

```
25 let a = db.courses.findOne({price:0})
26 console.log(a)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

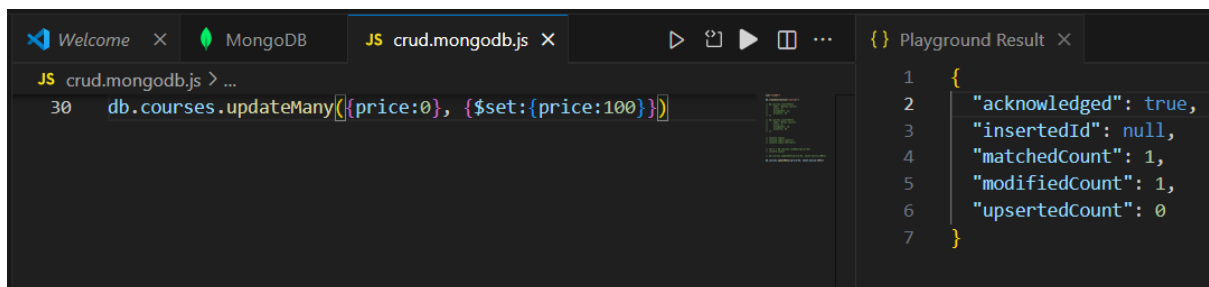
```
{
  _id: ObjectId('68f30512f028b447d6c1b1cf'),
  name: 'Aditya course',
  price: 0,
  assignment: 12,
  projects: 46
}
```

Now, in case we want to update the value we will use `.updateOne`, as shown below:



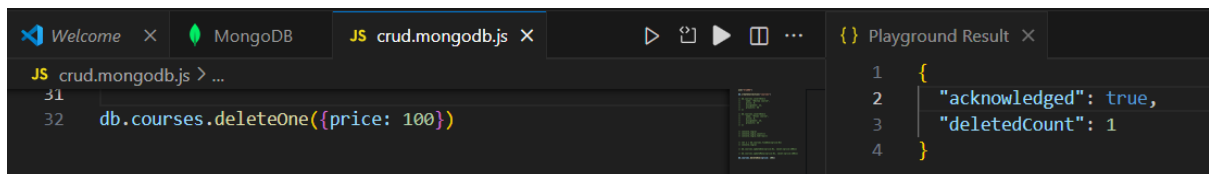
The screenshot shows the MongoDB Playground interface. The left pane contains the command `db.courses.updateOne({price:0}, {$set:{price:100}})` at line 28. The right pane, titled 'Playground Result', shows the result of the operation: `{ "acknowledged": true, "insertedId": null, "matchedCount": 1, "modifiedCount": 1, "upsertedCount": 0 }`.

Now, in case we want to update all documents matching the condition, we will use `.updateMany()` as shown below:



The screenshot shows the MongoDB Playground interface. The left pane contains the command `db.courses.updateMany({price:0}, {$set:{price:100}})` at line 30. The right pane, titled 'Playground Result', shows the result of the operation: `{ "acknowledged": true, "insertedId": null, "matchedCount": 1, "modifiedCount": 1, "upsertedCount": 0 }`.

Now, in case we want to delete the documents matching a certain condition we will use `.deleteOne()`, as shown below:



The screenshot shows the MongoDB Playground interface. The left pane contains the command `db.courses.deleteOne({price: 100})` at line 32. The right pane, titled 'Playground Result', shows the result of the operation: `{ "acknowledged": true, "deletedCount": 1 }`.

--The End--