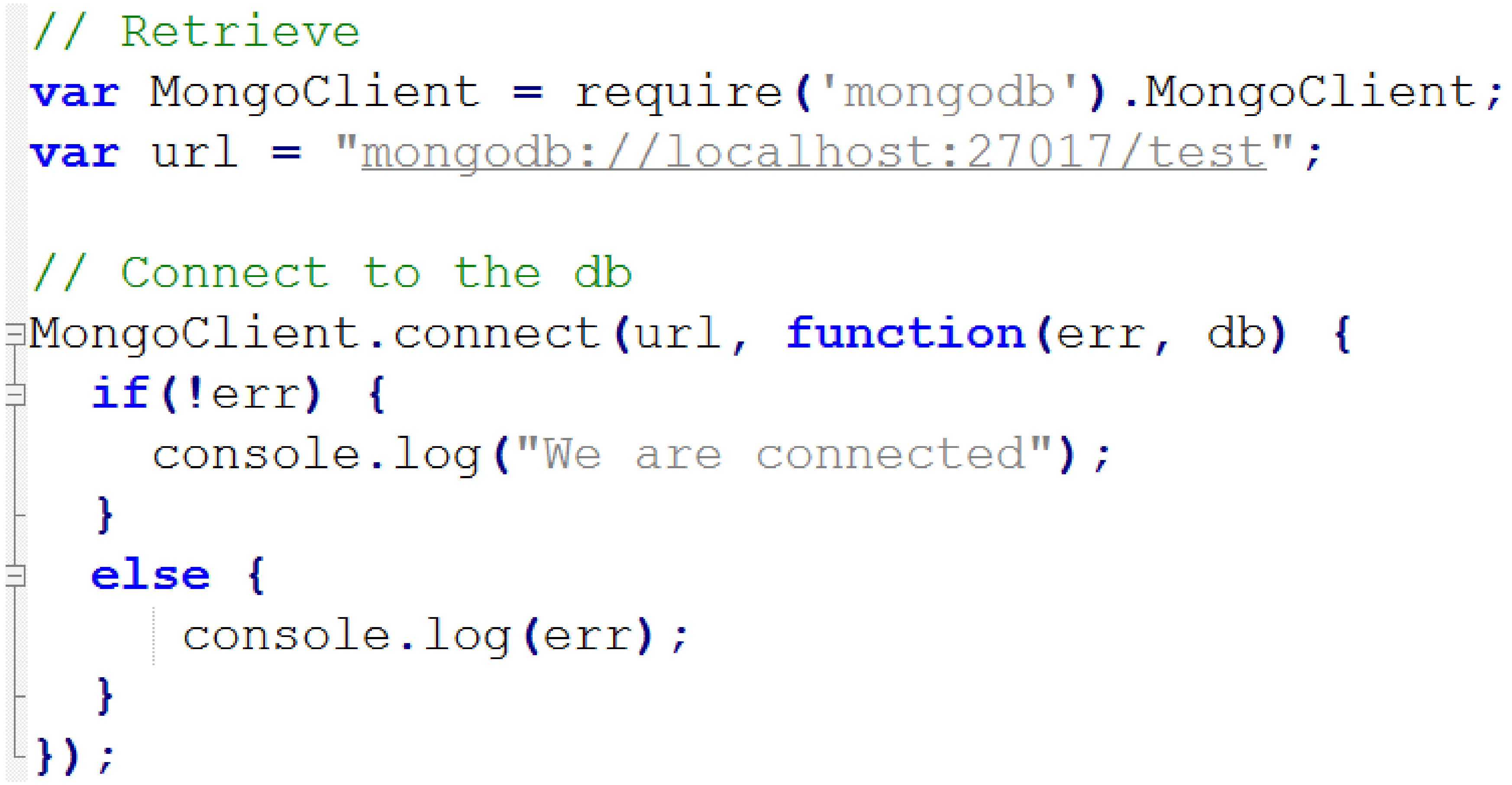
MongoDB queries with NodeJS

NodeJS MongoDB Driver

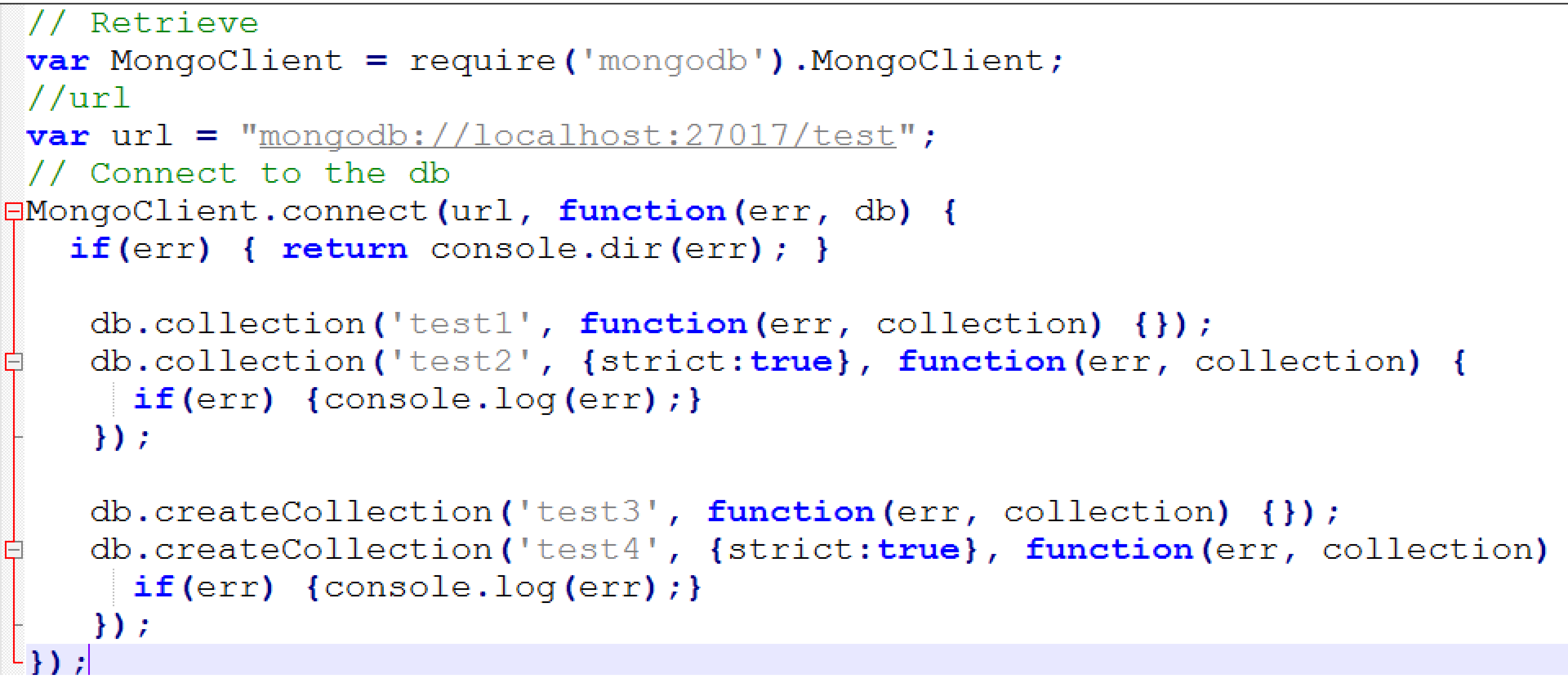
- To connect MongoDB in NodeJS, we can use node-mongodb-native driver.
- To Install “mongodb” module
  - npm install mongodb

Connection to the database



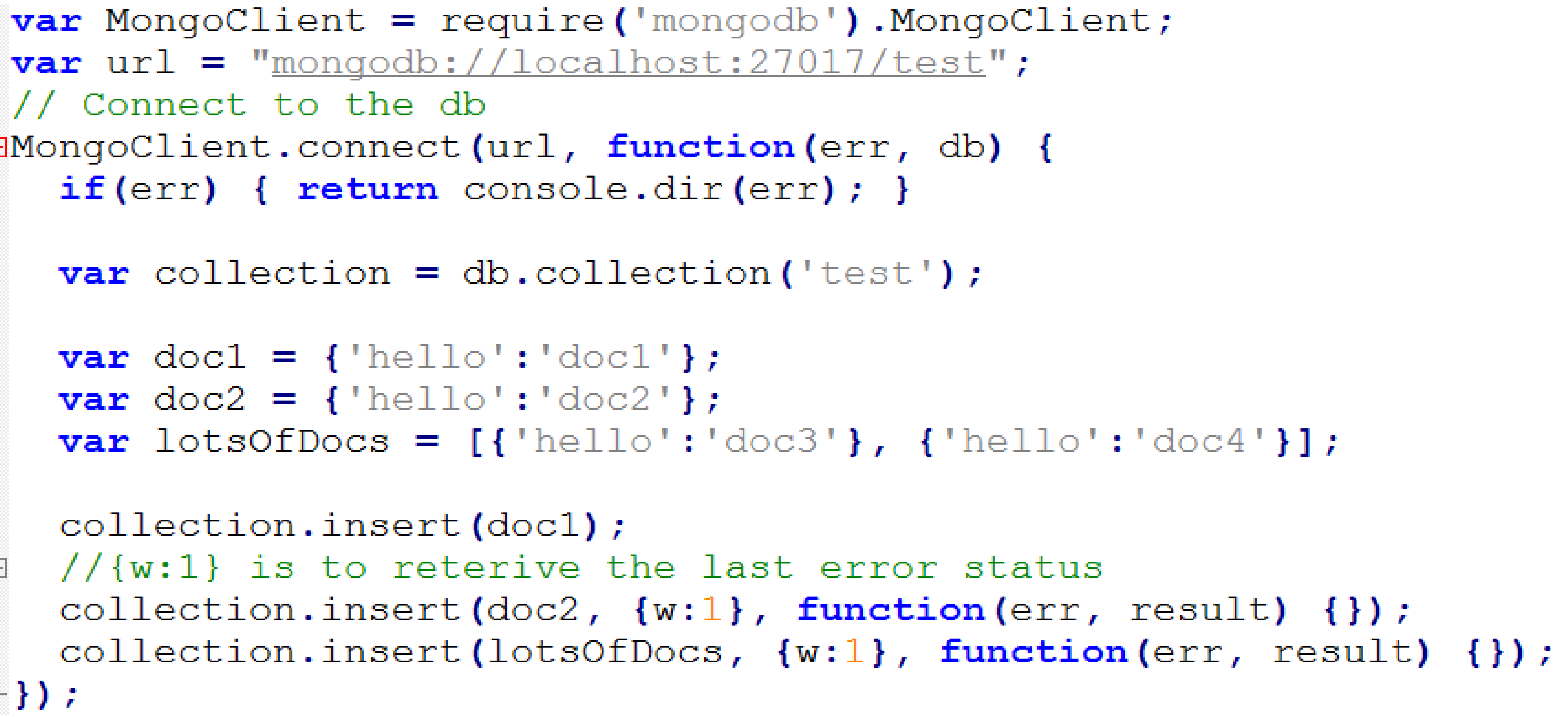
```
// Retrieve
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";

// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(!err) {
    console.log("We are connected");
  }
  else {
     console.log(err);
  }
});
```

MongoDB Collections



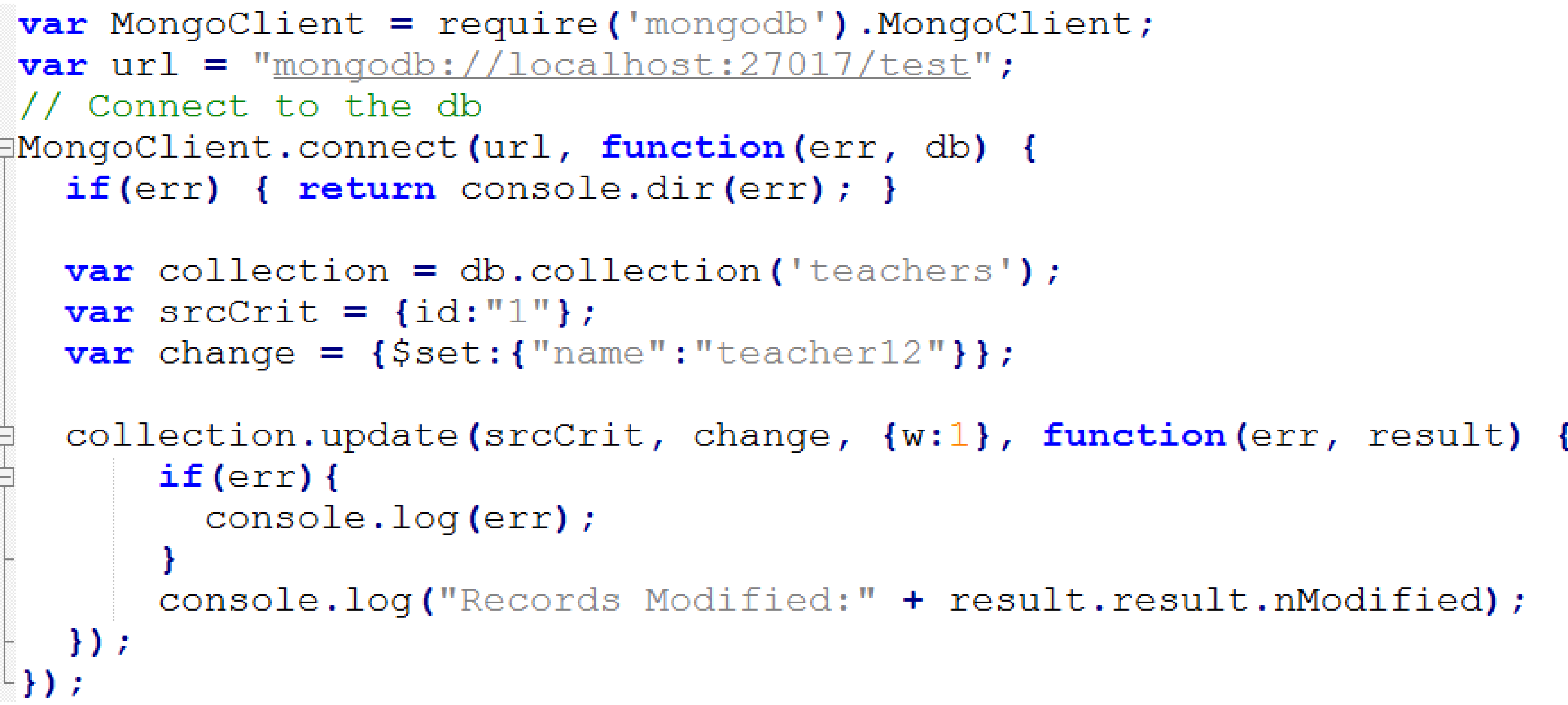
```
// Retrieve
var MongoClient = require('mongodb').MongoClient;
//url
var url = "mongodb://localhost:27017/test";
// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

   db.collection('test1', function(err, collection) {});
   db.collection('test2', {strict:true}, function(err, collection) {
     if(err) {console.log(err);}
   });

   db.createCollection('test3', function(err, collection) {});
   db.createCollection('test4', {strict:true}, function(err, collection)
     if(err) {console.log(err);}
   });
});
```

MongoDB CRUD-Insert



```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";
// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

  var collection = db.collection('test');

  var doc1 = {'hello':'doc1'};
  var doc2 = {'hello':'doc2'};
  var lotsOfDocs = [{'hello':'doc3'}, {'hello':'doc4'}];

  collection.insert(doc1);
  //{w:1} is to reterive the last error status
  collection.insert(doc2, {w:1}, function(err, result) {});
  collection.insert(lotsOfDocs, {w:1}, function(err, result) {});
});
```

MongoDB CRUD-Update

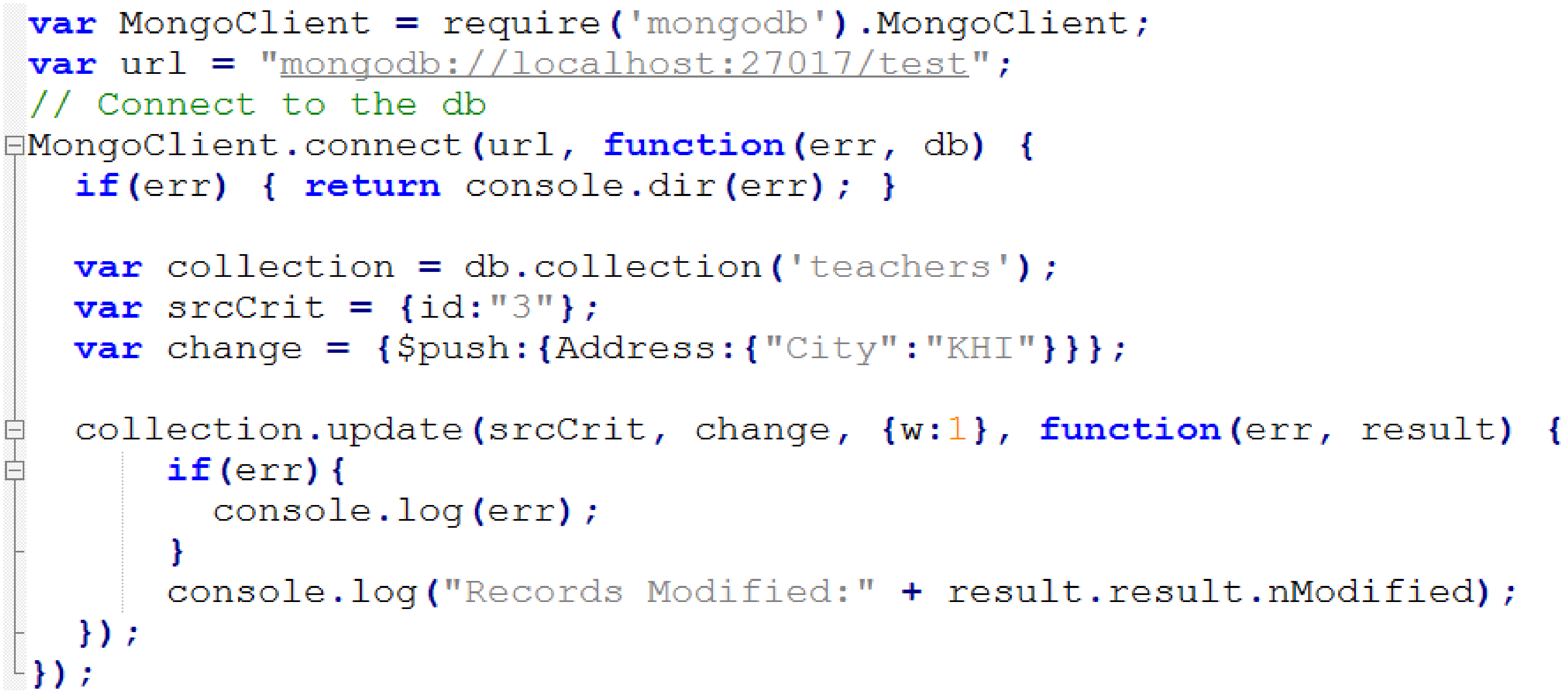


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";
// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

  var collection = db.collection('teachers');
  var srcCrit = {id:"1"};
  var change = {$set:{"name":"teacher12"}};

  collection.update(srcCrit, change, {w:1}, function(err, result) {
      if(err){
        console.log(err);
      }
      console.log("Records Modified:" + result.result.nModified);
  });
});
```

MongoDB CRUD-Update

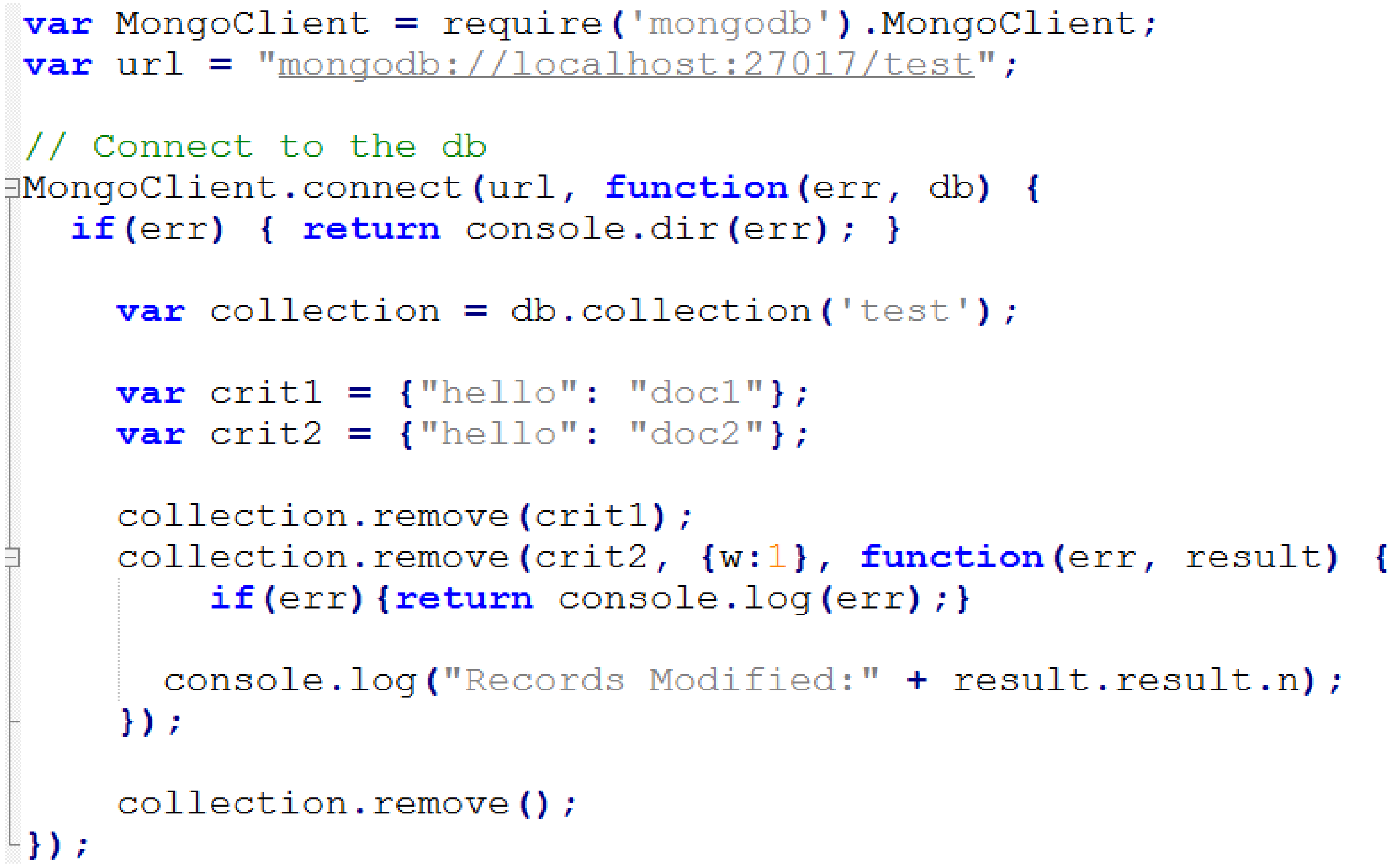
- $inc - increment a particular value by a certain amount
- $set - set a particular value
- $unset - delete a particular field (v1.3+)
- $push - append a value to an array
- $pushAll - append several values to an array
- $addToSet - adds value to the array only if its not in the array already
- $pop - removes the last element in an array
- $pull - remove a value(s) from an existing array
- $pullAll - remove several value(s) from an existing array
- $rename - renames the field
- $bit - bitwise operations

MongoDB CRUD-Update



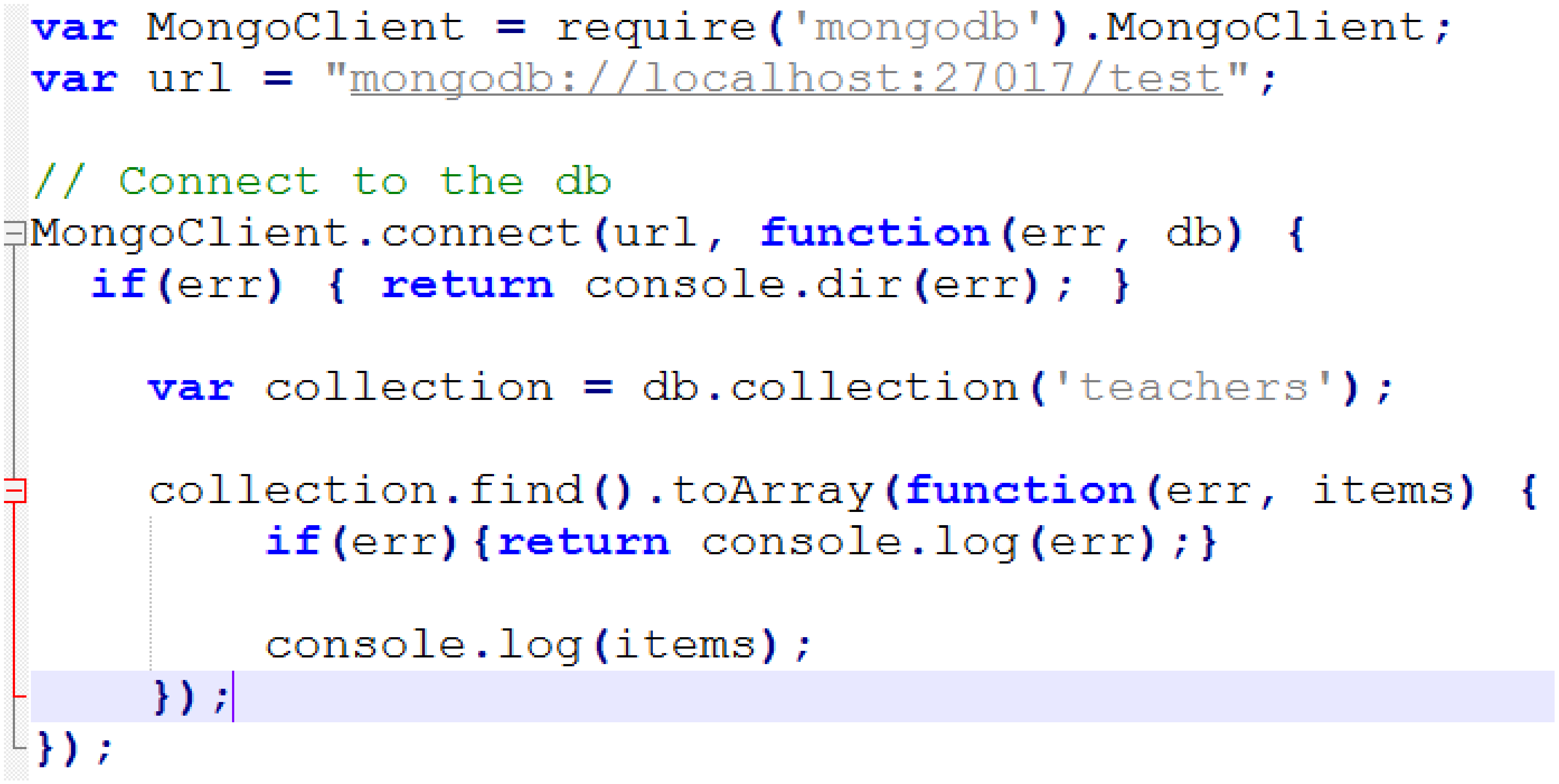
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";
// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

  var collection = db.collection('teachers');
  var srcCrit = {id:"3"};
  var change = {$push:{Address:{"City":"KHI"}}};

  collection.update(srcCrit, change, {w:1}, function(err, result) {
      if(err){
        console.log(err);
      }
      console.log("Records Modified:" + result.result.nModified);
  });
});
```

MongoDB CRUD-Remove



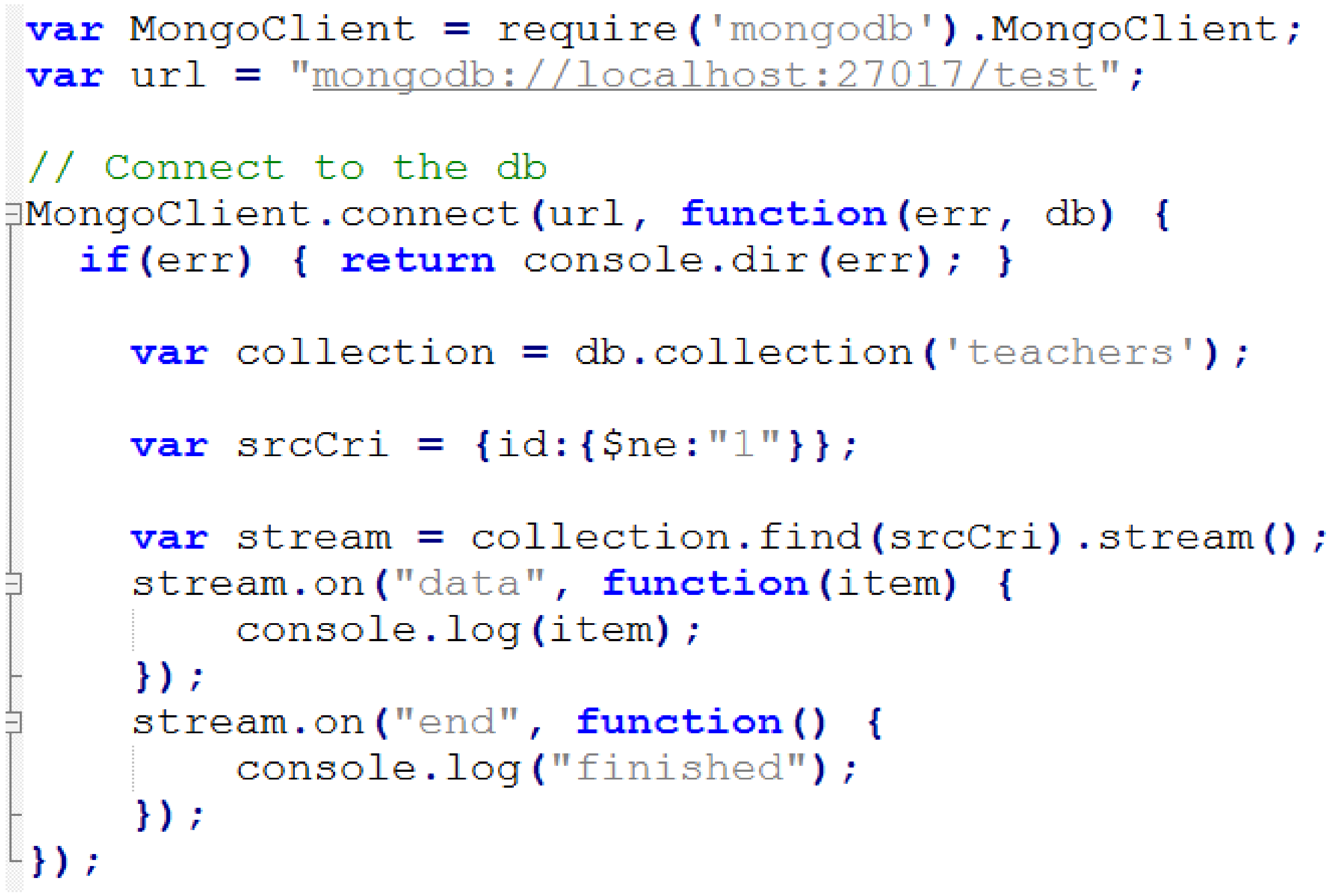
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";

// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

    var collection = db.collection('test');

    var crit1 = {"hello": "doc1"};
    var crit2 = {"hello": "doc2"};

    collection.remove(crit1);
    collection.remove(crit2, {w:1}, function(err, result) {
        if(err){return console.log(err);}

      console.log("Records Modified:" + result.result.n);
    });

    collection.remove();
});
```

MongoDB CRUD-Query



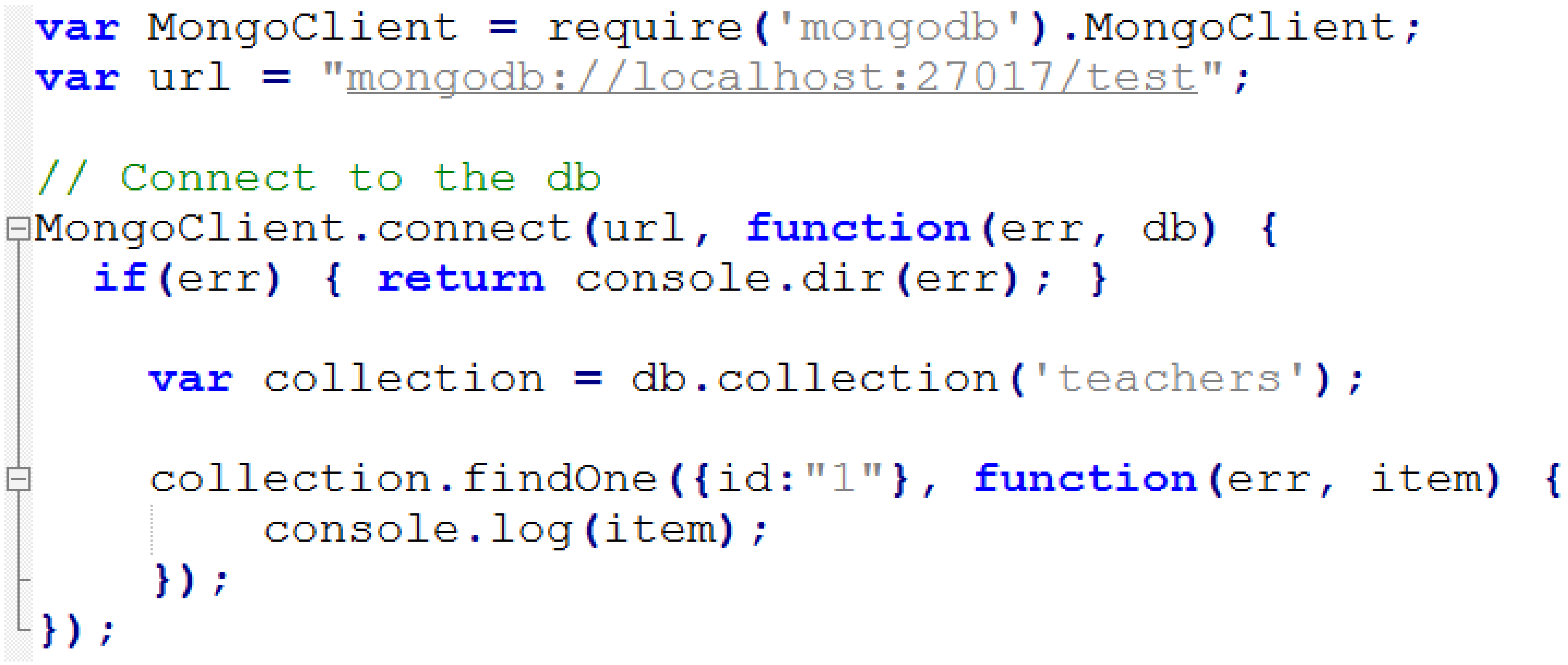
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";

// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

    var collection = db.collection('teachers');

    collection.find().toArray(function(err, items) {
        if(err){return console.log(err);}

        console.log(items);
    });
});
```

MongoDB CRUD-Query



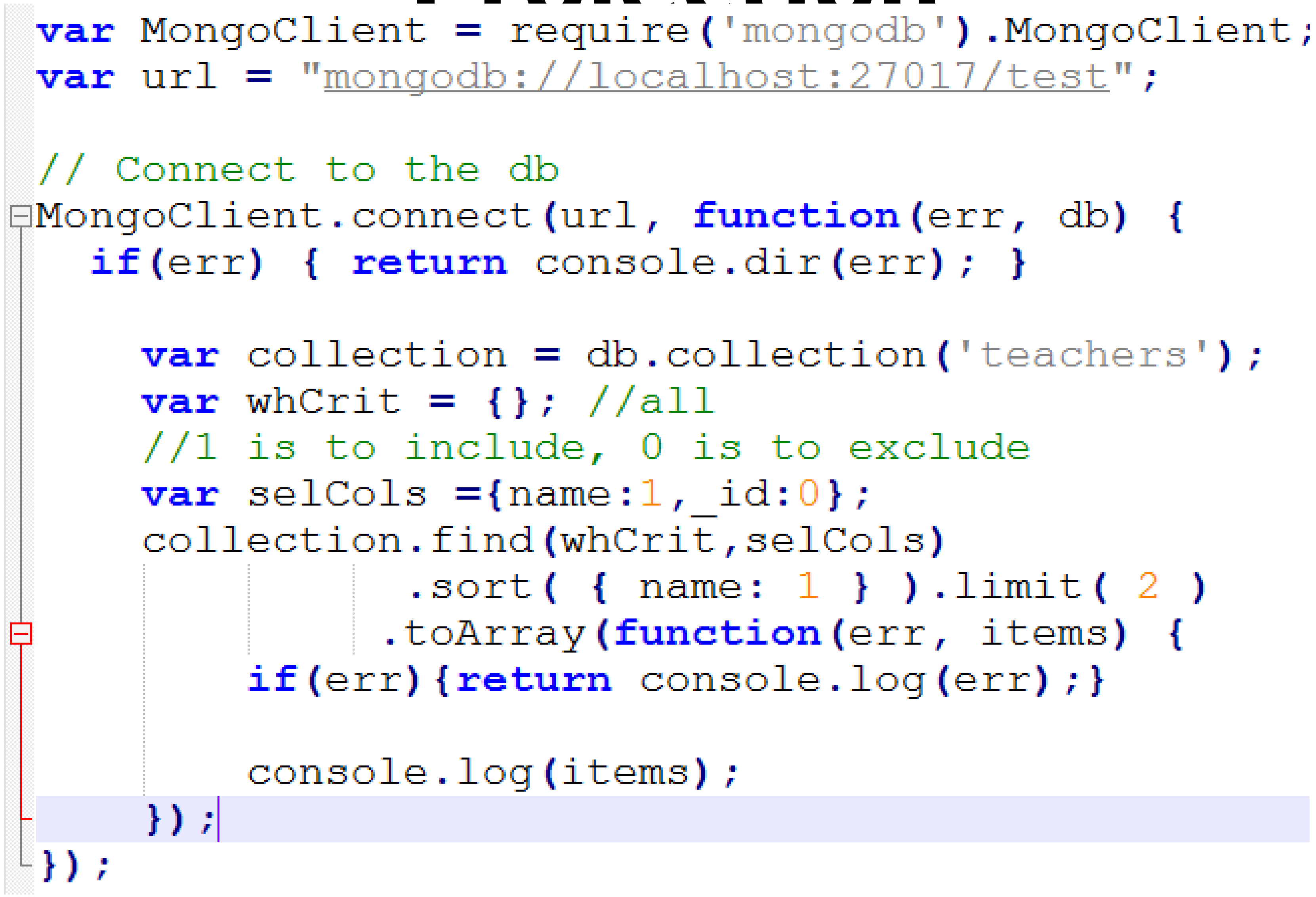
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";

// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

    var collection = db.collection('teachers');

    var srcCri = {id:{$ne:"1"}};

    var stream = collection.find(srcCri).stream();
    stream.on("data", function(item) {
        console.log(item);
    });
    stream.on("end", function() {
        console.log("finished");
    });
});
```

MongoDB CRUD-Query



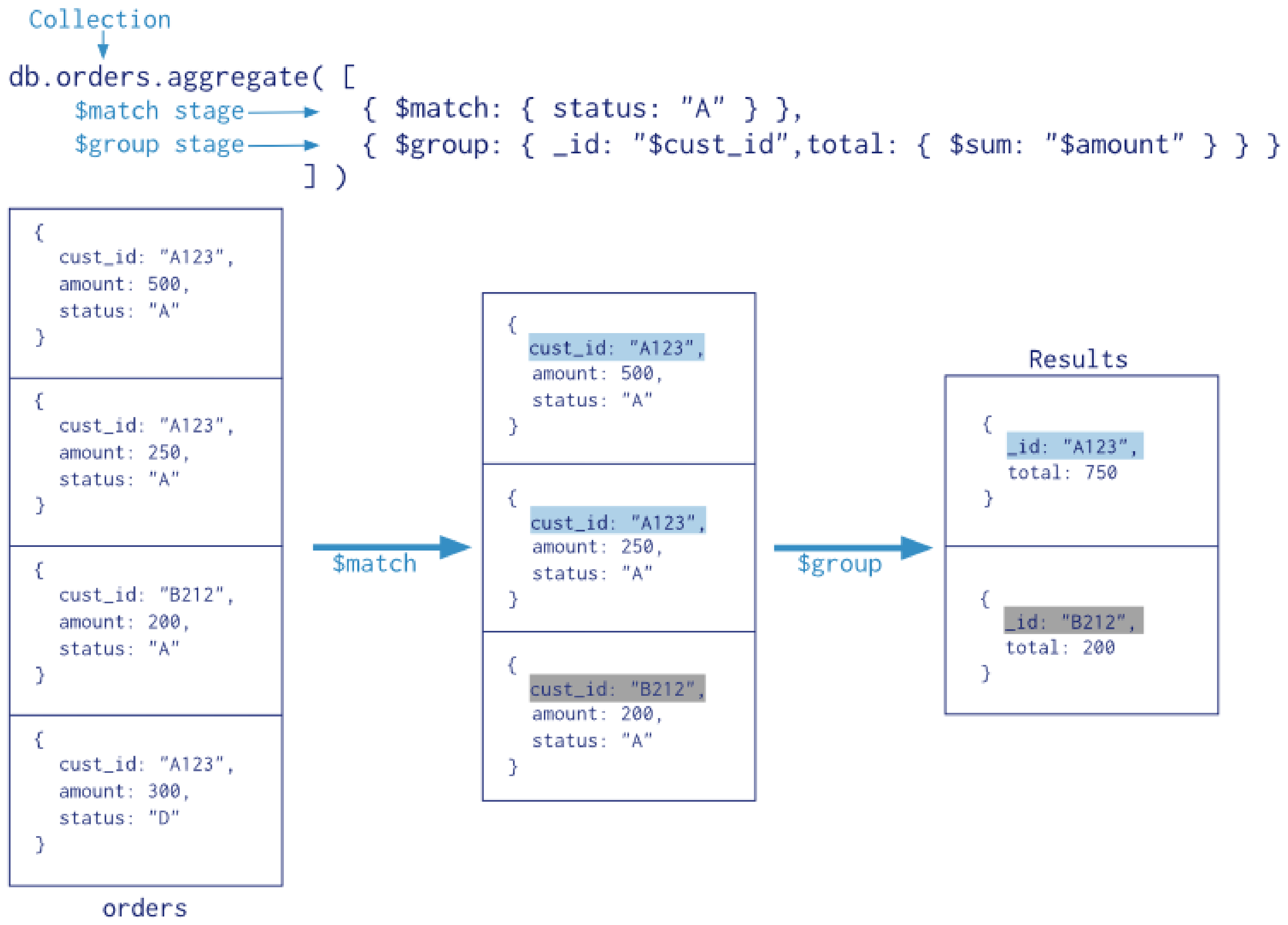
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";

// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

    var collection = db.collection('teachers');

    collection.findOne({id:"1"}, function(err, item) {
        console.log(item);
    });
});
```

MongoDB CRUD-Projection



```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/test";

// Connect to the db
MongoClient.connect(url, function(err, db) {
  if(err) { return console.dir(err); }

    var collection = db.collection('teachers');
    var whCrit = {}; //all
    //1 is to include, 0 is to exclude
    var selCols ={name:1,_id:0};
    collection.find(whCrit,selCols)
              .sort( { name: 1 } ).limit( 2 )
             .toArray(function(err, items) {
        if(err){return console.log(err);}

        console.log(items);
    });
});
```

MongoDB CRUD-Aggregation



```
db.orders.aggregate( [
      { $match: { status: "A" } },
      { $group: { _id: "$cust_id",total: { $sum: "$amount" } } }
] )
```

```
{
   cust_id: "A123",
   amount: 500,
   status: "A"
}
{
   cust_id: "A123",
   amount: 250,
   status: "A"
}
{
   cust_id: "B212",
   amount: 200,
   status: "A"
}
{
   cust_id: "A123",
   amount: 300,
   status: "D"
}
```

```
{
   cust_id: "A123",
   amount: 500,
   status: "A"
}
{
   cust_id: "A123",
   amount: 250,
   status: "A"
}
{
   cust_id: "B212",
   amount: 200,
   status: "A"
}
```

```
{
   _id: "A123",
   total: 750
}
{
   _id: "B212",
   total: 200
}
```

MongoDB - Indexing

- **ensureIndex()** method is used to create index
- db.teachers.ensureIndex({“name”:1})
  - Here we are creating an index on “name” field in ascending order in “teachers” collection.
  - To create index in descending order, use “-1”
- To create on multiple fields
  - db.teachers.ensureIndex({“name”:1,”Address”:-1})