

Exercise (Instructions): Node and MongoDB Part 2 | Coursera

Objectives and Outcomes

In this exercise you will continue to explore communicating from your Node application to the MongoDB server. At the end of this exercise you will be able to:

- Develop a Node module containing some common MongoDB operations
- Use the Node module in your application and communicate with the MongoDB server

Implementing a Node Module of Database Operations

- Create a new file named *operations.js* that contains a few MongoDB operations and add the following code:

```
var assert = require('assert');

exports.insertDocument = function(db, document, collection, callback) {
    // Get the documents collection
    var coll = db.collection(collection);
    // Insert some documents
    coll.insert(document, function(err, result) {
        assert.equal(err, null);
        console.log("Inserted " + result.result.n + " documents into the document
collection "
                    + collection);
        callback(result);
    });
};

exports.findDocuments = function(db, collection, callback) {
    // Get the documents collection
    var coll = db.collection(collection);

    // Find some documents
    coll.find({}).toArray(function(err, docs) {
        assert.equal(err, null);
        callback(docs);
    });
};
```

```

};

exports.removeDocument = function(db, document, collection, callback) {

  // Get the documents collection
  var coll = db.collection(collection);

  // Insert some documents
  coll.deleteOne(document, function(err, result) {
    assert.equal(err, null);
    console.log("Removed the document " + document);
    callback(result);
  });
};

exports.updateDocument = function(db, document, update, collection, callback)
{

  // Get the documents collection
  var coll = db.collection(collection);

  // Update document
  coll.updateOne(document
    , { $set: update }, null, function(err, result) {

    assert.equal(err, null);
    console.log("Updated the document with " + update);
    callback(result);
  });
};

```

Using the Node Module for Database Operations

- Create a new file named *server.js* and add the following code to it:

```

var MongoClient = require('mongodb').MongoClient,
    assert = require('assert');

var dboper = require('./operations');

// Connection URL

```

```

var url = 'mongodb://localhost:27017/conFusion';

// Use connect method to connect to the Server
MongoClient.connect(url, function (err, db) {
  assert.equal(null, err);
  console.log("Connected correctly to server");

  dboper.insertDocument(db, { name: "Vadonut", description: "Test" },
    "dishes", function (result) {
      console.log(result.ops);

      dboper.findDocuments(db, "dishes", function (docs) {
        console.log(docs);

        dboper.updateDocument(db, { name: "Vadonut" },
          { description: "Updated Test" },
          "dishes", function (result) {
            console.log(result.result);

            dboper.findDocuments(db, "dishes", function (docs) {
              console.log(docs)

              db.dropCollection("dishes", function (result) {
                console.log(result);

                db.close();
              });
            });
          });
        });
      });
    });
  });
});

```

- Run the server by typing the following at the prompt and observe the results:

```
node server
```

Conclusions

In this exercise you created a Node module to package some database operations, and then used the module to interact with the MongoDB server.

