

# Exercise (Instructions): Mongoose ODM I Coursera

## Objectives and Outcomes

In this exercise you will explore the Mongoose ODM and learn about creating schemas and interacting with the MongoDB database using Mongoose methods. At the end of this exercise, you will be able to:

- Install Mongoose ODM and connect to a MongoDB Server
- Create Mongoose Schemas
- Perform Database operations with Mongoose methods

## Installing Mongoose

- Create a folder named *node-mongoose* and move into the folder.
- In this folder, install Mongoose by typing the following at the prompt:

```
npm install mongoose --save
npm install assert --save
```

## Implementing a Node Application

- Create a sub-folder named *models* in the *node-mongoose* folder. Move to this folder.
- Create a file named *dishes-1.js* and add the following code to create a Mongoose schema:

```
// grab the things we need
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

// create a schema
var dishSchema = new Schema({
  name: {
    type: String,
    required: true,
    unique: true
  },
  description: {
    type: String,
    required: true
  }
}, {
```

```

    timestamps: true
  });

// the schema is useless so far
// we need to create a model using it
var Dishes = mongoose.model('Dish', dishSchema);

// make this available to our Node applications
module.exports = Dishes;

```

- Move to the *node-mongoose* folder and create a file named *server-1.js* and add the following code:

```

var mongoose = require('mongoose'),
    assert = require('assert');

var Dishes = require('./models/dishes-1');

// Connection URL
var url = 'mongodb://localhost:27017/conFusion';
mongoose.connect(url);
var db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function () {
  // we're connected!
  console.log("Connected correctly to server");

  // create a new user
  var newDish = Dishes({
    name: 'Uthapizza',
    description: 'Test'
  });

  // save the user
  newDish.save(function (err) {
    if (err) throw err;
    console.log('Dish created!');

    // get all the users
    Dishes.find({}, function (err, dishes) {
      if (err) throw err;

```

```

        // object of all the users
        console.log(dishes);
        db.collection('dishes').drop(function () {
            db.close();
        });
    });
});
});
});

```

- Make sure that your MongoDB server is up and running. Then at the terminal prompt type the following to start the server and see the result:

```
node server-1
```

- Now, create another file named *server-2.js* and add the following code to it:

```

var mongoose = require('mongoose'),
    assert = require('assert');

var Dishes = require('./models/dishes-1');
// Connection URL
var url = 'mongodb://localhost:27017/conFusion';
mongoose.connect(url);
var db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function () {
    // we're connected!
    console.log("Connected correctly to server");
    // create a new dish
    Dishes.create({
        name: 'Uthapizza',
        description: 'Test'
    }, function (err, dish) {
        if (err) throw err;
        console.log('Dish created!');
        console.log(dish);

        var id = dish._id;

        // get all the dishes
        setTimeout(function () {

```

```

        Dishes.findByIdAndUpdate(id, {
            $set: {
                description: 'Updated Test'
            }
        }, {
            new: true
        })
        .exec(function (err, dish) {
            if (err) throw err;
            console.log('Updated Dish!');
            console.log(dish);

            db.collection('dishes').drop(function () {
                db.close();
            });
        });
    }, 3000);
});
});

```

- Run this server on the console and see the result.

## Adding Sub-documents to a Document

- Create a file named *dishes-3.js* in the *models* folder and add the following code to it:

```

// grab the things we need
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var commentSchema = new Schema({
    rating: {
        type: Number,
        min: 1,
        max: 5,
        required: true
    },
    comment: {
        type: String,
        required: true
    },
    author: {

```

```

        type: String,
        required: true
    }
}, {
    timestamps: true
});

// create a schema
var dishSchema = new Schema({
    name: {
        type: String,
        required: true,
        unique: true
    },
    description: {
        type: String,
        required: true
    },
    comments:[commentSchema]
}, {
    timestamps: true
});

// the schema is useless so far
// we need to create a model using it
var Dishes = mongoose.model('Dish', dishSchema);

// make this available to our Node applications
module.exports = Dishes;

```

- Create another variation of the server named *server-3.js* and add the following code to it:

```

var mongoose = require('mongoose'),
    assert = require('assert');

var Dishes = require('./models/dishes-3');

// Connection URL
var url = 'mongodb://localhost:27017/conFusion';mongoose.connect(url);
var db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));

```

```

db.once('open', function () {
  // we're connected!
  console.log("Connected correctly to server");

  // create a new dish
  Dishes.create({
    name: 'Uthapizza',
    description: 'Test',
    comments: [
      {
        rating: 3,
        comment: 'This is insane',
        author: 'Matt Daemon'
      }
    ]
  }, function (err, dish) {
    if (err) throw err;
    console.log('Dish created!');
    console.log(dish);

    var id = dish._id;

    // get all the dishes
    setTimeout(function () {
      Dishes.findByIdAndUpdate(id, {
        $set: {
          description: 'Updated Test'
        }
      }, {
        new: true
      })
      .exec(function (err, dish) {
        if (err) throw err;
        console.log('Updated Dish!');
        console.log(dish);

        dish.comments.push({
          rating: 5,
          comment: 'I\'m getting a sinking feeling!',
          author: 'Leonardo di Carpaccio'
        });
      });
    }, 1000);
  });
});

```

```
        dish.save(function (err, dish) {
            console.log('Updated Comments!');
            console.log(dish);

            db.collection('dishes').drop(function () {
                db.close();
            });
        });
    }, 3000);
});
});
```

- Run the server and observe the result.

## Conclusions

In this exercise you learnt about the Mongoose module and used it to create schemas and interact with the MongoDB server.