Mongoose ODM

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

MongoDB

- MongoDB stores data in the form of documents
- No structure imposed on the document
  - Any document can be stored in any collection
  - Relies on developer’s discipline to maintain the structure of the documents

Mongoose

- Mongoose ODM
  - Object Data Model
  - Object Document Mapping
  - Object relational mapping (ORM)
- Adds structure to MongoDB documents through schema

Connecting to MongoDB

```
var mongoose = require('mongoose’);

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 () {
    console.log("Connected correctly to server");
    . . .
});
```

Mongoose Schema

- Schema:
  - Structure of the data to be stored
  - Defines all the fields of your document and their types
    - Can do validation
- Schema types: String, Number, Date, Buffer, Boolean, Mixed, ObjectId, Array
- Schema is used to create a Model function

Schema Example

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var dishSchema = new Schema({
   name: { type: String, required: true, unique: true},
   description: { type: String, required: true}},
   {timestamps: true});

var Dishes = mongoose.model('Dish', dishSchema);
```

Mongoose-based Database Operations

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

 var newDish = Dishes({
    name: 'Uthapizza',
    description: 'Test'
 });

 newDish.save(function (err) {
    if (err) throw err;

    Dishes.find({}, function (err, dishes) {
        if (err) throw err;

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

Embedded Documents Schema

```
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 });

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

var Dishes = mongoose.model('Dish', dishSchema);
```

Creating New Document

```
// 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;

    . . .
});
```

Adding an Embedded Document

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

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

     dish.save(function (err, dish) {

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

Exercise: Mongoose ODM

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