

Node.js + MongoDB

Mongoose

MODULE 9

NODE.JS AND MONGODB INTEGRATION

Mongoose

- > Mongoose is a Node.js library
- > It provides MongoDB object mapping similar to ORM with a familiar interface within Node.js.
- > Mongoose translates data in the database to JavaScript objects for use in your application.

Connecting to MongoDB

```
var db= mongoose.connection;
db.on('error', console.error);
db.once('open', function(){
  // create your schemas here
});

mongoose.connect(
  'mongodb://localhost:27017/world'
);
```

Schemas and Models

> Schemas define the structure of documents within a collection and models are used to create instances of data that will be stored in documents.

```
var citySchema = new mongoose.Schema({
  "district":String,
  "_id":mongoose.Schema.Types.ObjectId,
  "name":String,
  "population":Number
});
```

Schemas and Models

```
var countrySchema = new mongoose.Schema({
  "_id":mongoose.Schema.Types.ObjectId,
  "capital":Number,
  "cities":[ citySchema ],
  "continent":String,
  "gnp":Number,
  "governmentForm":String,
  "headOfState":String,
  "lifeExpectancy":Number,
  "localName":String,
  "name":String,
  "population":Number,
  "region":String,
  "surfaceArea":Number
});
var Country = mongoose.model('countries', countrySchema);
```

Create, Retrieve, Update, and Delete

```
var turkey = new Country({
  "_id": "TUR",
  "capital": 3358,
  "continent": "Asia",
  "gnp": 20000,
  "governmentForm": "Republic",
  "indepYear": 2013,
  "lifeExpectancy": 95,
  "localName": "Turkiye",
  "name": "Turkey",
  "population": 1,
  "region": "Middle East",
  "surfaceArea": 774828
});
turkey.save(function(err, centry){
  if (err) return console.error(err);
  console.dir(centry);
})
```

Create, Retrieve, Update, and Delete

```
Country.find(  
  {"continent": "Asia"},  
  function (err, asianCountries) {  
    console.log(asianCountries);  
  }  
);
```

Create, Retrieve, Update, and Delete

```
Country.update(  
  {"continent": "Asia"},  
  { $inc: { population: 1 } },  
  { multi: true },  
  function (err, numberOfEffecteds) {  
    if (err) return console.err(err);  
    console.log(numberOfEffecteds);  
  }  
);
```

Create, Retrieve, Update, and Delete

```
Country.remove(  
  {"_id": "FUR"},  
  function(err){  
    if (err) return console.err(err);  
    console.log("FUR is removed!");  
  });
```

distinct

```
Country.distinct('continent',  
  function (err, continents) {  
    if (err) return console.err(err);  
    console.log(continents);  
  });
```

group

```
var group = {
  key: {"continent": 1},
  cond: {},
  reduce: function(country, stat) {
    stat.count++;
    stat.avg_pop += country.population;
  },
  initial: {
    avg_pop: 0,
    count: 0
  },
  finalize: function(statistics) {
    stat.avg_pop = stat.avg_pop / stat.count;
  }
};
Country.collection.group(group.key, group.cond,
group.initial, group.reduce, group.finalize, true,
function(err, results) {
  console.log('group results %j', results);
});
```