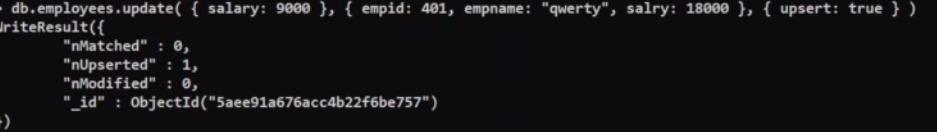
UPDATE:



Only First Record Gets Updated:





For Updating Multiple Records:



REMOVING:

By Default Removing happens for Multiple Records



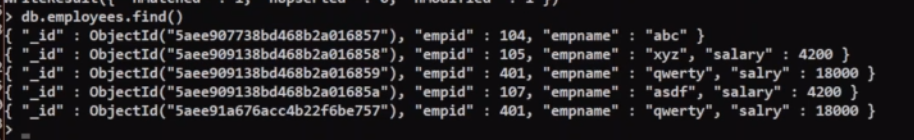
If we wanted it Happen for only one Record, Then



REMOVING SINGLE PROPERTY IN A ROW:

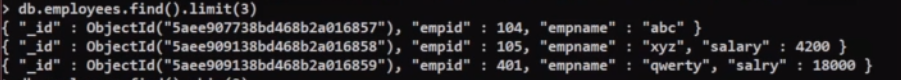


Result:



LIMIT, SKIP, SORT

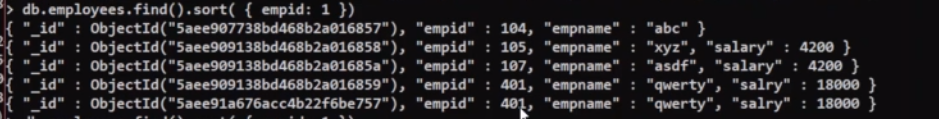
First 3 Records are displayed



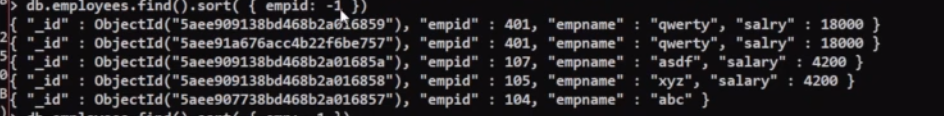
First 3 Record are skipped and remaining are seen:



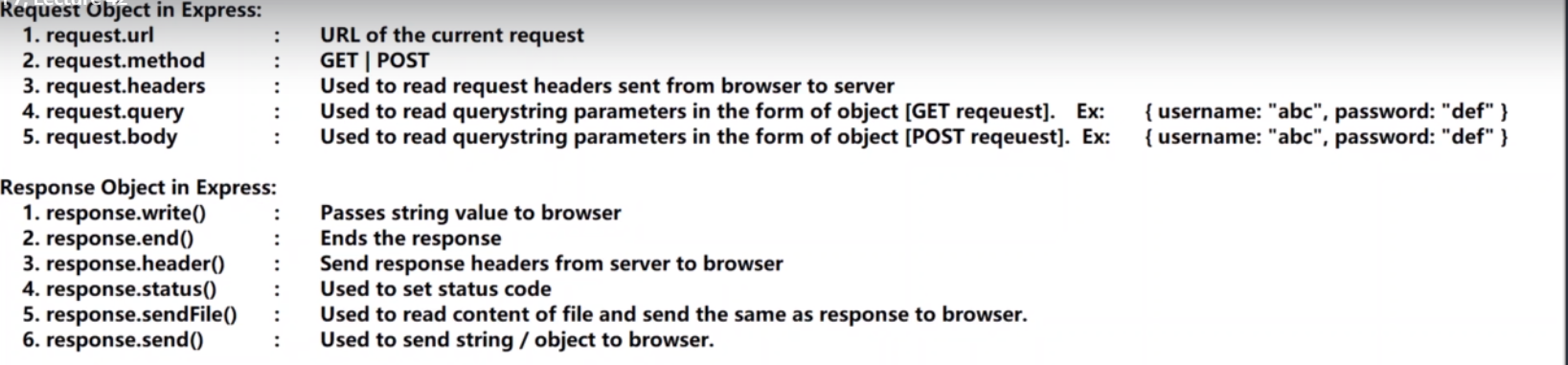
Sorted Order: Ascending Order



Sorted Order: Descending Order



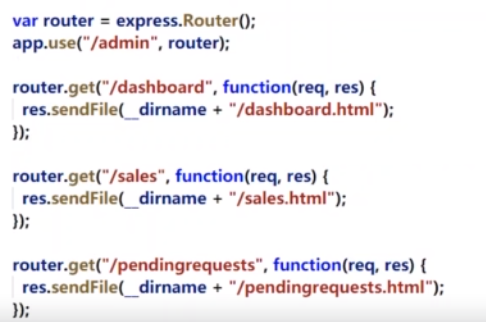
Express Request & Response Objects:



Console.log(request.query)🡪Sample URL: <http://localhost:7000/?x=10&y=30>

O/P:

{x:’10’, y:’30’}



Retrieve.js

var mongoose = require('mongoose');

var mongooseSchema = mongoose.Schema({ empId: Number, empName: String, salary: Number });

var mongooseModel = mongoose.model('employees', mongooseSchema);

mongoose.connect('mongodb://localhost/company');

mongooseModel.find({ salary: { $gt: 200000 } }, callBackFunction)

function callBackFunction(error, data) {

console.log(data[0].empName);

console.log(error);

mongoose.disconnect();

}

Update.js

var mongoose = require('mongoose');

var mongooseSchema = mongoose.Schema({ empId: Number, empName: String, salary: Number });

var mongooseModel = mongoose.model('employees', mongooseSchema);

mongoose.connect('mongodb://localhost/company');

var dataRetrived = mongooseModel.find({ empName: 'Samir' }, callBack1)

function callBack1(error1, data) {

console.log(data);

console.log(error1);

data[0].salary = 2000000;

data[0].save(callBack2);

function callBack2(error2) {

if (error2 == null) {

console.log('Updated..');

}

mongoose.disconnect();

}

}

Insert.js:

var mongoose = require('mongoose');

var employeeSchema = mongoose.Schema({ empId: Number, empName: String, salary: Number }, { versionKey: false });

var employeeModel = mongoose.model('employees', employeeSchema);

mongoose.connect('mongodb://localhost/company');

var newModel = new employeeModel({ empId: 107, empName: 'Varija', salary: 600000 });

newModel.save(callBackFunction);

function callBackFunction(error, data) {

console.log(data);

}

Delete.js:

var mongoose = require('mongoose');

var mongooseSchema = mongoose.Schema({ empId: Number, empName: String, salary: Number });

var mongoModel = mongoose.model('employees', mongooseSchema);

mongoose.connect('mongodb://localhost/company');

mongoModel.remove({ empId: 101 }, callBackFunction);

function callBackFunction(error, data) {

if (error == null) {

console.log('data Removed..')

}

mongoose.disconnect();

}