



Node.js + MongoDB

권기웅

1. MongoDB 기동

1.1 MongoDB 기동

```
C:\Windows\system32\cmd.exe - mongod --dbpath d:\kwon\db

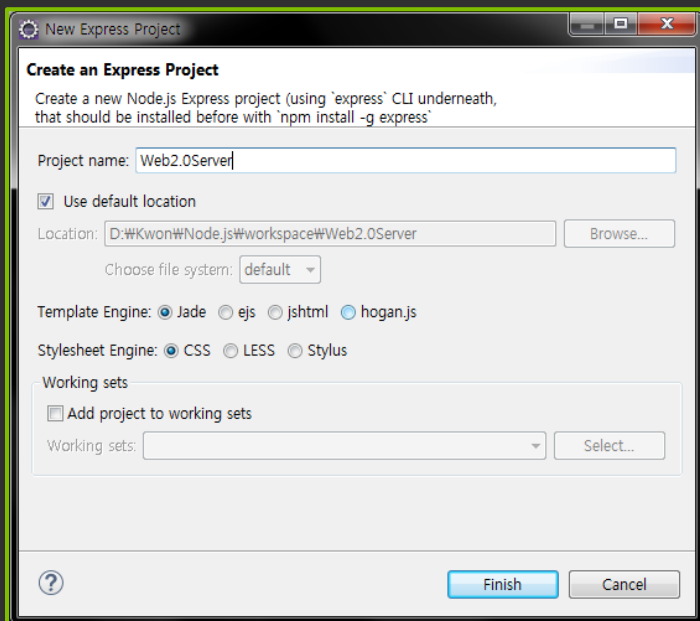
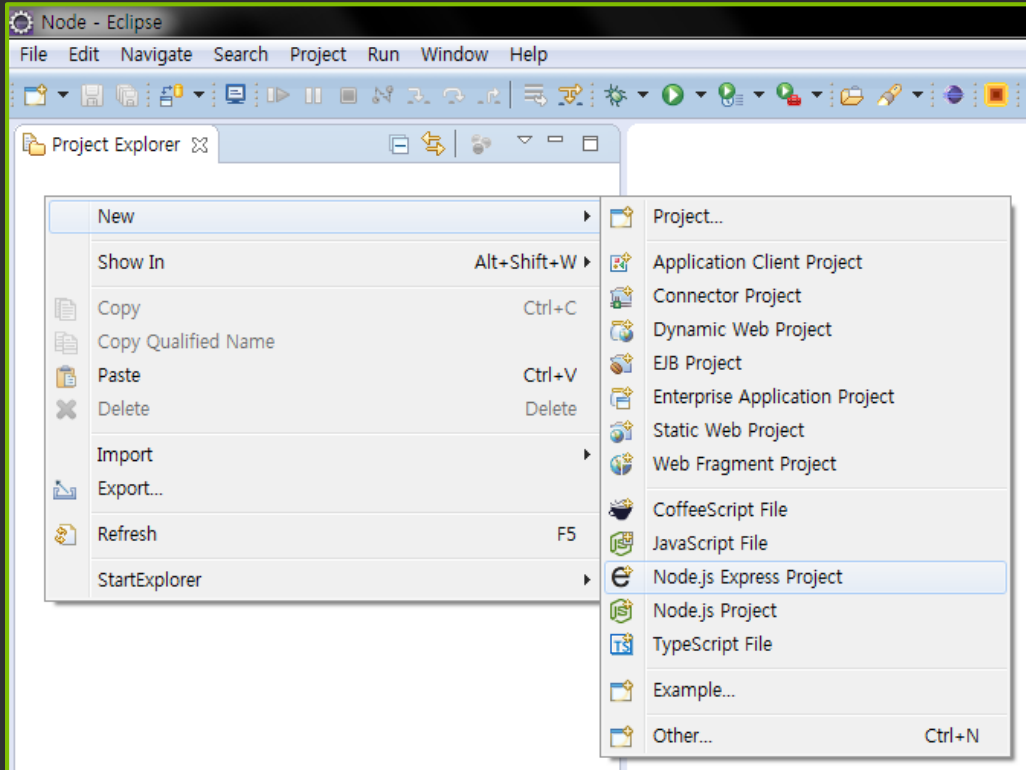
C:\>cd C:\Program Files\MongoDB\Server\3.0\bin

C:\Program Files\MongoDB\Server\3.0\bin>mongod --dbpath d:\kwon\db
2015-07-27T13:32:27.096+0900 I CONTROL  Hotfix KB2731284 or later update is not
installed, will zero-out data files
2015-07-27T13:32:27.254+0900 I JOURNAL  [initandlisten] journal dir=d:\kwon\db\j
ournal
```



2. 프로젝트 작성

2.1 Node.js Express Project 작성



2.2 MongoDB 연결 패키지 mongojs 설치

2.2.1 명령 프롬프트에서 현재 프로젝트 위치로 이동, mongojs 패키지 설치

npm install mongojs

```
C:\Windows\system32\cmd.exe

D:\Kwon\Node.js\workspace\Web2.0Server

D:\Kwon\Node.js\workspace\Web2.0Server>npm install mongojs
-
> kerberos@0.0.12 install D:\Kwon\Node.js\workspace\Web2.0Server\node_modules\mongojs\node_modules\mongodb-core\node_modules\kerberos
> (node-gyp rebuild 2> builderror.log) || (exit 0)
```

2.2.2 명령 프롬프트 온 감에 실행도

```
C:\Windows\system32\cmd.exe - nodemon app.js

D:\Kwon\Node.js\workspace\Web2.0Server>nodemon app.js
27 Jul 13:42:21 - [nodemon] v1.3.8
27 Jul 13:42:21 - [nodemon] to restart at any time, enter `rs`
27 Jul 13:42:21 - [nodemon] watching: *.*
27 Jul 13:42:21 - [nodemon] starting `node app.js`
27 Jul 13:42:22 - [nodemon] clean exit - waiting for changes before restart
```



2.3 요청 받기

2.3.1 SELECT

```
app.get('/getAllProduct', function(req,res){
  // var db = require('mongojs')('IP주소/DB명', ['사용할 배열1', '사용할 배열2', ...]);
  var db = require('mongojs')('172.16.101.141/kwon', ['product']);
  var cb = req.query.callback;

  //MongoDB기본 문법(function(err,data){});
  db.product.find(function(err, data){
    res.send(cb+'('+JSON.stringify(data)+')');
  });
});

app.get('/getProduct', function(req,res){
  var db = require('mongojs')('172.16.101.141/kwon', ['product']);
  var cb = req.query.callback;
  var p = req.query.price * 1;

  db.product.find({price: {'$gte': p}}, function(err, data){
    res.send(cb+'('+JSON.stringify(data)+')');
  });
});
```

2.3.2 INSERT

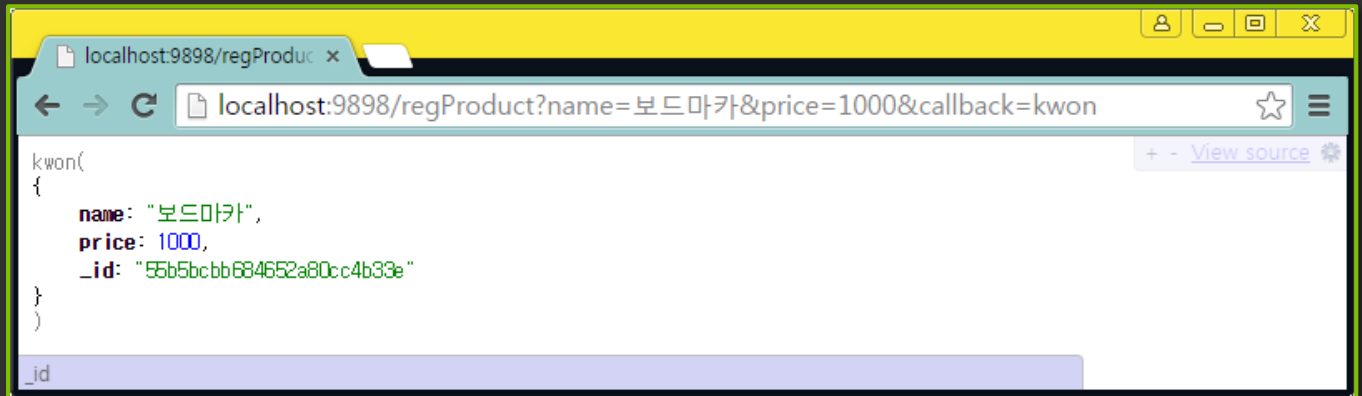
```
app.get('/regProduct', function(req,res){
  var db = require('mongojs')('172.16.101.141/kwon', ['product']);
  var cb = req.query.callback;
  var n = req.query.name;
  var p = req.query.price * 1;

  db.product.save({name:n, price:p}, function(err, data){
    res.send(cb+'('+JSON.stringify(data)+')');
  });
});
```

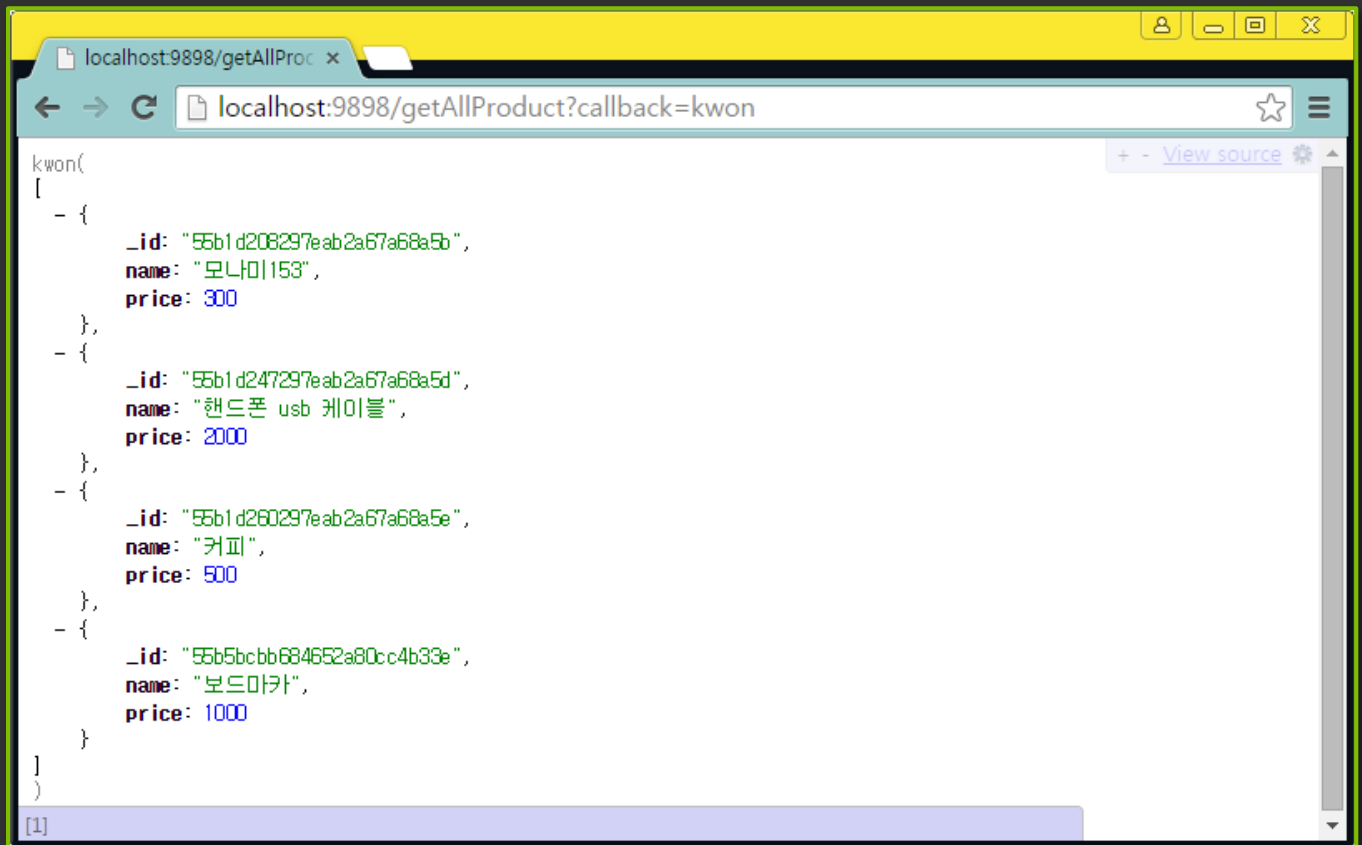


3. 실행

3.1 실행

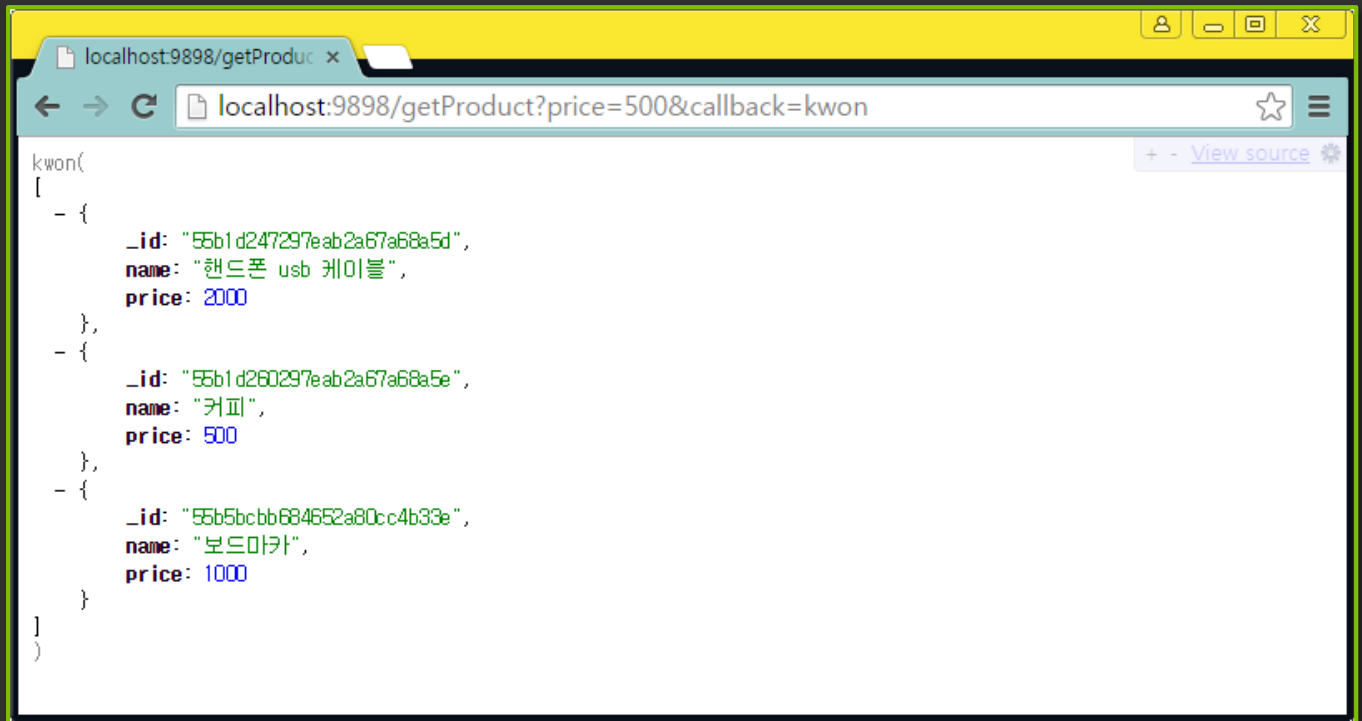


```
kwon(
{
  name: "보드마카",
  price: 1000,
  _id: "55b5bcb684652a80cc4b33e"
})
_id
```



```
kwon(
[
  - {
    _id: "55b1d208297eab2a67a68a5b",
    name: "모나미 153",
    price: 300
  },
  - {
    _id: "55b1d247297eab2a67a68a5d",
    name: "핸드폰 usb 케이블",
    price: 2000
  },
  - {
    _id: "55b1d260297eab2a67a68a5e",
    name: "커피",
    price: 500
  },
  - {
    _id: "55b5bcb684652a80cc4b33e",
    name: "보드마카",
    price: 1000
  }
]
)
[1]
```





```
kwon([
  - {
    _id: "55b1d247297eab2a67a68a5d",
    name: "핸드폰 usb 케이블",
    price: 2000
  },
  - {
    _id: "55b1d260297eab2a67a68a5e",
    name: "커피",
    price: 900
  },
  - {
    _id: "55b5bcbb684652a80cc4b33e",
    name: "보드마카",
    price: 1000
  }
])
```

