

Ejercicio Grupal de JS: FETCH+MEJSN.

PREGUNTAS

Usamos ES6/Node. Disponga del demo MEJSN corriendo debidamente. Considere el modelo `fetch.js` que se le entrega. Estudie [node-fetch](#). Logre el siguiente caso de uso modificando el modelo dado.

Instale `node-fetch`

```
22/08/2017 14:09 <DIR> .
22/08/2017 14:09 <DIR> ..
22/08/2017 15:32      1.163 fetch.js
22/08/2017 16:19      407 package.json
22/08/2017 16:16      971 package-lock.json
                3 archivos      2.541 bytes
                2 dirs      19.628.802.048 bytes libres

FETCH:npm install --save
added 4 packages in 3.111s
```

Levante mongod (asumimos que db bears existe y tiene la colección bears)

```
Simbolo del sistema - mongod
MONGODB:mongod
2017-08-22T16:21:21.594-0600 I CONTROL [initandlisten] MongoDB starting : pid=4644 port=27017 dbpath=C:\data\db\ 64-bit host=CarlosLoria-HP
2017-08-22T16:21:21.595-0600 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2017-08-22T16:21:21.595-0600 I CONTROL [initandlisten] db version v3.2.7
2017-08-22T16:21:21.595-0600 I CONTROL [initandlisten] git version: 4249c1d2b5999ebbf1fdf3b60e0e3b3ff5c0aaf2
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1p-fips
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten] allocator: tcmalloc
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten] modules: none
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten] build environment:
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten]   distmod: 2008plus-ssl
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten]   distarch: x86_64
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten]   target_arch: x86_64
2017-08-22T16:21:21.596-0600 I CONTROL [initandlisten] options: {}
2017-08-22T16:21:21.598-0600 I - [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'
2017-08-22T16:21:21.599-0600 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=10000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0);
2017-08-22T16:21:22.419-0600 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2017-08-22T16:21:22.419-0600 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'C:/data/db/diagnostic.data'
2017-08-22T16:21:22.423-0600 I NETWORK [initandlisten] waiting for connections on port 27017
```

Levante su app `mej sn`

```

15/08/2017 10:20 <DIR> .
15/08/2017 10:20 <DIR> ..
04/08/2017 11:00 543 README.md
15/08/2017 21:16 3.930 server.js
04/08/2017 10:57 <DIR> app
04/08/2017 11:05 <DIR> credits
04/08/2017 21:43 <DIR> public
15/08/2017 11:28 <DIR> node_modules
22/08/2017 13:32 469 package.json
22/08/2017 13:32 16.598 package-lock.json
4 archivos 21.540 bytes
6 dirs 19.627.393.024 bytes libres

SERVER:set PORT=9090

SERVER:node server.js
*** Server is up and running on port 9090 ***

```

1. Abra consola en carpeta fetch (use el mismo puerto del paso previo). Debe funcionar esto (note que se oyen unos beeps)

```

FETCH:set PORT=9090

FETCH:node fetch.js
API will be tested
{ message: '*** Rest API is working fine! ***' }

FETCH:

```

Logre esto:

```

FETCH:mongo
MongoDB shell version: 3.2.7
connecting to: test
> use bears
switched to db bears
> db.bears.drop()
true
> db.bears.find()
> quit()

FETCH:node fetch.js boo-boo-1
"boo-boo-1" will be inserted
{ message: 'Bear created!', bearId: '599cb11c5cea8514a099940b' }

FETCH:node fetch.js boo-boo-2
"boo-boo-2" will be inserted
{ message: 'Bear created!', bearId: '599cb11f5cea8514a099940c' }

FETCH:node fetch.js boo-boo-3
"boo-boo-3" will be inserted
{ message: 'Bear created!', bearId: '599cb1225cea8514a099940d' }

FETCH:mongo
MongoDB shell version: 3.2.7
connecting to: test
> use bears
switched to db bears
> db.bears.find()
{ "_id" : ObjectId("599cb11c5cea8514a099940b"), "name" : "boo-boo-1", "__v" : 0 }
{ "_id" : ObjectId("599cb11f5cea8514a099940c"), "name" : "boo-boo-2", "__v" : 0 }
{ "_id" : ObjectId("599cb1225cea8514a099940d"), "name" : "boo-boo-3", "__v" : 0 }
>

```

2. Provoque un retraso en el server.js y pruebe que los beeps siguen sonando

```
router.get('/', function(req, res) {  
  setTimeout( () =>  
    res.json({ message: '*** Rest API is working fine! ***' }),  
    5000)  
});
```

```
FETCH:node fetch.js  
API will be tested  
{ message: '*** Rest API is working fine! ***' }  
FETCH:
```