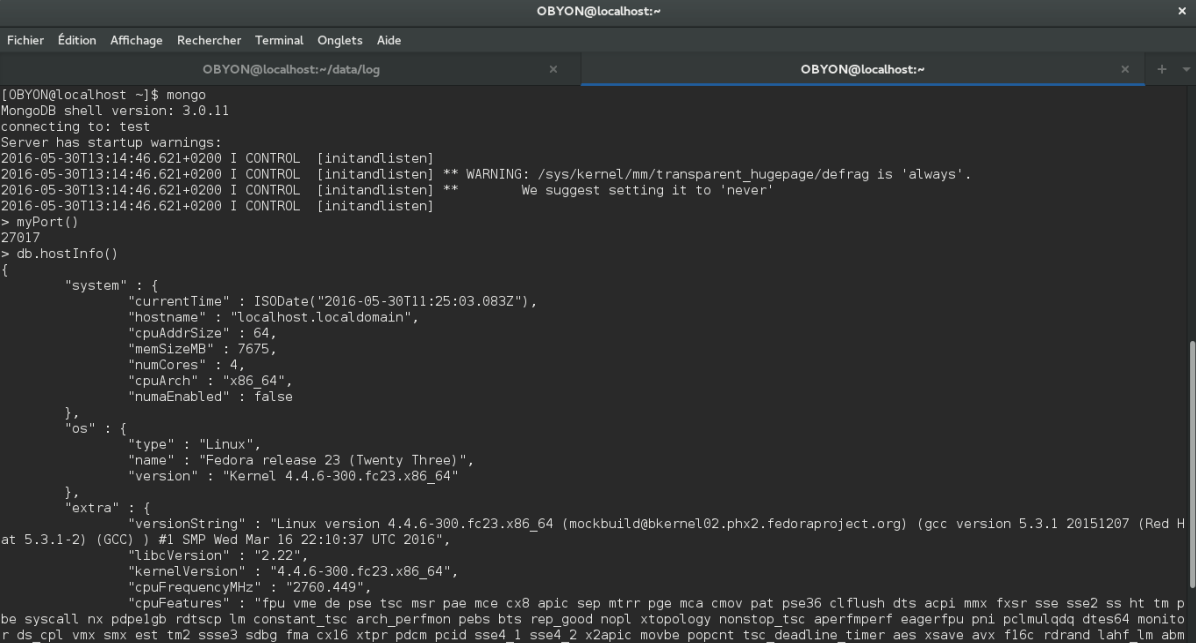


TP MONGODB

PARTIE N°1

1. `$ sudo mongod` (éventuellement “service mongod stop”)
`$ mongo`
`$ myPort()`
`$ db.hostInfo()`



```
OBYON@localhost:~$ mongo
MongoDB shell version: 3.0.11
connecting to: test
Server has startup warnings:
2016-05-30T13:14:46.621+0200 I CONTROL [initandlisten]
2016-05-30T13:14:46.621+0200 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2016-05-30T13:14:46.621+0200 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2016-05-30T13:14:46.621+0200 I CONTROL [initandlisten]
> myPort()
27017
> db.hostInfo()
{
  "system": {
    "currentTime": ISODate("2016-05-30T11:25:03.083Z"),
    "hostname": "localhost.localdomain",
    "cpuAddrSize": 64,
    "memSizeMB": 7675,
    "numCores": 4,
    "cpuArch": "x86_64",
    "numaEnabled": false
  },
  "os": {
    "type": "Linux",
    "name": "Fedora release 23 (Twenty Three)",
    "version": "Kernel 4.4.6-300.fc23.x86_64"
  },
  "extra": {
    "versionString": "Linux version 4.4.6-300.fc23.x86_64 (mockbuild@bkernel02.phx2.fedoraproject.org) (gcc version 5.3.1 20151207 (Red H
at 5.3.1-2) (GCC) ) #1 SMP Wed Mar 16 22:10:37 UTC 2016",
    "libcVersion": "2.22",
    "kernelVersion": "4.4.6-300.fc23.x86_64",
    "cpuFrequencyMHz": "2760.449",
    "cpuFeatures": "fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm p
be syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monito
r ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm
```

2. `$ use admin`
`$db.shutdownServer()`

3. `$ sudo mongod --dbpath /home/OBYON/data/db --logpath`
`/home/OBYON/data/log/mongod.log`

\$ db.serverCmdLineOpts()

```
OBYON@localhost:~  
Fichier Édition Affichage Rechercher Terminal Onglets Aide  
OBYON@localhost:~/data/log  
at 5.3.1-2) (GCC) ) #1 SMP Wed Mar 16 22:10:37 UTC 2016",  
    "libcVersion" : "2.22",  
    "kernelVersion" : "4.4.6-300.fc23.x86_64",  
    "cpuFrequencyMHz" : "2509.667",  
    "cpuFeatures" : "fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts api mmx fxsr sse sse2 ss ht tm p  
be syscall nx pdpe1gb rdtscp lm constant tsc arch_perfmon pebs bts rep good nopl xtopology nonstop tsc_aperfmperf eagerfpu pni pclmulqdq dtes64 monito  
r ds_cpl vmx smx est tm2 ssse3 sbdb fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm  
ida arat epb pln pts dtherm tpr shadow vmml flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2_erns invpcid xsaveopt",  
    "pageSize" : NumberLong(4096),  
    "numPages" : 1965020,  
    "maxOpenFiles" : 1024  
  },  
  "ok" : 1  
}  
> db.serverCmdLineOpts()  
{  
  "argv" : [  
    "mongod",  
    "--dbpath",  
    "/home/OBYON/data/db",  
    "--logpath",  
    "/home/OBYON/data/log/mongod.log"  
  ],  
  "parsed" : {  
    "storage" : {  
      "dbPath" : "/home/OBYON/data/db"  
    },  
    "systemLog" : {  
      "destination" : "file",  
      "path" : "/home/OBYON/data/log/mongod.log"  
    },  
    "ok" : 1  
  }  
}
```

\$ cat mongod.log

```
OBYON@localhost:~/data/log  
Fichier Édition Affichage Rechercher Terminal Onglets Aide  
OBYON@localhost:~/data/log  
[OBYON@localhost ~]$ cd data  
[OBYON@localhost data]$ cd log  
[OBYON@localhost log]$ cat mongod.log  
2016-05-30T13:27:34.760+0200 I CONTROL [initandlisten] MongoDB starting : pid=28884 port=27017 dbpath=/home/OBYON/data/db 64-bit host=localhost.local  
domain  
2016-05-30T13:27:34.760+0200 I CONTROL [initandlisten] db version v3.0.11  
2016-05-30T13:27:34.760+0200 I CONTROL [initandlisten] git version: 48f8b49dc30cc2485c6c1f3db31b723258fcbf39  
2016-05-30T13:27:34.760+0200 I CONTROL [initandlisten] build info: Linux buildvml-14-nfs.phx2.fedoraproject.org 4.3.5-300.fc23.x86_64 #1 SMP Mon Feb 1  
03:18:41 UTC 2016 x86_64 BOOST_LIB_VERSION=1_58  
2016-05-30T13:27:34.760+0200 I CONTROL [initandlisten] allocator: tcmalloc  
2016-05-30T13:27:34.760+0200 I CONTROL [initandlisten] options: { storage: { dbPath: "/home/OBYON/data/db" }, systemLog: { destination: "file", path:  
"/home/OBYON/data/log/mongod.log" } }  
2016-05-30T13:27:34.797+0200 I JOURNAL [initandlisten] journal dir=/home/OBYON/data/db/journal  
2016-05-30T13:27:34.797+0200 I JOURNAL [initandlisten] recover : no journal files present, no recovery needed  
2016-05-30T13:27:34.978+0200 I JOURNAL [durability] Durability thread started  
2016-05-30T13:27:34.978+0200 I JOURNAL [journal writer] Journal writer thread started  
2016-05-30T13:27:34.983+0200 I CONTROL [initandlisten]  
2016-05-30T13:27:34.983+0200 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.  
2016-05-30T13:27:34.983+0200 I CONTROL [initandlisten] ** We suggest setting it to 'never'  
2016-05-30T13:27:34.983+0200 I CONTROL [initandlisten]  
2016-05-30T13:27:34.985+0200 I NETWORK [initandlisten] waiting for connections on port 27017  
2016-05-30T13:27:41.471+0200 I NETWORK [initandlisten] connection accepted from 127.0.0.1:49154 #1 (1 connection now open)  
[OBYON@localhost log]$
```

\$ mongorestore -d music /home/OBYON/data/mymusic

```
OBYON@localhost:~/data/mymusic  
Fichier Édition Affichage Rechercher Terminal Onglets Aide  
OBYON@localhost:~/data/log  
[OBYON@localhost mymusic]$ mongorestore -d music --file /home/OBYON/data/mymusic  
2016-05-30T13:59:30.637+0200 error parsing command line options: unknown option "file"  
2016-05-30T13:59:30.637+0200 try 'mongorestore --help' for more information  
[OBYON@localhost mymusic]$ mongorestore -d music /home/OBYON/data/mymusic  
2016-05-30T13:59:40.397+0200 building a list of collections to restore from /home/OBYON/data/mymusic dir  
2016-05-30T13:59:40.398+0200 reading metadata file from /home/OBYON/data/mymusic/songs.metadata.json  
2016-05-30T13:59:40.398+0200 restoring music.songs from file /home/OBYON/data/mymusic/songs.bson  
2016-05-30T13:59:40.401+0200 restoring indexes for collection music.songs from metadata  
2016-05-30T13:59:40.402+0200 finished restoring music.songs  
2016-05-30T13:59:40.402+0200 done  
[OBYON@localhost mymusic]$
```

PARTIE N°2

1.

```
$ use music
$ db.songs.find(
  { "_id" : ObjectId("55328bd3f238ef5f0de2ad33"), "title" : "Papaoutai", "artist" :
    "Stromae", "album" : "Racine carrée", "year" : 2013 }
  { "_id" : ObjectId("55328c04f238ef5f0de2ad34"), "title" : "Alors on danse", "artist" :
    "Stromae", "album" : "Cheese", "year" : 2010 }
  { "_id" : ObjectId("55328c56f238ef5f0de2ad35"), "title" : "Formidable", "artist" :
    "Stromae", "album" : "Racine carrée", "year" : 2013 }
  ..... etc
```
2.

```
$ db.songs.count( )
20
```
3.

```
$ db.songs.find({artist: 'Coldplay', album: 'X&Y'})
{ "_id" : ObjectId("55328f9cf238ef5f0de2ad41"), "title" : "Fix You", "artist" : "Coldplay",
  "album" : "X&Y", "year" : 2005, "bpm" : 111.6 }
{ "_id" : ObjectId("55328facf238ef5f0de2ad42"), "title" : "Speed of Sound", "artist" :
  "Coldplay", "album" : "X&Y", "year" : 2005, "bpm" : 123.1 }
```
4.

```
$ db.songs.find({"artist": "Stromae"}, {"title":1, "album":1}).sort({"year": -1, "title":
1})
{ "_id" : ObjectId("55328c56f238ef5f0de2ad35"), "title" : "Formidable", "album" : "Racine
carrée" }
{ "_id" : ObjectId("55328bd3f238ef5f0de2ad33"), "title" : "Papaoutai", "album" : "Racine
carrée" }
{ "_id" : ObjectId("55328cb0f238ef5f0de2ad36"), "title" : "Tous les memes", "album" :
  "Racine carrée" }
{ "_id" : ObjectId("55328c04f238ef5f0de2ad34"), "title" : "Alors on danse", "album" :
  "Cheese" }
```
5.

```
$ db.songs.find({"artist": "Coldplay"}).map(function(song){return song.title + "(" +
song.album + ")"})
```

```
[
  "Paradise(Mylo Xyloto)",
  "The Scientist(A Rush of Blood to the Head)",
  "Clocks(A Rush of Blood to the Head)",
  "Fix You(X&Y)",
  "Speed of Sound(X&Y)"
]
```
6.

```
$ db.songs.distinct("artist", {$and: [{"year": {$gte: 2002}}, {"year": {$lte: 2005}}]})
[ "Maroon 5", "Coldplay" ]
```

```

7.    $ db.createCollection("recordLabel",{capped: true, size: 1000, max: 3,
validator:{$and: [{"nom": {$type: "string"}},{url": {$regex:
/^((https?:\V\)?(\da-z\.-)+)\.([a-z\.\]{2,6})([\Vw \.-]*)*V?$/$)}}}}
{ "ok" : 1 }

8.    $ db.recordLabel.insert([{'nom': 'toto', 'url': 'https://toto.com'},{'nom': 'titi', 'url':
'https://titi.com'},{'nom': 'tata', 'url': 'https://tata.com'}])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})

    $ db.recordLabel.insert({'nom': 'tutu', 'url': 'https://tutu'})
WriteResult({ "nInserted" : 1 })
> db.recordLabel.find()
{ "_id" : ObjectId("574d5afe674daa76176359c3"), "nom" : "titi", "url" : "https://titi.com" }
{ "_id" : ObjectId("574d5afe674daa76176359c4"), "nom" : "tata", "url" : "https://tata.com" }
}
{ "_id" : ObjectId("574d5b429415e77b6c8f3c00"), "nom" : "tutu", "url" : "https://tutu" }

```

On constate que le premier document à été écrasé lors de l'insertion du 4e, pour garder comme demandé uniquement 3 documents en mémoire.

```

9.    $ db.runCommand({"collMod": "recordLabel","validator": {$and: [{"nom": {$type:
"string"}},{url": {$regex: /^((https?:\V\)?(\da-z\.-)+)\.([a-z\.\]{2,6})([\Vw \.-]*)*V?$/$}},{pays":
{$regex:
/^((AF|AX|AL|DZ|AS|AD|AO|AI|AQ|AG|AR|AM|AW|AU|AT|AZ|BS|BH|BD|BB|BY|BE|BZ|BJ|BM|
BT|BO|BQ|BA|BW|BV|BR|IO|BN|BG|BF|BI|KH|CM|CA|CV|KY|CF|TD|CL|CN|CX|CC|CO|KM|
CG|CD|CK|CR|CI|HR|CU|CW|CY|CZ|DK|DJ|DM|DO|EC|EG|SV|GQ|ER|EE|ET|FK|FO|FJ|FI|FR
|GF|PF|TF|GA|GM|GE|DE|GH|GI|GR|GL|GD|GP|GU|GT|GG|GN|GW|GY|HT|HM|VA|HN|HK|H
U|IS|IN|ID|IR|IQ|IE|IM|IL|IT|JM|JP|JE|JO|KZ|KE|KI|KP|KR|KW|KG|LA|LV|LB|LS|LR|LY|LI|LT|L
U|MO|MK|MG|MW|MY|MV|ML|MT|MH|MQ|MR|MU|YT|MX|FM|MD|MC|MN|ME|MS|MA|MZ|
MM|NA|NR|NP|NL|NC|NZ|NI|NE|NG|NU|NF|MP|NO|OM|PK|PW|PS|PA|PG|PY|PE|PH|PN|PL|
PT|PR|QA|RE|RO|RU|RW|BL|SH|KN|LC|MF|PM|VC|WS|SM|ST|SA|SN|RS|SC|SL|SG|SX|SK|
SI|SB|SO|ZA|GS|SS|ES|LK|SD|SR|SJ|SZ|SE|CH|SY|TW|TJ|TZ|TH|TL|TG|TK|TO|TT|TN|TR|T
M|TC|TV|UG|UA|AE|GB|US|UM|UY|UZ|VU|VE|VN|VG|VI|WF|EH|YE|ZM|ZW)$/$}}}}})
{ "ok" : 1 }

```

10. TTL = Time To Live, C'est une option disponible sur les éléments d'une collection pour indiquer un temps en seconde au bout duquel l'élément sera définitivement effacé par le processus Mongo. On peut également indiquer un horaire précis pour la suppression.

Pour utiliser un TTL il faut avoir un champ de type date, ou de type tableau de date, pour utiliser `db.collection.createIndex()` avec l'option `expireAfterSeconds` sur ce champ.

Concrètement, sur notre collection `recordLabel`, il faudrait ajouter un champ date de création, et utiliser `db.recordLabel.createIndex` avec `expireAfterSeconds` sur ce champ.

```
> db.recordLabel.createIndex({ "createdAt": 1 }, { expireAfterSeconds: 10 })
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

PARTIE N°3

1. `createUsers.js`
2. `addFavorite.js`
3. `addNotes.js`
4. `getNoFavUsers.js` et `getColdplayFans.js`
5. `getTopTen.js`

PARTIE N°4

Pour cette partie je considère que chaque personne dont on parle dispose d'un CV.
Le cas échéant l'interprétation pourrait être différente..

En utilisant un Embedded Design, on peut imaginer deux représentation, l'une basée sur les personnes, l'autre sur les CV. Deux possibilités:

Centrer sur les personnes:

```
{  
  nom: "florile"  
  prenom: "maxime"  
  ...  
  cv: {  
    intitule: "cv developpeur web"  
    ...  
  }  
}
```

Exemple: Sur une site de recrutement classique, lorsque l'on accède au profil d'un utilisateur, et que son CV nous est directement présenté dans le profil. En d'autre terme si le CV est toujours affiché avec le profil de la personne par exemple..

Centrer sur les CV:

```
{  
  intitule: "cv developpeur web"  
  ...  
  personne: {  
    nom: "florile"  
    prenom: "maxime"  
  }  
}
```

Exemple: J' imagine un site de recrutement un peu différent, dans lequel les entreprises filtrent les CV par compétences, pour ne choisir que ceux qui leur correspondent vraiment, pour ensuite accéder au profil de la personne via le CV. On a donc un système plutôt centré sur le CV, mais qui amène tout de même toujours à consulter le profil de la personne associée.

D'autre part, pour les deux ont l'avantage de permettre la modification commune du CV et de la personne sans risquer de perte de données.

Avec un separatedDesign, on aurait deux tables, une pour les personnes, une pour les CV.

Deux possibilités:

Table personnes

```
{
    id: 1
    nom: "florile"
    ...
    id_cv: 1
}
```

Table CV

```
{
    id: 1
    intitule: "developpeur web"
    ...
}
```

Exemple: Une plateforme qui stockerait beaucoup d'informations sur les personnes et dans leur CV. Cependant on y afficherait souvent un CV sans personne associée. Une banque de CV consultable, à coté d'un système plus classique de profil + CV associé.

Table CV

```
{
    id: 1
    intitule: "developpeur web"
    ...
    id_personne: 1
}
```

Table personnes

```
{
    id: 1
    nom: "florile"
    ...
}
```

Exemple: Pour cet exemple, j'imagine toujours beaucoup d'informations stockées puisque le separated collection design s'y prête mieux. Cependant ici on affichera souvent des personnes sans leur CV, donc peut être un site moins axé sur le recrutement par exemple, mais avec tout de même une possibilité de consulté un CV

PARTIE N°5

```
1. mongodump --db music --collection songs --out /home/OBYON/data/export/
2016-06-07T11:30:11.782+0200    writing music.songs to
2016-06-07T11:30:11.783+0200    done dumping music.songs (20 documents)

2. mongodump --db music --collection users --query '{favoriteSongs: {$size: 0}}' --out
/home/OBYON/data/export/
2016-06-07T11:31:12.431+0200    writing music.users to
2016-06-07T11:31:12.432+0200    done dumping music.users (94 documents)

3. mongorestore --db noo-favorites --collection users
/home/OBYON/data/export/music/users.bson
2016-06-07T11:43:58.373+0200    checking for collection data in
/home/OBYON/data/export/music/users.bson
2016-06-07T11:43:58.374+0200    reading metadata for noo-favorites.users from
/home/OBYON/data/export/music/users.metadata.json
2016-06-07T11:43:58.427+0200    restoring noo-favorites.users from
/home/OBYON/data/export/music/users.bson
2016-06-07T11:43:58.488+0200    restoring indexes for collection noo-favorites.users
from metadata
2016-06-07T11:43:58.489+0200    finished restoring noo-favorites.users (94 documents)
2016-06-07T11:43:58.489+0200    done
```

4. Il existe:

Export: mongodump, mongoexport,
Import: mongorestore, mongoimport

Les différences:

mongodump sauvegarde l'intégralité des documents, y compris les informations de typage, sous un format unique Bson. C'est donc la commande à privilégier pour des sauvegardes de qualité en vue d'un backup/restore par exemple.

Cette commande va de paire avec mongorestore, qui prend en entrée uniquement le format Bson, et importe donc tout les documents dans leur intégralité, y compris le typage.

mongoexport sauvegarde les documents en Json ou CSV, mais omet les informations de typage, utile si l'on souhaite ré-importer les documents dans une base.

C'est donc une commande à éviter pour faire un backup, qui sert plutôt pour simplement exporter une liste de documents, ou les exporter vers un autre système par exemple.

mongoimport prend en entrée du Json, CSV et même TSV. Il est donc possible d'importer des données issues d'un autre générateur que mongoexport, même si ces fichiers ne mentionnent pas les informations de typage et sont donc souvent à éviter sur un environnement de production.

