- **Accéder à une base de données**
  - Pour accéder à une base de données nous allons installer mongoDB sur notre poste :
  - A adapter en fonction de votre système d'exploitation.

```
$ sudo apt install mongodb
```

  - Dans le répertoire de votre application vous devez ajouter mongodb aux dépendances et dans les **node_modules** :

```
$ npm install mongodb --save
```

  - Nous allons faire des tests dans un module pour valider la connexion avec la base de données '**testmongodb.js**' :

- **Accéder à une base de données :**

```
// a écrire dans un fichier testmongodb.js à la racine de votre projet myExpressHdbApp
var dbClient = require('mongodb').MongoClient;
console.log('--> mongoClient : ', dbClient);
var assert = require('assert');
var url = 'mongodb://localhost:27017/gretajs'; // Connection URL
// Use connect method to connect to the server
dbClient.connect(url, { useNewUrlParser: true },
  function(err, client) {
  assert.equal(null, err);
  console.log("Successfully connected to server");
  global.db = client.db('gretajs');
  console.log('global.db : ' , global.db);
  client.close();
});
```

Il faut que vous lanciez la base de données mongoDB :

```
$ sudo mongod
(…) [initandlisten] waiting for connections on port 27017
```

Nous allons pouvoir tester cette connexion dans un terminal :

```
$ node ./testmongodb.js
Connected successfully to server
```

- **Sécuriser la connexion à mongoDB**
  - Pour cela il faut créer un user admin sur la base via le shell:

```
$ mongo
> use admin
Switched to db admin
> db.createUser(
...   {
...     user: "myUserAdmin",
...     pwd: "abc123",
...     roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
...   }
... );
Successfully added user: {
    "user" : "myUserAdmin",
    "roles" : [
        {
            "role" : "userAdminAnyDatabase",
            "db" : "admin"
        },
        "readWriteAnyDatabase"
    ]
}
```

  - Ensuite il faut lancer la base avec l'option "--auth"

```
$ sudo mongod --auth
```

- **Sécuriser la connexion à mongoDB**
  - Maintenant il faut se connecter avec le login et le password défini précédemment :

```
$ mongo -u "myUserAdmin" -p "abc123" --authenticationDatabase "admin"
MongoDB shell version: 3.2.20
connecting to: test
> use gretajs
switched to db gretajs
> db.createUser (
  {
    user:"greta",
    pwd: "azerty",
    roles: [
      {role: "readWrite", db: "gretajs"},
      {role: "read", db: "reporting"}
    ]
  }
)
Successfully added user: {
    "user" : "greta",
    (…)
}
> exit
```

- **Sécuriser la connexion à mongoDB**
  - Maintenant il faut se connecter avec le login et le password défini précédemment :

```
$ mongo -u "myUserAdmin" -p "abc123" --authenticationDatabase "admin"
MongoDB shell version: 3.2.20
connecting to: test
> use gretajs
switched to db gretajs
> db.createUser (
  {
    user:"greta",
    pwd: "azerty",
    roles: [
      {role: "readWrite", db: "gretajs"},
      {role: "read", db: "reporting"}
    ]
  }
)
Successfully added user: {
    "user" : "greta",
    (…)
}
> exit
```

- **Sécuriser la connexion à mongoDB :**

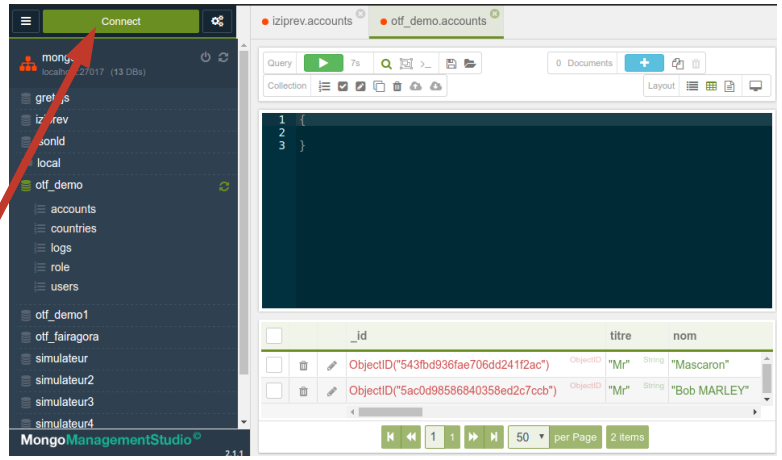
```
var dbClient = require('mongodb').MongoClient;
console.log('--> mongoClient : ', dbClient);
var f = require('util').format;
var assert = require('assert');
var user = encodeURIComponent('greta');
var password = encodeURIComponent('azerty');
const authMechanism = 'DEFAULT';
// formatage de l'URL de connexion
var url = f('mongodb://%s:%s@localhost:27017/gretajs?authMechanism=%s',
            user, password, authMechanism); // Connexion URL
// Connexion effective à la base de donées
dbClient.connect(url, { useNewUrlParser: true },
  function(err, client) {
    assert.equal(null, err);
    console.log("Successfully connected to server");
    global.db = client.db('gretajs');
    console.log('global.db : ' , global.db);
    client.close();
  }
);
```

Nous allons pouvoir tester cette connexion dans un terminal :

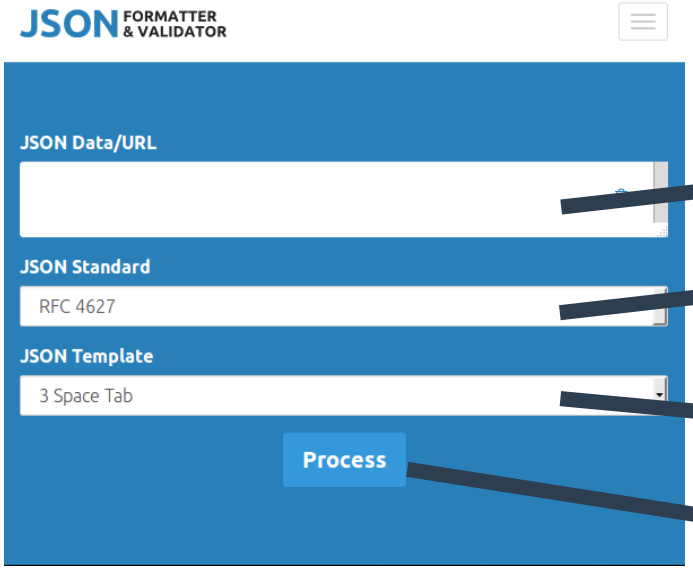
```
$ node ./testmongodbauth.js
Connected successfully to server
(…)
```

Node, Express, MongoDB

- **Outils pour le développement avec mongoDB :**
  - Mongo Management Studio cliquez sur ce bouton pour la configuration de MMS (Création collection exos)



- **JSON Formatter-Validator**



Coller ici le texte en JSON

Format RFC (laissez par défaut)

Indentation du code JSON

Cliquez sur “Process”

Node, Express, MongoDB

- Création de la collection '**exercices**' via le shell :

```
> db.createCollection("exercices")
{ "ok" : 1 }
> show collections
exercices
> db.exercices.insert({libelle: "Exos1"})
WriteResult({ "nInserted" : 1 })
> db.exercices.insert({libelle: "Exos2"})
WriteResult({ "nInserted" : 1 })
> db.exercices.insert({libelle: "Exos3"})
WriteResult({ "nInserted" : 1 })
> db.exercices.insert({libelle: "Exos4"})
WriteResult({ "nInserted" : 1 })
> db.exercices.insert({libelle: "Exos5"})
```

- Voyons les enregistrements qui ont été insérés :

```
> db.exercices.find()
{ "_id" : ObjectId("5babb4981be41ac869f07764"), "libelle" : "Exos1" }
{ "_id" : ObjectId("5babb4af1be41ac869f07765"), "libelle" : "Exos2" }
{ "_id" : ObjectId("5babb4b31be41ac869f07766"), "libelle" : "Exos3" }
{ "_id" : ObjectId("5babb4b51be41ac869f07767"), "libelle" : "Exos4" }
{ "_id" : ObjectId("5babb4bc1be41ac869f07768"), "libelle" : "Exos5" }
>
```

Node, Express, MongoDB

- **Ajoutons une connexion base de données pour une action ‘/exos’ qui listera les enreg. d’exercices :**
  - Créons la connexion dans ‘**app.js**’ :

```
(…)
global.db={};
var mongoClient = require('mongodb').MongoClient;

// Connexion URL
//var url = 'mongodb://simplon:azerty@127.0.0.1:27017/simplonjs?authMechanism=DEFAULT';
var url = ‘mongodb://127.0.0.1:27017/simplonjs’;
// Utilisation de la methode “connect” pour se connecter au serveur
mongoClient.connect(url, function(err, client) {
  global.db = client.db('simplonjs');  //On met en global la connexion à la base
  console.log("Connected successfully to server: global.db initialized");
});
(…)
module.exports = app;
```

  - Pensez à ajouter le require et le use

```
var usersRouter = require('./routes/users');
var exosRouter = require('./routes/exos');
(…)
app.use('/exos', exosRouter);
```

Node, Express, MongoDB

- Creation de la route '**exos.js**' :

```
var express = require('express');
var router = express.Router();
/* GET home page. */
router.get('/', function(req, res, next) {
  global.db.collection('exercices').find().toArray(function(err, result) {
    if (err) {
      throw err;
    }
    console.log(result);
    res.render('exos', {stitle: 'First Cnx Mongo',
                        title: 'Liste déroulante',
                        exos: result});
  });
});
module.exports = router;
```

- Voyons le code dans la vue '**exos.hbs**' pour afficher une liste déroulante

```
<h1>{{ stitle}}</h1>
<h2>Express App : {{title}}</h2><br>
   <select name="exos">
    {{#each exos}}
        <option value="{{this._id}}">{{this.libelle}}</option>
    {{/each}}
</select>
```

Node, Express, MongoDB

- **Résultat de l’appel de l’action ‘exos’ :**

Accueil Toto (error)

First Cnx Mongo

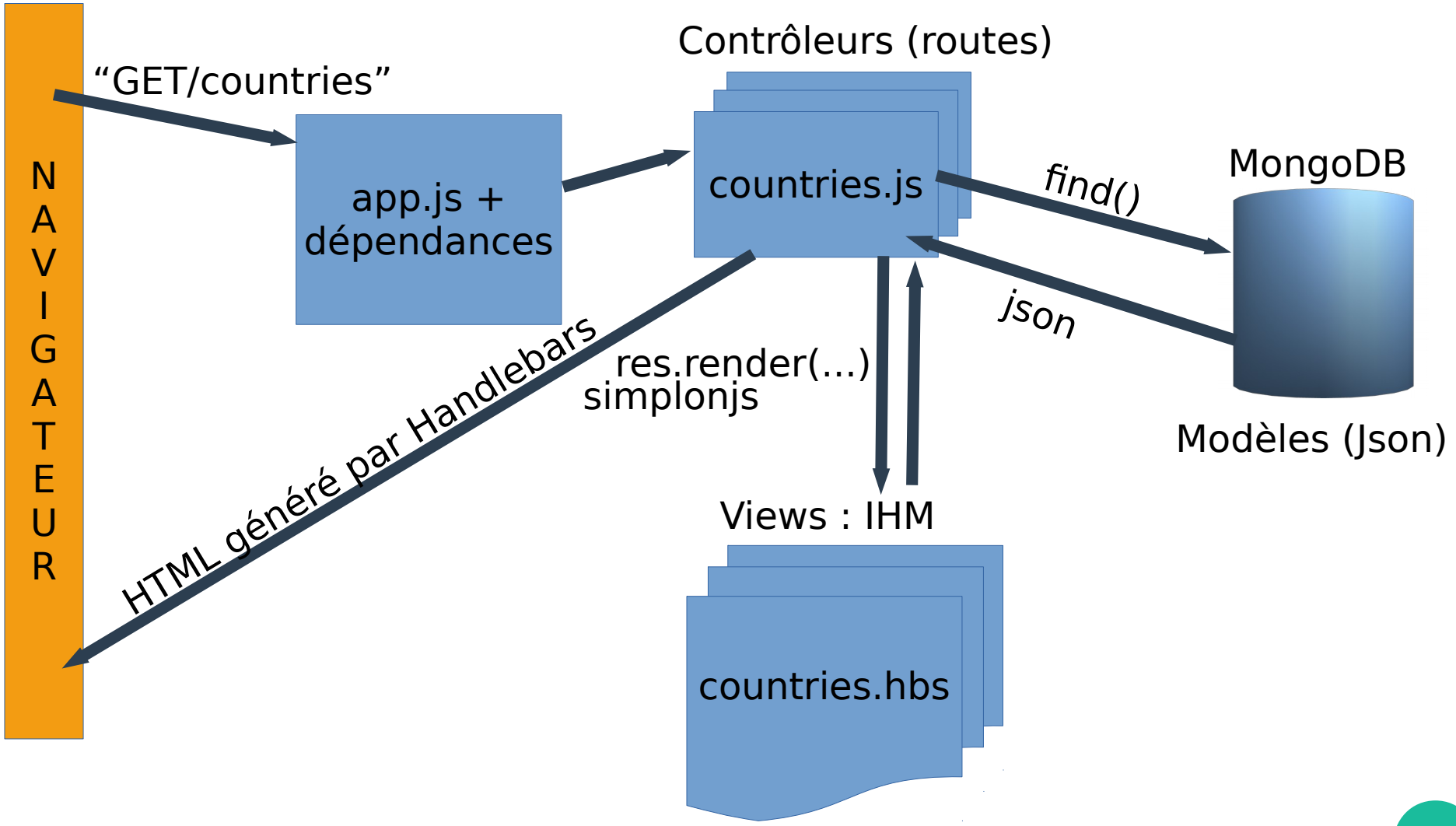
Express App : Liste déroulante

Exos1
Exos1
Exos2
Exos3
Exos4
Exos5

NB : Attention en copiant depuis le PDF il se peut que des caractères invisibles déclenchent des erreurs silencieuses !!!

Node, Express, MongoDB

- **Schéma architecture Application Express/Handlebars/MongoDB :**



- **Exercice :**
  - En utilisant le code vu précédemment, vous ajouterez une action nommée '**/countries**' un controleur '**countries.js**' et une vue '**countries.hbs**' qui affiche la liste des pays dans un select (liste déourlante).
  - Le fichier countries est téléhargeable ici

```
[{
      "_id":"572377f1e3d6f2d245000001",
      "name":"Afghanistan",
      "code":"AF",
   },
(…)
]
```

Pour importer en ligne de commande un fichier json dans une collection (une seule ligne) : "mongoimport"

```
$ mongoimport -c countries -d gretajs -u greta -p azerty --jsonArray
--file ../countries.json
```