Mongoose y Async

Este ejemplo muestra como podemos usar `mongoose`y `async`, dos módulos de NodeJS que podemos instalar mediante `npm`
y que nos ayudarán a programar nuestro servidor con un código más limpio, seguro y legible.

Mongoose

Mongoose es un ODM (Object Document Model) para Node.JS y MongoDB. Como ya sabemos, los documentos de MongoDB
no tienen esquema. Mongoose nos permite **definir esquemas** para nuestras colecciones de MongoDB,
obtener documentos de la base de datos directamente en un objeto de JavaScript y validar documentos respecto al esquema que hemos definido para ellos.

Para instalar mongoose en nuestra app, y como ya hemos visto en el ejemplo 02 de módulos, debemos hacer lo siguiente:

1. **Añadirlo al fichero `package.json` como dependencia**. Lo podemos hacer con el comando npm install --save:

   ```
   [sudo] npm install --save mongoose
   ```

2. **Requerir el módulo** desde nuestro fichero javascript principal:

   ```
   var mongoose = require('mongoose');
   ```

3. Ahora podemos **conectarnos a nuestra base de datos de mongodb** con el siguiente método:

   ```
   mongoose.connect(uri, [opciones], [callback])
   ```

   donde `uri` es un string que especifica el nombre de la base de datos, el host, el puerto, el usuario y la password, con el siguiente formato:

   ```
   mongodb://username:password@host:port/database
   ```

   cuando la bd es local, y así es como trabajaremos en este curso,
   podemos utilizar la siguiente uri, sin username ni puerto ni contraseña,
   ya que no usamos autenticación para acceder a la bd y el puerto se pone
   por defecto a 27017, que es por el que escucha el servidor de mongodb:

   ```
   mongoose.connect("mongodb://localhost/nombre-bd")
   ```

4. Para **definir nuevos modelos y esquemas** para las colecciones de nuestra base de datos, para tener el código más organizado
   los definiremos en un módulo aparte, situado en una carpeta llamada `models` o `modelos`, donde habrá un fichero js por cada
   modelo que definamos. De este modo, en la carpeta `models` habrá un fichero principal `index.js` que será el que se incluirá al hacer
   el require de la carpeta `models` y que tendrá una función para incluir los ficheros js de los módulos que hayamos definido, como podéis ver en el ejemplo.

Aquí tenéis una guía con más información de como usar Mongoose:
http://mongoosejs.com/docs/guide.html

Async

Async es un módulo de que contiene herramientas para poder tratar mejor con la asincronía de NodeJS.
Lo usaremos principalmente para evitar tener que escribir callbacks anidadas, que son muy comunes
en NodeJS dada su naturaleza asíncrona, y que nos permitirán reemplazar el siguiente código:

```
obj.hacerAlgo(function(err1, resultado1) {
    .
    .
    .
    obj2.hacerAlgo2(function(err2, resultado2) {
        .
        .
        .
        obj3.hacerAlgo3(function(err3, resultado3) {
            .
            .
            .
            obj4.hacerAlgo4(function(err4, resultado4) {
                .
                .
                .
            });
            .
            .
        });
        .
        .
    });
    .
    .
});
```

Con este, más limpio:

```
async.waterfall([
    function(callback) {
        obj.hacerAlgo(callback);
    },
    function(resultado, callback) {
        obj2.hacerAlgo(callback);
    },
    function(resultado, callback) {
        obj3.hacerAlgo(callback);
    },
    function(resultado, callback) {
        obj4.hacerAlgo(callback);
    }
],
    function(error, resultado) {
        if (error) // Un error ha ocurrido en alguna de las funciones
        else // Resultado contiene el resultado de la última callback de
obj4.hacerAlgo
    }
);
```

Podemos incluir el módulo async del mismo modo que lo hemos hecho con mongoose.

Para más información de como usar async, podéis visitar: https://github.com/caolan/async