

Entrega 6: mod6_quiz_BBDD

Versión: Versión: 12 de Febrero de 2019

Objetivo

Entender y aprender la utilización de bases de datos para almacenar información de forma persistente en aplicaciones y comandos.

Resumen de la práctica

En esta entrega se debe desarrollar un programa **quiz_bbdd** con la misma interfaz y comportamiento que el de la entrega 3: **quiz_cmd**. Este debe almacenar las preguntas y respuestas (quizzes) en una base de datos **SQLite3**, en vez de en un fichero.

La mayor parte del código desarrollado para la entrega anterior (**quiz_cmd**) es idéntico al de esta entrega, por lo que se recomienda reutilizar el código entregado en la practica anterior y empezar a trabajar a partir de él.

El proyecto https://github.com/practicass-ging/mooc_node-mod6_quiz_bbdd contiene los mismos ficheros **cmds.js**, **main.js**, **model.js** y **out.js** de **mod5_quiz_cmd**. Solo se han actualizado las baterías de test. Para realizar esta practica debe actualizar el fichero **cmds.js** con el desarrollo entregado en **mod5_quiz_cmd**, para modificar a continuación los ficheros **cmds.js**, **main.js**, **model.js** y **out.js** para que los quizzes se identifiquen por su índice en la BBDD y además se guarden en esta. Una vez completado se deberá entregar a través de la plataforma.

El siguiente screencast de video <https://www.youtube.com/watch?v=qqhSoUpzoQs> describe los detalles de como realizar esta practica.

¡Cuidado! El código dado funciona en UNIX/Linux. Para ejecutar en Windows habrá que aplicar los cambios que se indican en las instrucciones sobre compatibilidad UNIX-Windows para node.js.

Comandos del Programa Quiz

El comportamiento de esta práctica debe ser idéntico al de la práctica **mod5_quiz_cmd**, excepto en la salida mostrada por el comando **list**. En esta práctica, los quizzes se identifican por el valor de su clave primaria (**id**) en la base de datos, por lo que los índices mostrados por el comando **list** pueden ser no consecutivos.

Persistencia

En la práctica **mod5_quiz_cmd** se implementó el modelo de datos de los quizzes como un array de objetos que se guardaban en un fichero llamado **quizzes.json**.

En esta práctica se pide sustituir este modelo por otro en el que los datos se guarden en una base de datos **SQLite3** albergada en un fichero de nombre **quizzes.sqlite**. A esta base de datos debe accederse usando el ORM **Sequelize**.

En la BBDD debe crearse una tabla llamada **quizzes** para almacenar los quizzes creados. Esta tabla debe tener los siguientes campos:

- **question** - String con el texto de la pregunta.
- **answer** - String con el texto de la respuesta.
- **id** - Número usado como clave primaria.
- **createdAt** - Fecha de creación del quiz.
- **updatedAt** - Fecha de última modificación del quiz.

La tabla debe crearse usando el método **sync** de Sequelize, sin usar migraciones.

Los dos primeros campos, **question** y **answer**, deben definirse explícitamente en el modelo. Pero los tres últimos, **id**, **createdAt** y **updatedAt**, los creará automáticamente el método sync.

Desde el método **sync** de Sequelize debe comprobarse si la tabla quizzes está vacía, y en caso de ser así, debe rellenarse con las mismas preguntas de la práctica **mod5_quiz_cmd**.

Realización y prueba de la práctica

Para comprobar que la práctica ha sido realizada correctamente hay que utilizar el validador de este repositorio

https://github.com/practicas-ging/mooc_node-mod6_quiz_bbdd

Recuerde que para utilizar el validador se debe tener node.js (y npm) (<https://nodejs.org/es/>) y Git instalados. El proyecto se descarga, instala y ejecuta en el ordenador local con estos comandos:

```
$                                ## El proyecto debe clonarse en el ordenador local
$ git clone https://github.com/practicas-ging/mooc_node-mod6_quiz_bbdd
$
$ cd mooc_node-mod6_quiz_bbdd    ## Entrar en el directorio de trabajo
$
$ npm install                    ## Instala el programa de test
$
$                                ## -> Incluir la solución en el esqueleto clonado
$
$ npm run checks                 ## Pasa los tests a los ficheros solicitados
.....                          ## en el directorio de trabajo
.....
... (resultado de los tests)
$
```

Una vez que el proyecto está en el ordenador local, se deben instalar los paquetes **sqlite3** y **sequelize** invocando en el directorio del proyecto los comandos npm:

```
$ npm install --save sequelize
```

```
$ npm install -g sqlite3
```

sqlite3 debe instalarse con la opción -g (instalación global), para que el entorno de test funcione bien también en Windows. Una vez instalados estos dos paquetes se recomienda comenzar el desarrollo del código.

Para realizar esta practica se debe actualizar el fichero **cmds.js** con el desarrollo entregado en **mod5_quiz_cmd**. A continuación se debe modificar con el editor los ficheros **cmds.js**, **main.js**, **model.js** y **out.js** con las modificaciones indicadas en este enunciado para que los quizzes se gestionen con una BBDD. Los tests determinarán si los nuevos desarrollos son correctos o no

respecto a este enunciado. Es conveniente guardar versiones en la rama principal, cada vez que se consolide una nueva funcionalidad.

¡Cuidado! El código dado funciona en UNIX/Linux. Para ejecutar en Windows habrá que aplicar los cambios que se indican en las instrucciones sobre compatibilidad UNIX-Windows para node.js.

Los tests pueden pasarse las veces que sea necesario. Si se pasan nada más descargar el proyecto, indicarán que no se ha realizado todavía nada de esta practica. También pueden utilizarse para probar el programa de otro compañero sustituyendo los ficheros que se desee probar. El programa de test incluye además un comando para generar el fichero ZIP

```
$  
$ npm run zip          ## Comprime los ficheros del directorio en un fichero .zip  
$
```

Este genera el fichero **mooc_node-mod6_quiz_bbdd.zip** con el directorio de la practica comprimido. Este fichero ZIP debe subirse a la plataforma para su evaluación.

Instrucciones para la Entrega y Evaluación.

La entrega debe realizarse subiendo a la plataforma un fichero ZIP con el contenido comprimido del directorio del proyecto **mooc_node-mod6_quiz_bbdd**, generado con el comando “npm run zip”. Este contiene los ficheros **cmds.js**, **main.js**, **model.js** y **out.js**, **package.json**.

El evaluador debe descargar el fichero entregado y comprobar que funciona correctamente. El fichero descargado es un paquete npm, que puede instalarse con todas sus dependencias con npm. Una vez instalado puede ejecutarse o pueden pasarse los test.

RUBRICA. Se puntuará el ejercicio a corregir sumando el % indicado a la nota total si la parte indicada es correcta:

- 40%: si se implementa correctamente el cuerpo de la función **test()**, **play** y **credits**
- 40%: si lee correctamente el fichero **mod4-quotes_of_the_day.json**
- 20%: si muestra por consola correctamente una cita aleatoria del fichero

Si pasa todos los tests se deberá dar la máxima puntuación.

El objetivo de este curso es sacar el máximo provecho al trabajo dedicado y para ello lo mejor es utilizar las evaluaciones para ayudar al evaluado, especialmente a los principiantes. Al evaluar se debe dar comentarios sobre la corrección del código, su claridad, legibilidad, estructuración y documentación, siempre que puedan ayudar al evaluado.

¡Cuidado! Una vez enviadas, tanto la entrega, como la evaluación, no se pueden cambiar. Esperar a tener completa y revisada, tanto la entrega, como la evaluación antes de enviarlas.