


Proyecto MERN 6

Enunciado de proyecto

Ahora que has creado tu primer servidor en Express, vamos a pedirte como siguiente proyecto que vuelvas a crear otro servidor, pero esta vez debes tener al menos dos modelos de datos relacionados entre si .

Para preparar este proyecto, sigue los siguientes pasos:

- Crea una carpeta para el proyecto.
- Navega a la carpeta y lanza `npm init -y`.
- Instala las dos dependencias principales con `npm i express mongoose`.
- Asegúrate de tener Mongo preparado ya sea instalado o con Docker.
- Prepara tu server para escuchar en el puerto 4001 tal y como hemos explicado en clase y has visto en los videos del contenido.
- Crea un archivo `.gitignore` en la raíz de tu proyecto y añade dentro el texto `node_modules` para no subir la carpeta de módulos instalados a Github.
- Lanza ahora `git init` e inicializa tu repositorio. Crea un repositorio en Github y haz commit/push del contenido.

En este proyecto tendrás que crear un servidor de **Express** que trabaje con **MongoDB** a través de mongoose. Conectarás tu proyecto cuando el servidor se inicialice.

Al igual que en el proyecto anterior, crearás un modelo de datos a tu gusto, pero esta vez debes buscar algo que tenga otro modelo de datos relacionado, por ejemplo:

- Un atleta olímpico es un documento
- Un juego olímpico es un documento

- Ambos se pueden relacionar por medio de las participaciones del atleta en un juego correspondiente.

En este caso, crearíamos dos modelos `Athlete` y `OlympicGame` que relacionaríamos con una relación 1:N desde el documento **athlete** al documento **olympicGame**.



En tu caso, tendrás que buscar un ejemplo adecuado en el que deberás relacionar al menos dos documentos distintos entre si, de forma que el elemento A tenga un array de elementos B, y el elemento B tenga un solo elemento A como referencia. ¡Tu eliges la temática, añade como mínimo cuatro campos a cada documento!

Ten en cuenta que comprobaremos la relación entre modelos a raíz de lo que enumeramos en los criterios de aceptación, por lo que tendrás que trabajar con arrays de elementos de mongo y funciones de update en profundidad.

Criterio de aceptación

Estos criterios son generales y estarán referidos al servidor y base de datos. Cumplirlos es obligatorio para considerar el proyecto completado y que podamos así certificarte como Backend Developer una vez acabes todos los proyectos.

- ☐ El servidor arranca en el puerto **4001** cuando lanzo el comando `npm run start`.
- ☐ El servidor se conecta con una base de datos **MongoDB** en mi equipo local que corre en el puerto 27017.
- ☐ En el README del proyecto están definidos los endpoints que tengo a mi disposición. **Esto ocurre para las dos colecciones diferenciadas:**
 - ☐ Hay un endpoint GET que me permite traer todos los documentos de un tipo de colección.
 - ☐ Hay un endpoint GET que me permite traer un solo elemento por su id de una colección.

- ☐ Hay un endpoint POST que me permite crear un nuevo elemento en la colección correspondiente.
- ☐ Hay un endpoint PUT que me permite editar un elemento por su id en una colección.
- ☐ Hay un endpoint DELETE que me permite borrar un elemento por su id en una colección.
- ☐ El **servidor no se rompe si pido una URL no existente**, es decir, dispone de un middleware para capturar errores 404 o rutas no encontradas.
- ☐ Los **códigos de red son correctos** cuando hago peticiones (200 y 201 para objetos creados).
- ☐ De cara a la relación entre modelos, en el README tendré explicados los endpoints que debo consumir para tener acceso a:
 - ☐ Un endpoint GET que me permite recuperar un elemento de la colección A y traer los datos de los elementos de la colección B con los que esté relacionado.
 - ☐ Un endpoint GET que me permite recuperar un elemento de la colección B y los datos de un elemento de la colección A con el que esté relacionado.
 - ☐ Un endpoint PUT que me permite añadir (o quitar si ya existe) un nuevo elemento B al array de elementos relacionado de un documento de la colección A.
 - ☐ Un endpoint PUT que me permite cambiar o eliminar el campo de un documento de la colección B que apunta a un elemento de la colección A con el que está relacionado.