

Memoria parcial 2 Ingeniería Web

Trabajo hecho por Mario Merino Zapata

URL Repositorio GITHUB: <https://github.com/MarioMerZap/Ingenier-aWebExamen2.git>

Explicación de la base de datos y entidades

Tarea:

responsable: {

type: *String*,

required: true,

match: /.+\@.+\..+\/ // Validación simple para correos electrónicos

},

- Persona encargada de la tarea

descripcion: {

type: *String*,

required: true,

maxlength: 50 // Máximo 50 caracteres

},

- Descripción de la tarea

habilidades: {

type: [*String*] // Arreglo de habilidades como términos

},

- Habilidades recomendadas de la tarea

segmentos: {

type: *Number*, // Duración en segmentos de 1 hora

required: true,

min: 1 // Asegurarse que sea al menos 1 segmento

}

- Tiempo en horas que tarda la tarea

Colaborador:

email: {

type: *String*,

require: true,

match: /.+\@.+\.+/ // Validación simple para correos

electrónicos

-Email del colaborador

},

nombre: {

type: *String*,

require: true

-Nombre con apellido del colaborador

},

habilidades: {

type: [*String*] // Arreglo de habilidades como términos

}

-Lista de términos que tendrán las habilidades (en este caso informáticas) del colaborador listadas.

En el código del ejercicio los dos incluyen al final las timestamps necesarias.

De manera que como se ve arriba he creado 2 entidades (llamadas Tarea y Colaborador respectivamente), las cuales

Todo este ejercicio se encontrará en una base de datos local, siendo el enlace un localhost. Además, el puerto a utilizar para el backend será el 9999

Tecnologías utilizadas

- **Node.js**

Runtime de JavaScript

- **Express.js**

Express.js es un framework de desarrollo de JavaScripts, utilizado para desarrollar el backend de nuestra aplicación.

- **MongoDB**

Base de datos no relacional que hemos utilizado para almacenar la información que estamos manejando.

- **Docker**

Para el despliegue de los microservicios y la base de datos local.

LIBRERÍAS

- **Nodemon**

Empleada para depurar más fácilmente el código, ya que relanza la aplicación cada vez que guardas un fichero.

- **dotenv**

Usada para manejar variables de entorno, como la URI de conexión a la base de datos.

- **Jest Supertest**

La usamos como herramienta de pruebas de códigos javascript, sirve para hacer los test de integración de la API.

- **Mongoose**

Utilizada para acceder a nuestra base de datos de MongoDB.

Dificultades encontradas en la solución

Como se puede ver en el código, mi solución sigue utilizando los `ObjectId` para buscar los objetos, esto podría llegar a causar problemas, pero en principio debería permitir funcionar perfectamente.

Por desgracia, tampoco he tenido el tiempo suficiente para poder arreglar algunos problemas en las pruebas de postman de la entidad tarea, aunque las de Colaborador deberían de funcionar perfectamente.

Cuando yo exporto el archivo postman se me abre por algún motivo todo puesto en `get`, pero como se puede ver tanto en `ColaboradorRouter` como en `TareaRouter`, sé que cuando tengo que crear algo es `post`, cuando tengo que actualizar es `put` y cuando tengo que borrar es `delete`.