Ejercicios 1.1.-1.6.

1

NB: Se recomienda hacer todos los ejercicios de esta parte en un nuevo repositorio de git dedicado y colocar su código fuente justo en la raíz del repositorio. De lo contrario, tendrá problemas en el ejercicio 1.10.

NB: Inicializa este proyecto con el comando npm init que se demostró anteriormente en esta parte del material.

Fuerte Recomendación: cuando esté trabajando en código de backend, siempre esté atento a lo que sucede en el terminal que ejecuta su aplicación.

1.1: backend de la agenda telefónica, paso 1

Implemente una aplicación Node que devuelva una lista hardcodeada de entradas de la agenda telefónica desde la dirección http://localhost:3001/api/persons:

```
← → C ① localhost:3001/api/persons
[
 - {
       id: 1,
       name: "Arto Hellas",
       number: "040-123456"
   },
 - {
       id: 2,
       name: "Ada Lovelace",
       number: "39-44-5323523"
   },
 - {
       id: 3,
       name: "Dan Abramov",
       number: "12-43-234345"
   },
 - {
       id: 4,
       name: "Mary Poppendick",
       number: "39-23-6423122"
   }
```

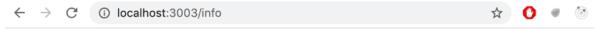
Observe que la barra inclinada en la ruta api/persons no es un carácter especial y es como cualquier otro carácter dl string.

La aplicación debe iniciarse con el comando npm start.

La aplicación también debe ofrecer un comando npm run dev que ejecutará la aplicación y reiniciará el servidor siempre que se realicen cambios y se guarden en un archivo en el código fuente.

1.2: backend de la agenda telefónica, paso 2

Implemente una página en la dirección http://localhost:3001/info que se parezca más o menos a esto:



Phonebook has info for 4 people

Sat Jan 25 2020 19:03:51 GMT+0200 (Eastern European Standard Time)

La página tiene que mostrar la hora en que se recibió la solicitud y cuántas entradas hay en la agenda telefónica en el momento de procesar la solicitud.

1.3: backend de la agenda telefónica, paso 3

Implemente la funcionalidad para mostrar la información de una sola entrada de la agenda. La URL para obtener los datos de una persona con la identificación 5 debe ser http://localhost:3001/api/persons/5

Si no se encuentra una entrada para la identificación dada, el servidor debe responder con el código de estado apropiado.

1.4: backend de la agenda telefónica, paso 4

Implemente la funcionalidad que hace posible eliminar una sola entrada de la agenda telefónica mediante una solicitud HTTP DELETE a la URL única de esa entrada de la agenda.

Pruebe que su funcionalidad funcione con Postman o el cliente REST de Visual Studio Code.

1.5: backend de la agenda telefónica, paso 5

Expanda el backend para que se puedan agregar nuevas entradas a la agenda telefónica realizando solicitudes HTTP POST a la dirección http://localhost:3001/api/persons.

Genere una nueva id para la entrada de la agenda con la función Math.random. Use un rango lo suficientemente grande para sus valores aleatorios de modo que la probabilidad de crear identificadores duplicados sea pequeña.

1.6: backend de la agenda telefónica, paso 6

Implemente el manejo de errores para crear nuevas entradas. No se permite que la solicitud se realice correctamente si:

- Falta el nombre o número
- El nombre ya existe en la agenda

Responda a solicitudes como estas con el código de estado apropiado y también envíe información que explique el motivo del error, por ejemplo:

{ error: 'name must be unique' }