Ejercicios 1.1.-1.6.

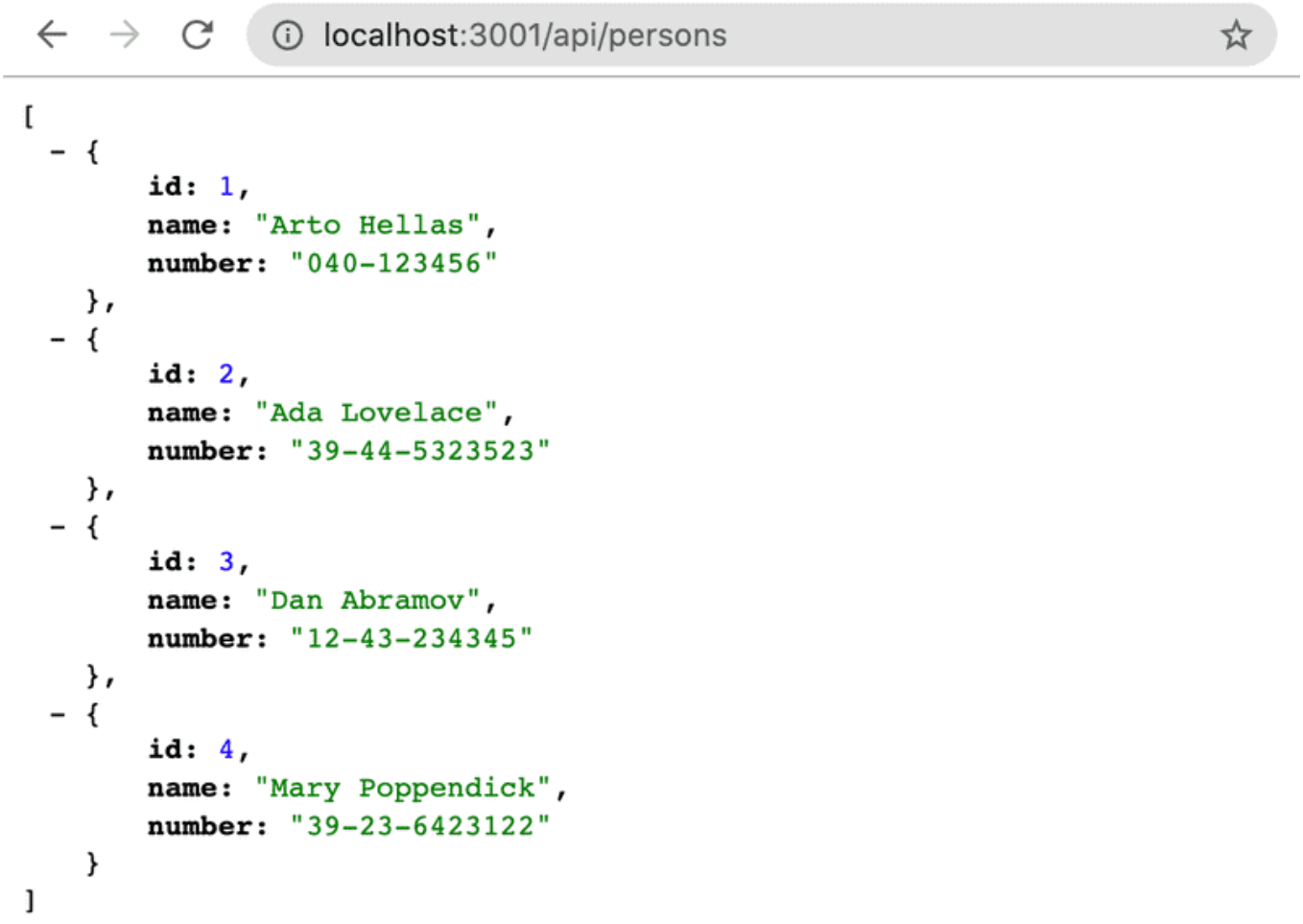
NB: Se recomienda hacer todos los ejercicios de esta parte en un nuevo repositorio de git dedicado y colocar su código fuente justo en la raíz del repositorio. De lo contrario, tendrá problemas en el ejercicio 1.10.

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:



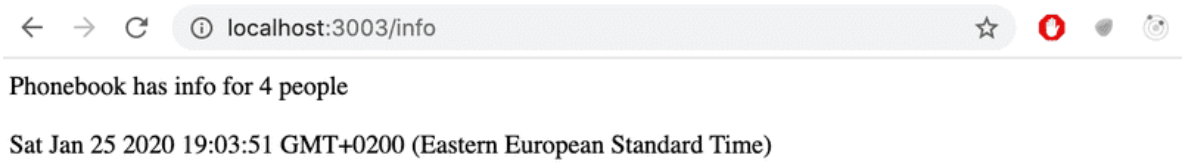
Observe que la barra inclinada en la ruta *api/persons* no es un carácter especial y es como cualquier otro carácter dl string.

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:



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' }
```

1.7: backend de la agenda telefónica, paso 7

Agregue el middleware morgan a su aplicación para iniciar sesión. Configúrelo para registrar mensajes en su consola según la *pequeña* configuración.

La documentación de Morgan no es la mejor y es posible que deba dedicar algún tiempo a averiguar cómo configurarla correctamente. Sin embargo, la mayor parte de la documentación del mundo cae en la misma categoría, por lo que es bueno aprender a descifrar e interpretar documentación críptica en cualquier caso.

Morgan se instala como todas las demás bibliotecas con el comando npm install. La puesta en funcionamiento de Morgan ocurre de la misma manera que la configuración de cualquier otro middleware mediante el comando app.use.

1.8*: backend de la agenda telefónica, paso 8

Configure morgan para que también muestre los datos enviados en las solicitudes HTTP POST:

```
Server running on port 3001
POST /api/persons 200 61 - 4.896 ms {"name":"Liisa Marttinen","number":"040-243563"}
```

Tenga en cuenta que el registro de datos incluso en la consola puede ser peligroso, ya que puede contener datos confidenciales y puede violar la ley de privacidad local (por ejemplo, GDPR en la UE) o el estándar comercial. En este ejercicio, no tiene que preocuparse por los problemas de privacidad, pero en la práctica, intente no registrar ningún dato sensible.

Este ejercicio puede resultar bastante complicado, aunque la solución no requiere mucho código.

Este ejercicio se puede completar de diferentes formas. Una de las posibles soluciones utiliza estas dos técnicas:

- creando nuevos tokens
- JSON.stringify