

Documentación de end-points de la API

Antes de comenzar

Antes que nada, mencionar en caso de no saberlo, que las instrucciones para la instalación están en el README del repositorio ubicado en el siguiente link: [IsraelF55/prueba-tecnica-backend](https://github.com/IsraelF55/prueba-tecnica-backend). La API consta de 6 end-points codificados en el fichero routes.js, se ordenan según los puntos del documento que describe los requerimientos.

La ruta a la que deben de realizarse todas las peticiones es <http://localhost:4000/api/v1/empleados/>

Desarrollo de los end-points

1. Crear Empleado

Para emplear esta funcion el verbo HTTP y ruta es POST /

Ademas, todos los campos son obligatorios. En caso de que falte cualquiera, se enviara un mensaje de error.

La respuesta en caso de error es el objeto err obtenido, en caso de éxito la respuesta es "Empleado creado".

2. Editar Empleado

Para emplear esta funcion el verbo HTTP y ruta es PUT /edit/:id

Ademas, se requiere enviar los campos a editar en un objeto.

La respuesta en caso de error es el objeto err obtenido, en caso de éxito la respuesta es "Empleado editado".

3. Eliminar Empleado

Para emplear esta funcion el verbo HTTP y ruta es PUT /delete/:id

El metodo es PUT debido a que la eliminación es lógica. Esto implica que cuando se consultan empleados estos deben de tener el campo 'status' como 1.

La respuesta en caso de error es el objeto err obtenido, en caso de éxito la respuesta es "Empleado eliminado".

4. Obtener un Empleado

Para emplear esta funcion el verbo HTTP y ruta es GET /:id

Es un end-point breve, puesto que se trata únicamente de consultar el id requerido y que el status es igual a 1.

La respuesta en caso de error es el objeto err obtenido, en caso de éxito la respuesta es un json con la informacion del empleado.

5. Obtener todos los Empleados

Para emplear esta funcion el verbo HTTP y ruta es GET /page=<página_a_mostrar>

Los mismo que lo anterior devuelve todos los empleados con status 1, ademas de emplear la paginación con intervalos de 6. Si no se indica el parámetro page se muestra la pagina 1 por defecto. Si se busca una pagina superior redirige a la ultima pagina disponible. Si se quiere modificar el tamaño del intervalo solo se debe modificar el valor del campo resultsPerPage.

La respuesta en caso de error es el objeto err obtenido, en caso de éxito la respuesta es un json con la informacion de los empleados en esa página.

6. Filtrar por departamentos

Para emplear esta funcion el verbo HTTP y ruta es GET /departamento/:departamento

El trabajo de este end-point es devolver la información de los empleados que trabajan en el departamento recibido mediante el parámetro departamento de la url. Dicho parámetro es numérico.

La respuesta en caso de error es el objeto err obtenido, en caso de éxito la respuesta es un json con la informacion de los empleados que trabajan en el departamento coincidente.