Testeo API REST De Autenticación

Camilo Nicolas Castillo Lemus

Análisis Y Desarrollo De Software, Servicio Nacional De Aprendizaje

2834816: Componente Técnico

Juan Manuel Aldana Zambrano

17 de Diciembre de 2024

Versión de API: 1.0

Introducción

La siguiente API se ha desarrollado cumpliendo las metas propuestas en el material de aprendizaje, donde indican la creación de una API REST que haga el proceso de autenticación de usuarios en los sectores de inicio de sesión y registro de usuarios, esto como premisa principal. Junto a estos requerimientos se ha decidido extender el funcionamiento con el objetivo de que pueda empezar a funcionar en el proyecto de software formativo final, el cual es un software hotelero.

Las funciones extras que se han empezado a implementar son las funciones de CRUD para el apartado de los empleados en el cual ellos podrían o podrán crear, leer, actualizar y eliminar reservas en la necesidad que surja, esto es necesario para mantener una unidad en la información que se presenta online y físicamente (Ya que un hotel no opera digitalmente como única opción).

La API REST al tener una finalidad mayor a la especificada en la guía, va a seguir su mejoramiento de la fecha de entrega del documento en adelante, con un mayor cambio adelantado, **el cual será el traslado del proyecto de Node JS a React JS,** con deseos de que este gran cambio no afecte el funcionamiento de la API.

Teniendo lo anterior en cuenta, este documento tendrá el objetivo de documentar el uso de la API enfocándose principalmente en las pruebas que se realizaron a este, con el fin de comprobar no solo el correcto funcionamiento del programa, sino también ilustrar a los posibles usuarios de la API en un futuro.

URL Base de la API

La URL base de la API será manejada de manera local, es decir, con el enlace de "localhost:(puerto)" será suficiente para ejecutar la API, esto se tendrá que modificar en la medida del proyecto al cual se vaya a implementar, en el caso se ejecuta de manera local con un puerto número 3000, el URL final quedaría determinado de la siguiente forma: **localhost:3000**.

En el caso de proyección del software hotelero, se espera que todos los controladores de ruta apunten desde www.abrajkudai.com/... y funcionen correctamente con el resto de las redirecciones.

Autenticación

Para poder acceder a las funciones que tiene la API como tal, se debe hacer un proceso de registro de usuario y luego un inicio de sesión, por este último y gracias a los Json Web Tokens, al autenticar las credenciales registradas con las ingresadas, en caso de ser correcto, generará un código que permitirá el acceso a las demás funciones, este código es el token, este token será guardado en una variable llamada "x-access-token" la cual se guardara en los headers de la página, cada token tiene una duración inicial de 24 horas (86400 segundos), una vez expira el token, el usuario tendrá que realizar el inicio de sesión nuevamente para generar un token, en este caso el token tendría que ser puesto manualmente en los headers del postman, sin embargo, ya pensando en producción, este token se almacenaría automáticamente en el perfil de usuario.

Por otro lado, algunas rutas tienen una protección extra, por defecto un usuario sin roles es un usuario tipo "usuario", sin embargo, existen otros roles que nos permiten restringir el uso

y estancia de un visitante a ciertas rutas, en el caso de una ruta de eliminación de usuarios, donde se pueden eliminar muchísimos usuarios, lo mejor es tenerla restringida y lejos de un visitante "usuario" común, por ende se crean los roles de "admin" y "empleado", con esto y junto a los tokens es una capa extra de seguridad a nuestras rutas, sin embargo en este estado aún son vulnerables.

Formato de Solicitudes y Respuestas

Los formatos usados para el desarrollo de testeos fue el de JSON en su gran mayoría, se envían peticiones en JSON cuando son necesarias, en algunas ocasiones cuando la eliminación de registros se hace mediante URL, podría decirse que no usa JSON, pero la premisa principal es un formato de solicitud y respuesta en formato JSON.

Endpoints o Rutas

Los Endpoints utilizados en estas pruebas abarcan casi toda la funcionalidad de los usuarios, para el apartado de autenticación de usuarios, se cuentan con dos Endpoint POST, para el control de usuarios, son cinco Endpoints los cuales incluyen todas las premisas del CRUD (Create, Read, Update & Delete), de igual manera tenemos cinco Endpoints para el control de reservas.

Extendiendo un posible uso futuro, estos diez Endpoint de control solo serian accesibles para los empleados u administradores, debido a que, aunque los usuarios pudieran crear reservas y leerlas, lo mejor es hacer Endpoints específicos para ellos y evitar el posible acceso a funciones mayores que ponen en peligro la función, datos y funcionamiento del hotel.

Los Endpoint se dividen en tres secciones, Reserva, Usuarios y Autenticación:

Reserva

- [GET] Lista de Reservas
- [GET] Obtener reserva por ID
- [POST] Crear reserva
- [PUT] Actualizar reserva (usa el ID)
- [DELETE] Eliminar reserva (usa el ID)

Usuario

- [GET] Lista de usuarios
- [GET] Obtener un usuario por ID
- [POST] Crear un usuario
- [PUT] Actualizar un usuario (usa el ID)
- [DELETE] Eliminar un usuario (usa el ID)

Autenticación

- [POST] Sign In (Iniciar Sesión)
- [POST] Sign Up (Registro)

Códigos de Estado HTTP

Para cada solicitud se ha hecho la intención de crear o generar un status en caso de completitud o fallo de la ejecución de la tarea, sin embargo, he cometido el gran error de solo basarme en W3School el cual decía que todos los códigos 200 eran de operaciones exitosas,

códigos 400 para fallos de tarea y 500 para fallos del servidor, sin embargo, entiendo que si existen ciertos códigos universales, o al menos, en el programa Postman, leía algunos códigos con ciertos nombre por defecto, por ejemplo, el 201 significaba creado, el 204 completado sin mensaje, por ende considero que en este apartado la API aún le falta trabajo.

Ejemplos de Uso

El ejemplo que se presentará a continuación será de el Endpoint que refiere al inicio de sesión de usuario, teniendo en cuenta que ya se haya creado el usuario previamente en el Endpoint de registro de usuario.

Excepción

En caso de que no haberse creado el usuario, desde el Endpoint de Sign Up o Registro de Usuario, se debe configurar el perfil de lectura de Postman para JSON y escribir las siguientes credenciales:

Formato	Contenido

JSON	{
	"nombre": "Luis",
	"apellido": "Perez",
	"contrasena": "Lupe123",
	"correo":"lupe2024@empresa.com",
	"telefono": 1234567890
	}

Tabla 1 – Ejemplo de uso

Inicio de Sesión

Para el inicio de sesión y ejecutándose en el enlace “localhost:3000/auth/signin”, elegiremos el “content-type” del header a “application/json” y procederemos a ingresar en el “Body” el contenido de la siguiente forma:

Formato	Contenido

JSON	{
	"correo":"lupe2024@empresa.com",
	"contrasena": "Lupe123"
	}

Tabla 2 – Ejemplo de uso

Ejecutando este comando ejecutara una estado con código 200 indicando que se ejecutó correctamente la función esperada para dicha ruta, a su vez, este generará un token que será de utilidad para el acceso de otras rutas y funciones.