

UNIVERSIDAD ORT URUGUAY

Obligatorio 1 Diseño de Aplicaciones 2

Documentación

Martín Slepian - 266959

Leopoldo Perez - 257341

Eric Poplawski - 258327

Profesores: Nicolás Fierro, Alexander Wieler, Sofia
Duclos

2024

Link al repositorio: <https://github.com/IngSoft-DA2/257341-266959-258327>

Link al video de Postman: <https://youtu.be/3RsRK7BICTM>

Declaración de autoría

Nosotros, Martín Slepian, Leopoldo Perez y Eric Poplawski declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

1. La obra fue producida en su totalidad mientras cursamos Diseño de Aplicaciones
2. Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
3. Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
4. En la obra, hemos acusado recibo de las ayudas recibidas.
5. Cuando la obra se basa en el trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y que fue construido por nosotros.
6. Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Martín Slepian

7/10/2024

Leopoldo Perez

7/10/2024

Eric Poplawski

7/10/2024

Índice

Carátula.....	1
Declaración de autoría.....	2
Índice.....	3
Abstract.....	4
Funcionalidad: Mantenimiento de cuentas de administrador.....	5
Endpoint: Crear cuenta de administrador.....	5
Caso: Crear cuenta sin autenticarse.....	5
Caso: Crear cuenta de administrador con dueño de hogar o dueño de empresa.	6
Caso: Formato de Email erróneo.....	7
Caso: Administrador creado correctamente.....	7
Caso: Email repetido.....	8
Endpoint: Borrar administrador.....	8
Caso id inválido:.....	8
Caso intentar eliminar un usuario que no es administrador:.....	9
Caso correcto:.....	10
Endpoint: Crear empresa.....	10
Caso: Crear empresa sin autorización:.....	10
Caso: Crear empresa con un administrador o dueño de hogar.....	11
Caso: Crear empresa con campo vacío.....	11
Caso: Crear empresa correctamente.....	12
Caso: Crear una empresa con un dueño de empresa que ya tiene una empresa asociada.....	12

Caso: Crear una empresa con nombre, logo o RUT repetido.....	13
Endpoint: Asociar dispositivos al hogar.....	14
Caso: Asociar dispositivo sin autenticarse.....	14
Caso: Asociar dispositivo con usuario de rol distinto a dueño de hogar.....	14
Caso: Asociar dispositivos a hogar con sesión de usuario con rol dueño de hogar no perteneciente al hogar.....	15
Caso: Asociar dispositivos a hogar con miembro del hogar sin permiso.....	15
Caso correcto:.....	16
Endpoint: Detección de movimiento.....	16
Caso tipo de dispositivo incorrecto:.....	16
Caso motion detection disabled:.....	17
Caso correcto:.....	17

Abstract

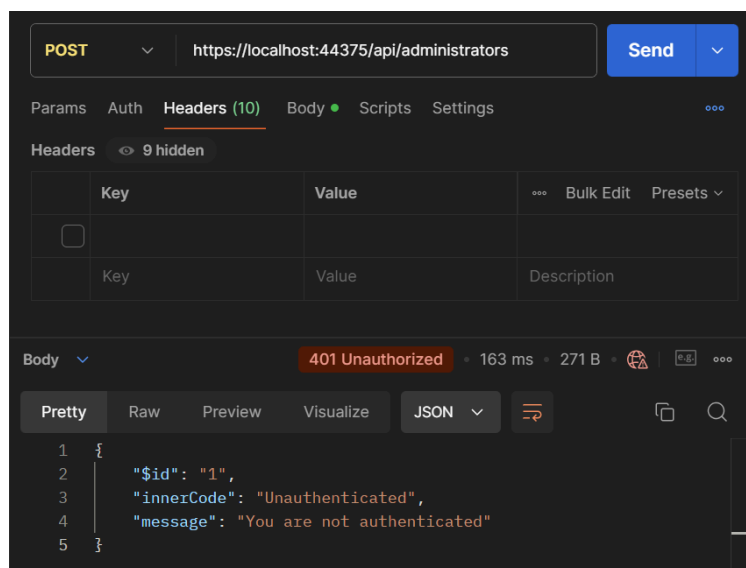
Este proyecto desarrolla una aplicación para la gestión integral de dispositivos inteligentes en el hogar, centrada en la seguridad, la eficiencia y la usabilidad. Utiliza una arquitectura basada en API REST, diseñada con ASP.NET Core y Entity Framework Core, que permite a los usuarios administrar roles y dispositivos, enviar notificaciones y realizar tareas de mantenimiento. Las pruebas fueron realizadas bajo un enfoque TDD, garantizando una alta cobertura y calidad del código. Las funcionalidades clave incluyen la creación y gestión de cuentas de administrador y dispositivos, con un sistema de autenticación robusto y detalladas validaciones de entrada para mantener la integridad de la información. Se ha documentado cada aspecto técnico, con un enfoque especial en la evidencia de todas las pruebas realizadas mediante Postman, lo que asegura que los endpoints cumplan con los requisitos de seguridad y funcionalidad. El enfoque aplicado asegura un sistema robusto, escalable y de fácil integración, ideal para hogares que buscan modernizarse con tecnología inteligente, con la tranquilidad de un exhaustivo proceso de validación.

Funcionalidad: Mantenimiento de cuentas de administrador

Esta funcionalidad la tenemos dividida en dos endpoints distintos, uno que se encarga de crear cuentas de administrador y otro que se encarga de borrar cuentas de administrador.

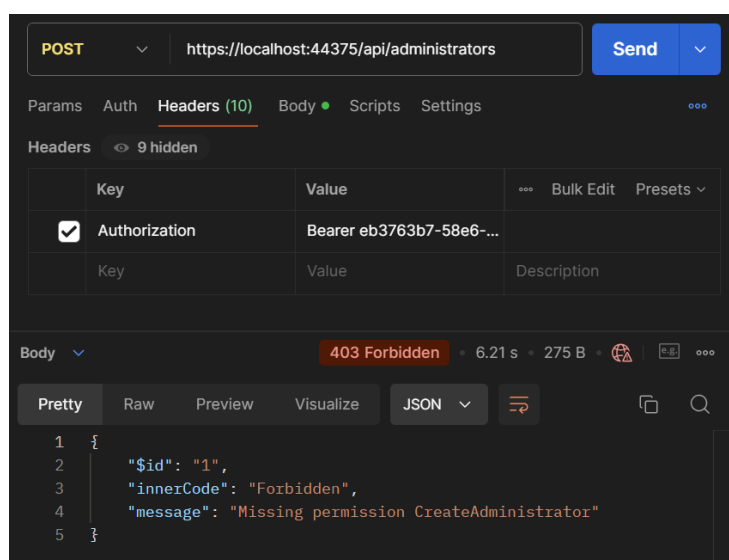
Endpoint: Crear cuenta de administrador

Caso: Crear cuenta sin autenticarse



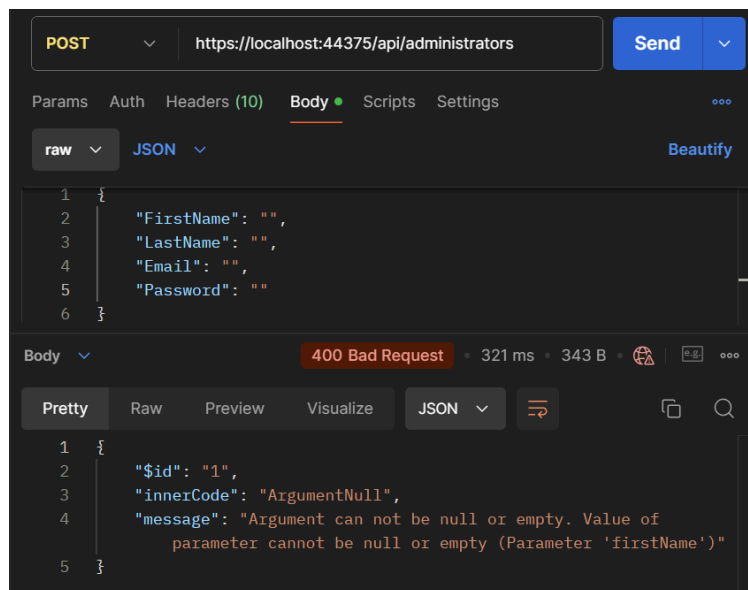
Cuando se intenta crear una cuenta de administrador sin estar autenticado se lanza un error de tipo “401 Unauthorized” ya que se requiere que un administrador esté en sesión para crear una cuenta de administrador

Caso: Crear cuenta de administrador con dueño de hogar o dueño de empresa



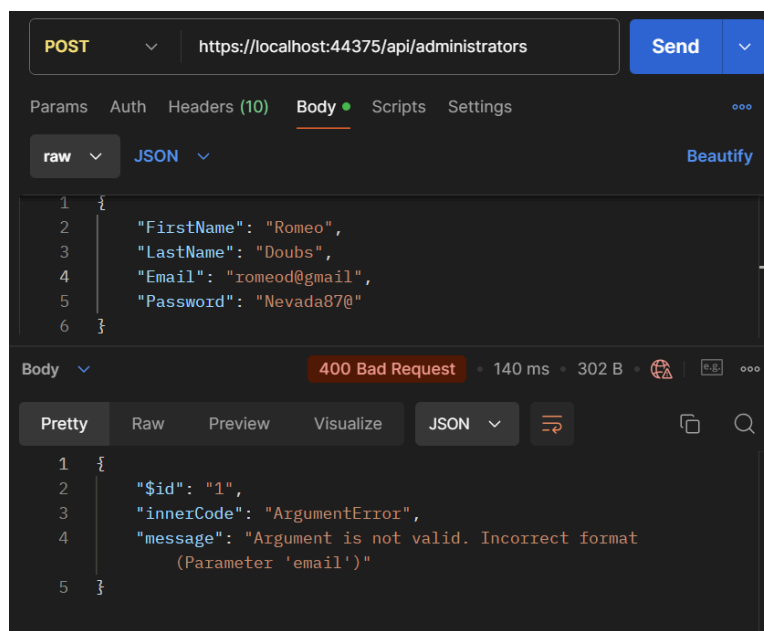
Cuando un usuario que no tiene permiso de crear una cuenta de administrador (usuario con rol dueño de hogar o dueño de empresa) lanza un error de tipo “403 Forbidden” ya que esos usuarios no tienen permisos para crear una cuenta con rol de Administrador.

Caso: Campos vacíos



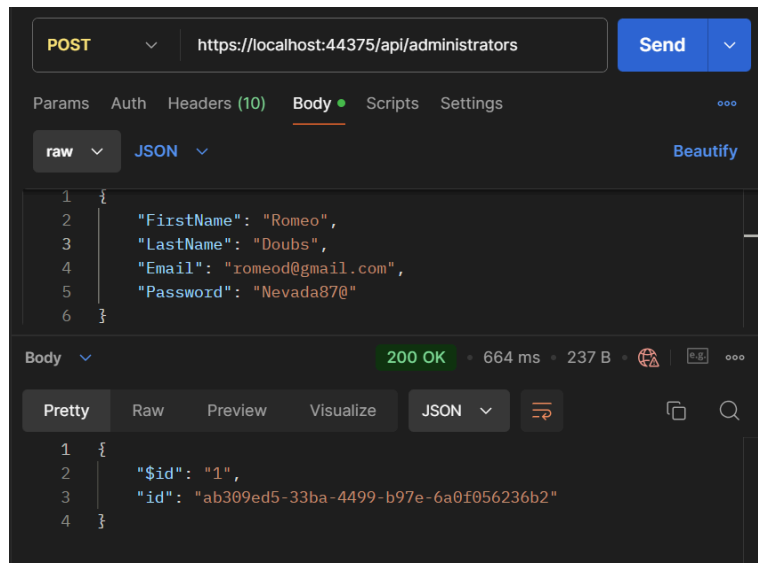
Cuando se deja cualquier campo vacío al intentar registrar un administrador se lanza un error de tipo “400 Bad Request”, ya que todos los campos son obligatorios.

Caso: Formato de Email erróneo



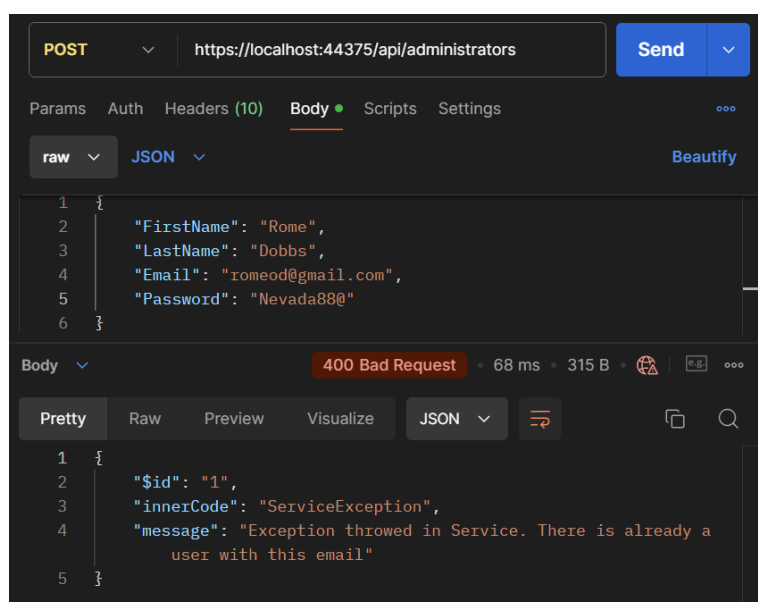
Para todo campo que requiere un correo, se verifica que el formato de este sea correcto, cuando se envía un correo con formato erróneo se lanza un error de tipo “400 Bad Request”.

Caso: Administrador creado correctamente



Cuando los campos se ingresan correctamente y se agrega un header con el token de la sesión de un usuario con rol administrador, se crea un administrador, con respuesta “200 OK” y el id del nuevo administrador en el cuerpo de la respuesta.

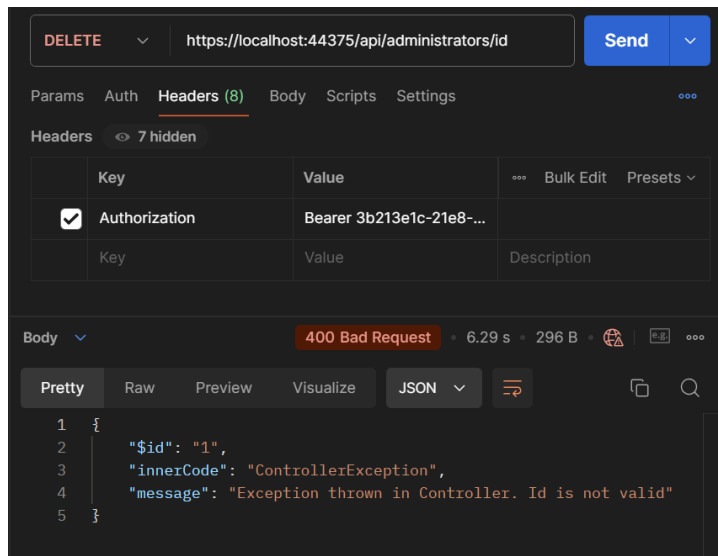
Caso: Email repetido



Cuando intentamos crear un administrador con un email que ya pertenece a un usuario, se lanza un error de tipo “400 Bad Request”.

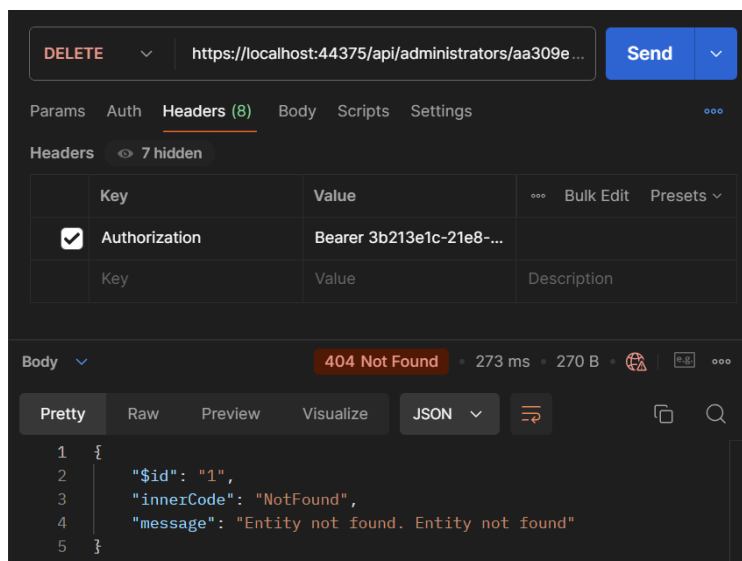
Endpoint: Borrar administrador

Caso id inválido:



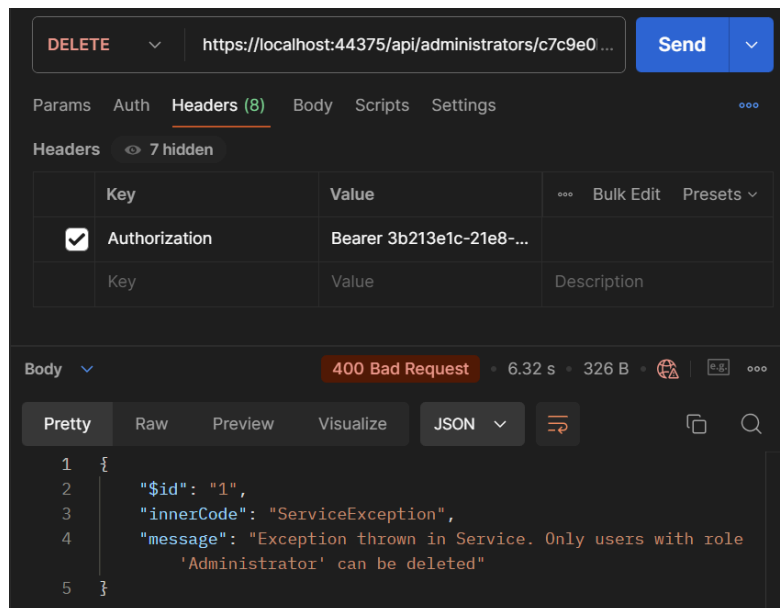
Cuando se intenta borrar una cuenta de administrador con un id con formato inválido lanza un error de tipo “400 Bad Request”.

Caso id inexistente:



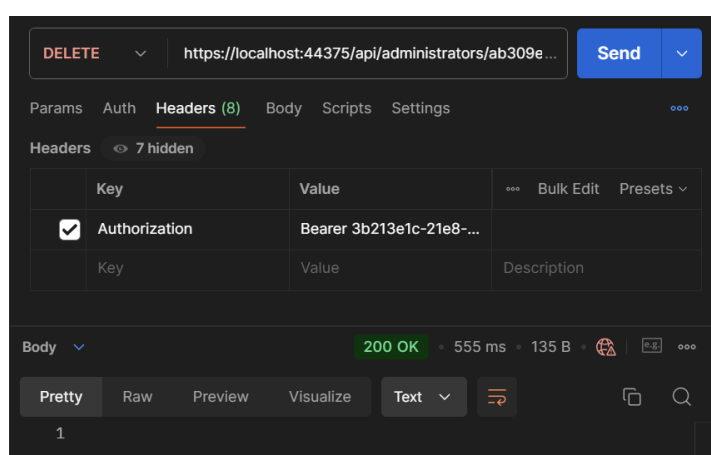
Cuando se intenta eliminar un administrador a través de un id que no está registrado en la base de datos se lanza un error de tipo “404 Not Found”.

Caso intentar eliminar un usuario que no es administrador:



Cuando se intenta eliminar un usuario con otro rol que no es rol de administrador se lanza un error de tipo “400 Bad Request” indicando que solo se pueden eliminar administradores.

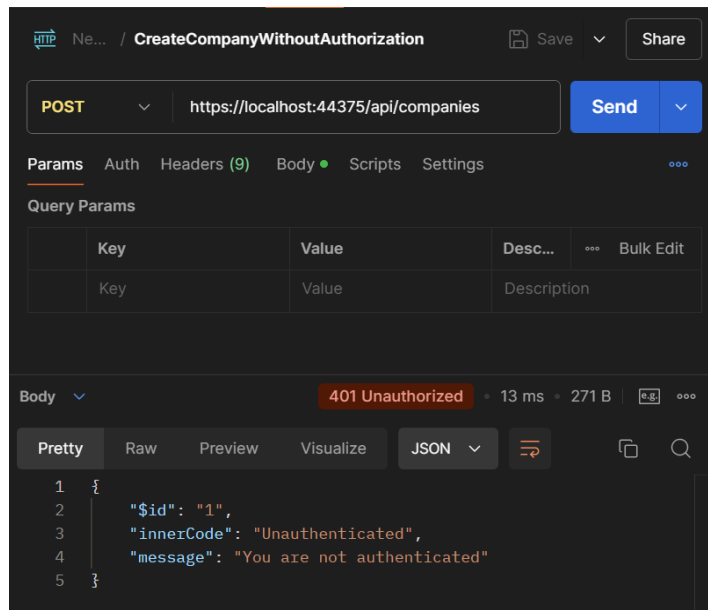
Caso correcto:



Se elimina el administrador correctamente.

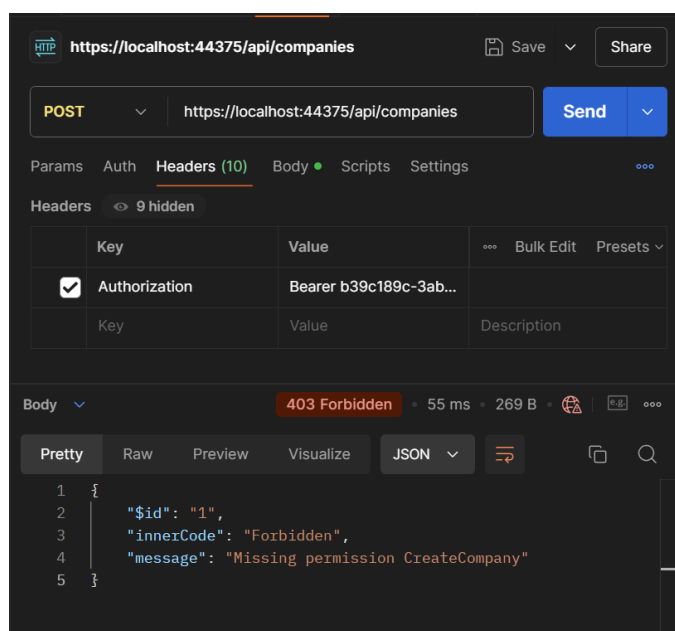
Endpoint: Crear empresa

Caso: Crear empresa sin autorización:



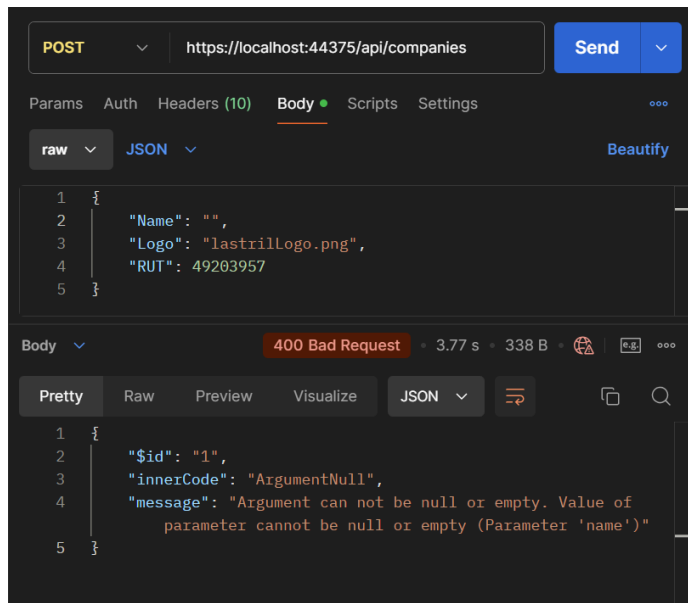
Cuando se intenta crear una compañía sin estar autenticado (sin poner el valor del token de la sesión), se lanza un error de tipo “401 Unauthorized” con mensaje “You are not authenticated”. Esto implica que se requiere estar autenticado para realizar esta acción.

Caso: Crear empresa con un administrador o dueño de hogar



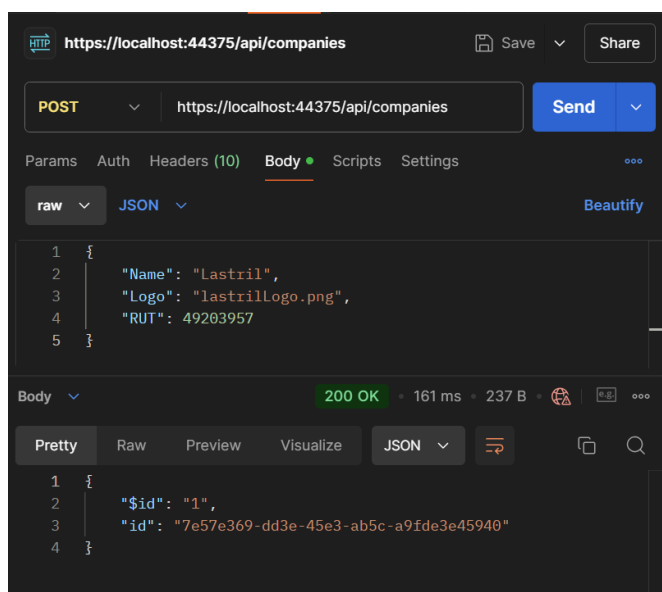
Cuando se intenta crear una compañía en la sesión de un administrador o dueño de hogar se lanza un error de tipo “403 Forbidden”, ya que solo los usuarios de rol dueño de empresa tienen el permiso para crear una empresa.

Caso: Crear empresa con campo vacío



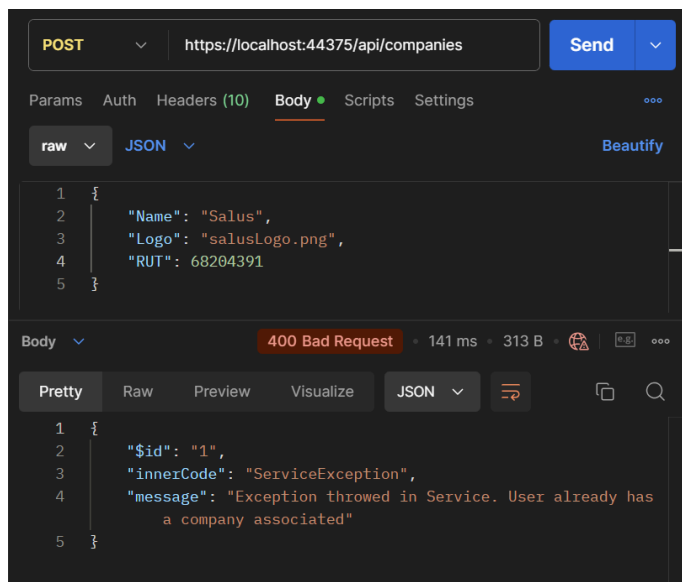
Cuando se intenta crear una empresa con el nombre vacío lanza el error “400 Bad Request”, con un mensaje de que el parámetro “Name” no puede ser vacío. Lo mismo ocurre con el logo.

Caso: Crear empresa correctamente



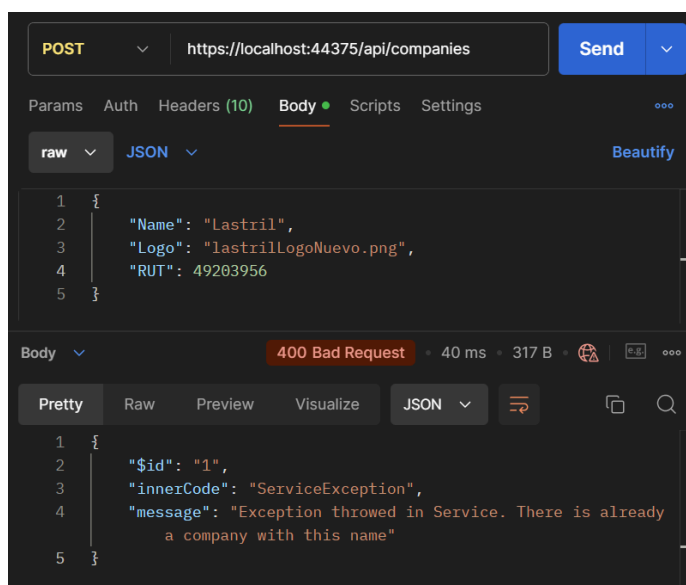
Cuando la sesión es de un dueño de empresa sin ninguna empresa asociada y todos los datos de la empresa son correctos, el response es “200 OK” con un mensaje con el id de la nueva empresa.

Caso: Crear una empresa con un dueño de empresa que ya tiene una empresa asociada



Cuando un dueño de empresa que ya creó una empresa anteriormente intenta crear una nueva empresa lanza un error de tipo “400 Bad Request”, indicando en el mensaje que el usuario ya tiene una empresa asociada.

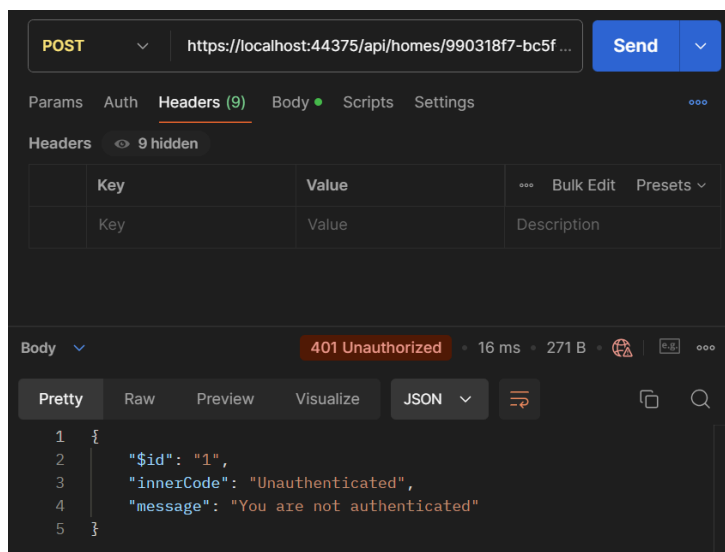
Caso: Crear una empresa con nombre, logo o RUT repetido



Cuando se intenta crear una empresa con un nombre que ya está siendo utilizado por otra empresa lanza un error de tipo “400 Bad Request” con un mensaje de que ya existe una empresa con ese nombre. El nombre de la empresa es único y no puede ser utilizado por otra empresa. Lo mismo ocurre con el logo y RUT:

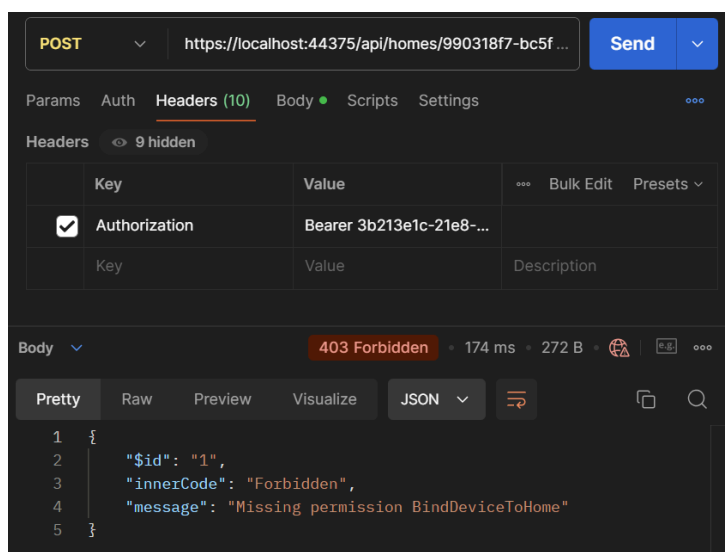
Endpoint: Asociar dispositivos al hogar

Caso: Asociar dispositivo sin autenticarse



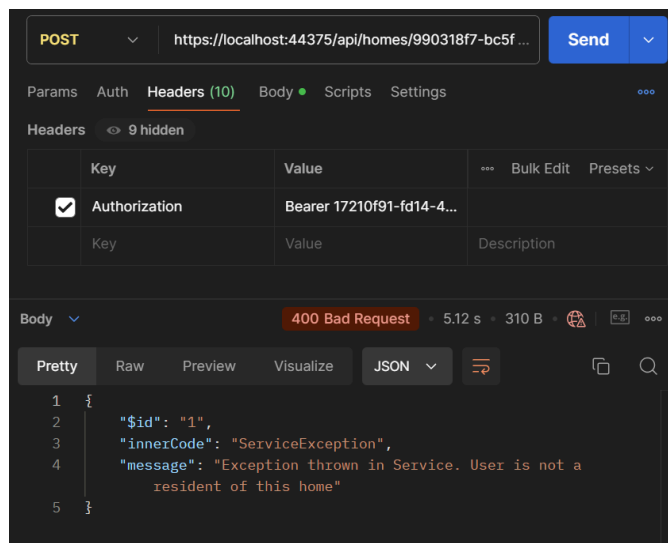
Cuando se intenta asociar un dispositivo a un hogar sin autenticarse lanza un error de tipo “401 Unauthorized”

Caso: Asociar dispositivo con usuario de rol distinto a dueño de hogar



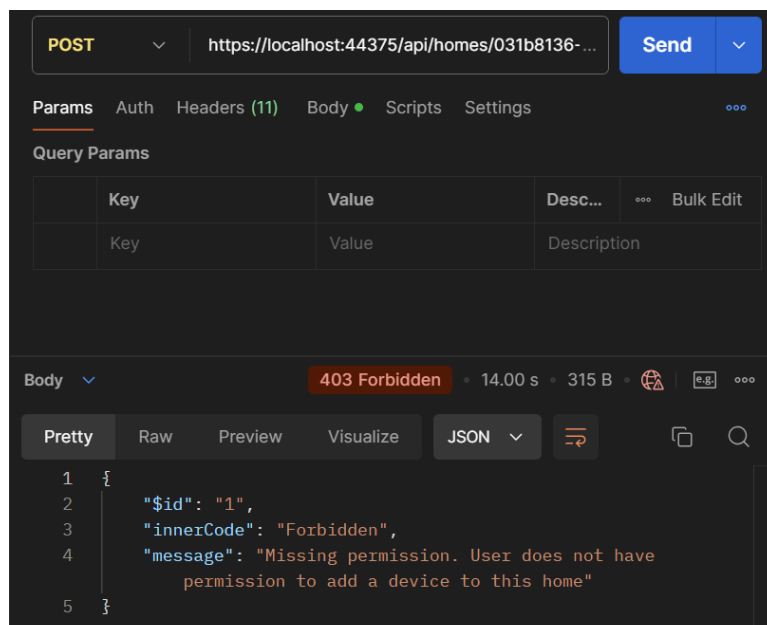
Cuando se intenta asociar un dispositivo a un hogar con la sesión de un administrador o dueño de empresa lanza un error de tipo “403 Forbidden”, ya que no tienen los permisos.

Caso: Asociar dispositivos a hogar con sesión de usuario con rol dueño de hogar no perteneciente al hogar



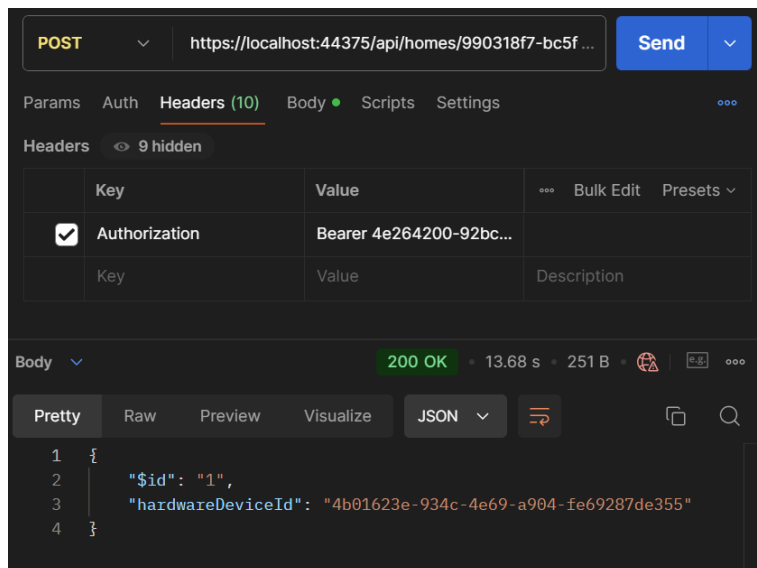
Solo los usuarios pertenecientes al hogar que tengan el permiso de asociar dispositivos pueden realizar la acción, en este caso lanza un error de tipo “400 Bad Request”.

Caso: Asociar dispositivos a hogar con miembro del hogar sin permiso



Cuando un miembro del hogar que no tiene permiso de asociar un dispositivo lo intenta, lanza un error de tipo “403 Forbidden”.

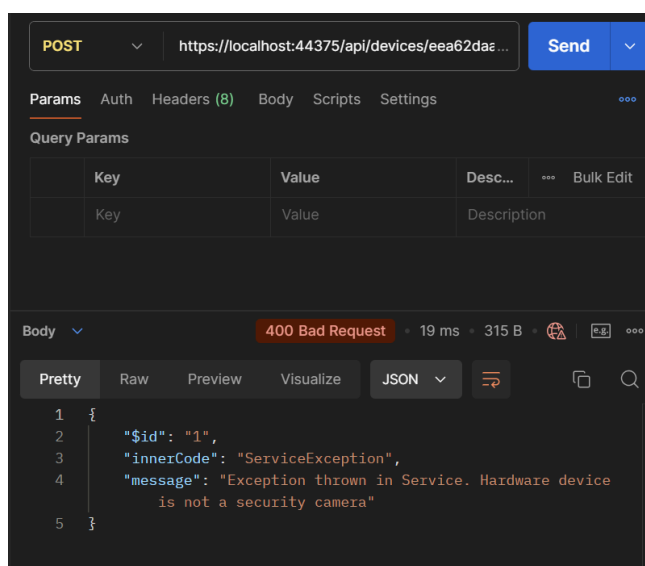
Caso correcto:



Se asocia el dispositivo al hogar correctamente, creando un nuevo hardware device y retornando su id.

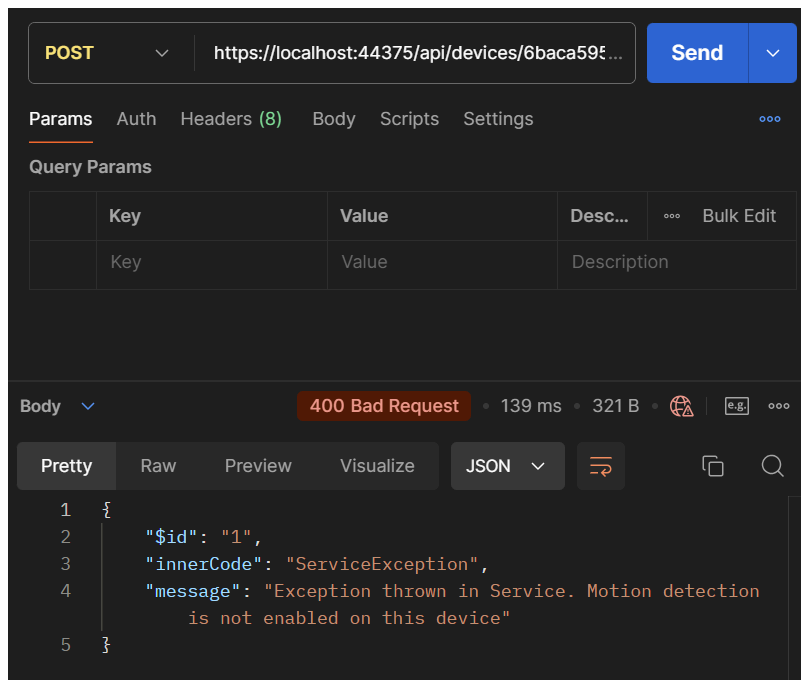
Endpoint: Detección de movimiento

Caso tipo de dispositivo incorrecto:



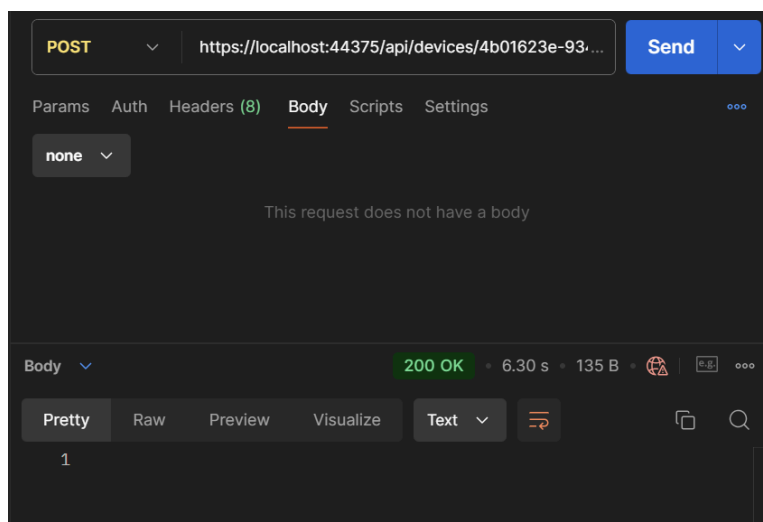
Cuando se trata de capturar movimiento y enviar una notificación con un sensor de ventana se lanza un error de tipo “400 Bad Request”.

Caso motion detection disabled:



Cuando se intenta realizar una acción de capturar movimiento que genera una notificación con un dispositivo que no tiene habilitada la detección de movimiento lanza un error de tipo “400 Bad Request”.

Caso correcto:



Se detecta movimiento y se envía la notificación a los usuarios con ese privilegio dentro del hogar.