

Universidad ORT Uruguay

Facultad de Ingeniería

<https://github.com/IngSoft-DA2-2023-2/276280-243218>

# **1er Obligatorio Diseño de Aplicaciones**

*Evidencia de la ejecución de las pruebas de la API con  
Postman.*

Gonzalo Loureiro – 243218

Santiago Alfonso – 276280

*02 de Mayo de 2024*

# Índice

<b>Índice.....</b>	<b>1</b>
<b>Evidencia de la ejecución de las pruebas de la API con Postman.....</b>	<b>2</b>
Descripción General.....	2
Funcionalidades Clave.....	3
LogIn.....	4
Invitar encargados.....	5
Eliminar Invitaciones.....	6
Mantenimiento de edificio.....	7
Detalles Importantes.....	9

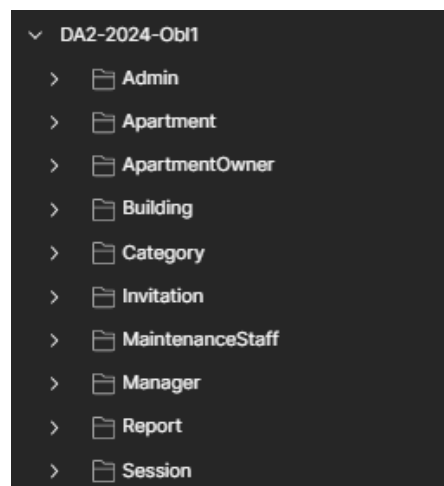
# Evidencia de la ejecución de las pruebas de la API con Postman.

## Link Video Youtube

<https://www.youtube.com/watch?v=ClhNB3-jqG4>

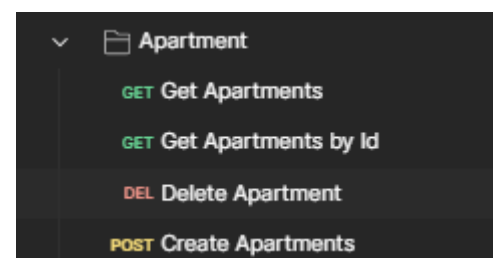
## Descripción General

Como se muestra en la siguiente imagen, se decidió separar las pruebas en carpetas, una por cada controller que creamos



Dentro de cada carpeta tenemos las requests necesarias para probar cada endpoint de cada recurso, por ejemplo, dentro de Apartment, tenemos las siguientes llamadas:

- GetApartments (GET)
  - *https://localhost:7126/api/apartments*
- GetApartmentsById (GET)
  - *https://localhost:7126/api/apartments/1*
- CreateApartment (POST)
  - *https://localhost:7126/api/apartments*
- DeleteApartment (DELETE)
  - *https://localhost:7126/api/apartments/1*



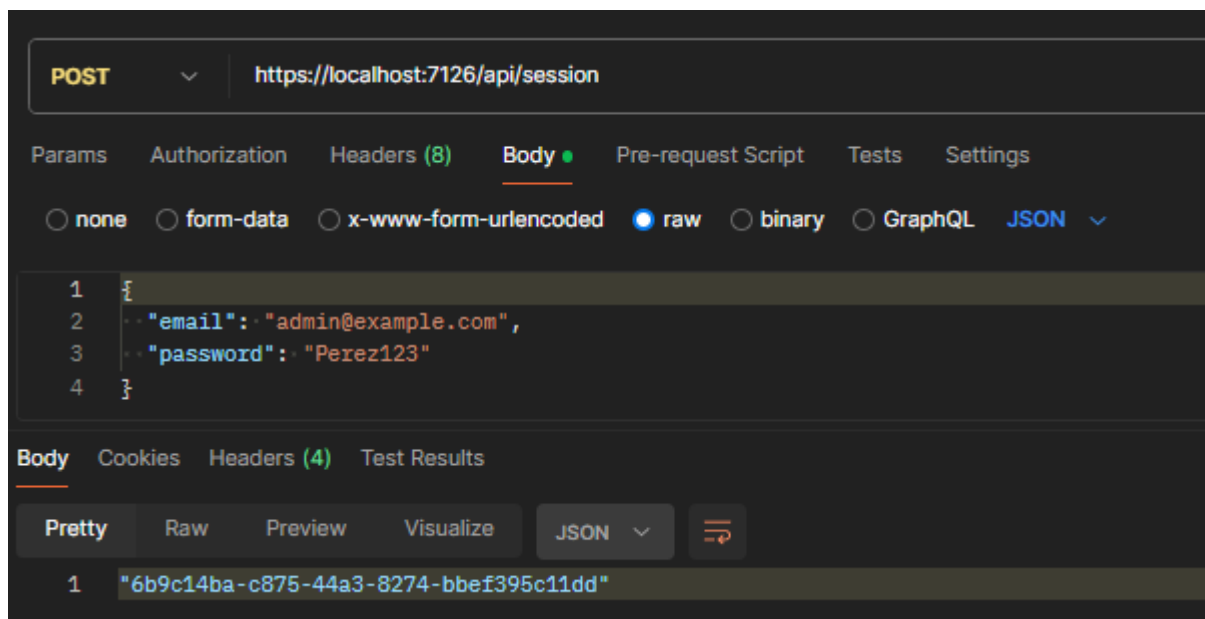
## Funcionalidades Clave

A continuación se detalla cómo se utilizan las funcionalidades marcadas como clave, estas son:

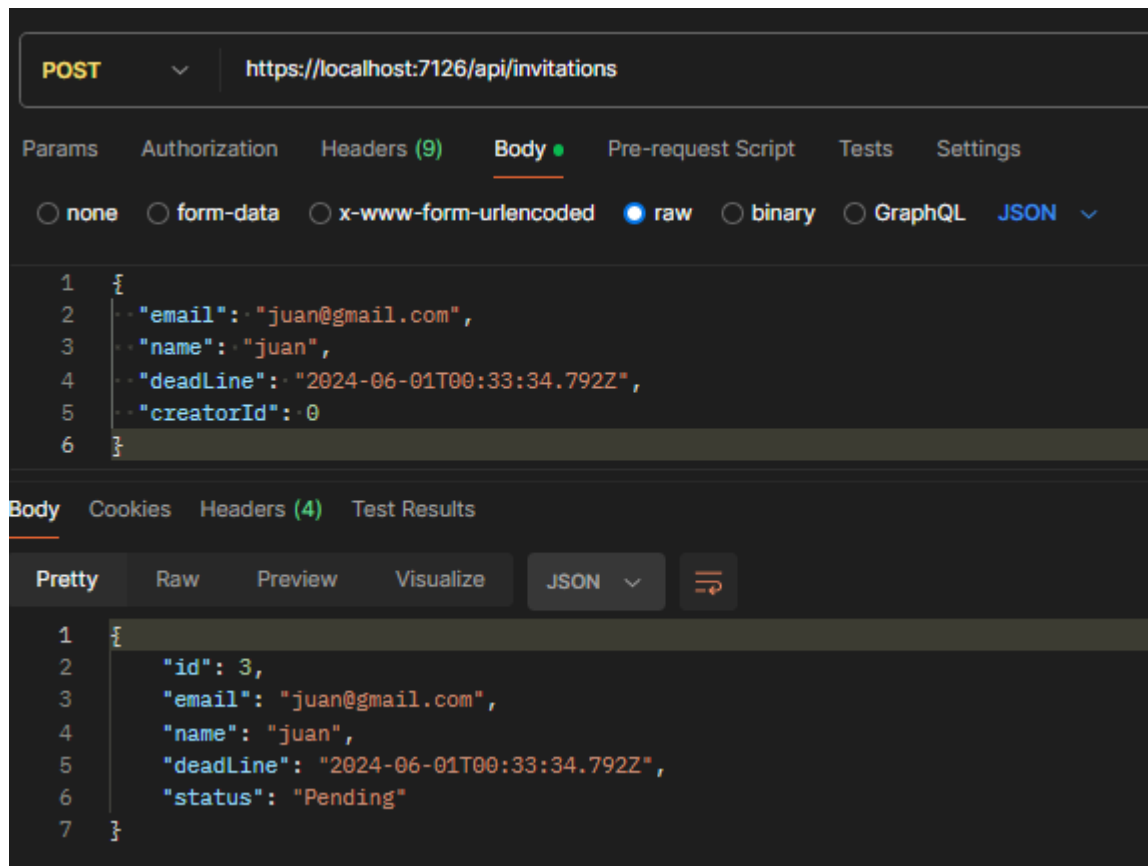
- **Login:** Crear un token para poder realizar las diferentes tareas tanto de administrador como de Encargado
- **Invitar encargados a unirse a la plataforma:** un administrador puede crear invitaciones a personas adjudicándose el rol encargado. El administrador indicará el email del nuevo usuario, nombre y fecha límite para aceptar la invitación.
- **Eliminar invitaciones:** un administrador puede eliminar invitaciones no aceptadas.
- **Mantenimiento de edificio:** un encargado puede mantener un edificio (crear, modificar y eliminar). La información del edificio es la siguiente: nombre (único), dirección, ubicación, empresa constructora, gastos comunes, departamentos. La empresa constructora puede ser una existente o una nueva para el sistema indicando el nombre de esta. Los departamentos cuentan con la siguiente información: piso, número, dueño, cantidad de cuartos, cantidad de baños, con o sin terraza. La información del dueño del departamento es: nombre, apellido y email.

## Login

Por default, el sistema viene con un administrador cosa de ya poder realizar las tareas iniciales. Como se ve en la imagen siguiente, hacemos una llamada al endpoint de session con los datos del usuario (email y contraseña), en este caso el administrador original (admin), para así obtener un token en formato GUID que nos permitirá autenticarnos como administradores y poder hacer las tareas asignadas a los mismos.



## Invitar encargados



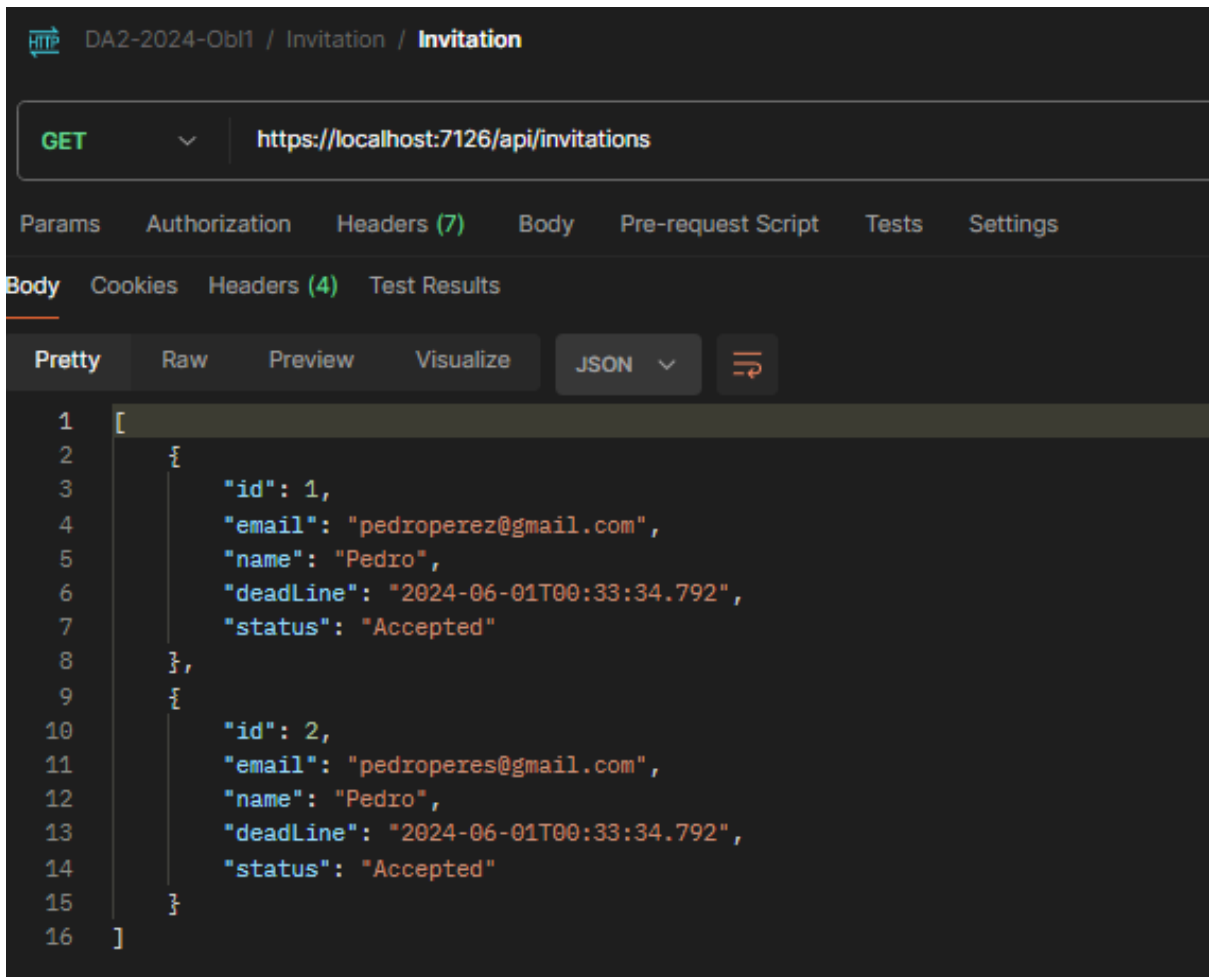
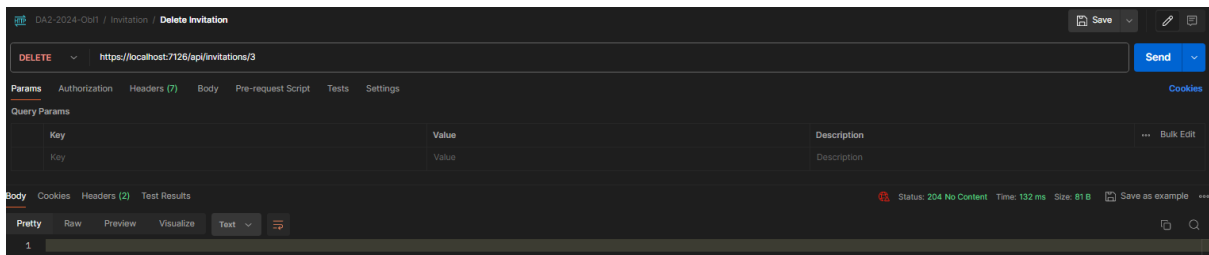
Utilizando el endpoint de **invitations**, un administrador puede crear una invitación para un usuario con el rol de encargado. Se debe hacer una llamada **POST** a `https://localhost:7126/api/invitations` con los datos del usuario a agregar. Este usuario luego hace una llamada **PUT** al mismo endpoint con el siguiente body:

```
{
  "email": "juan@gmail.com",
  "password": "Prueba!23",
  "acceptInvitation": true
}
```

El email debe ser los incluidos en la creación de la request por parte del administrador

## Eliminar Invitaciones

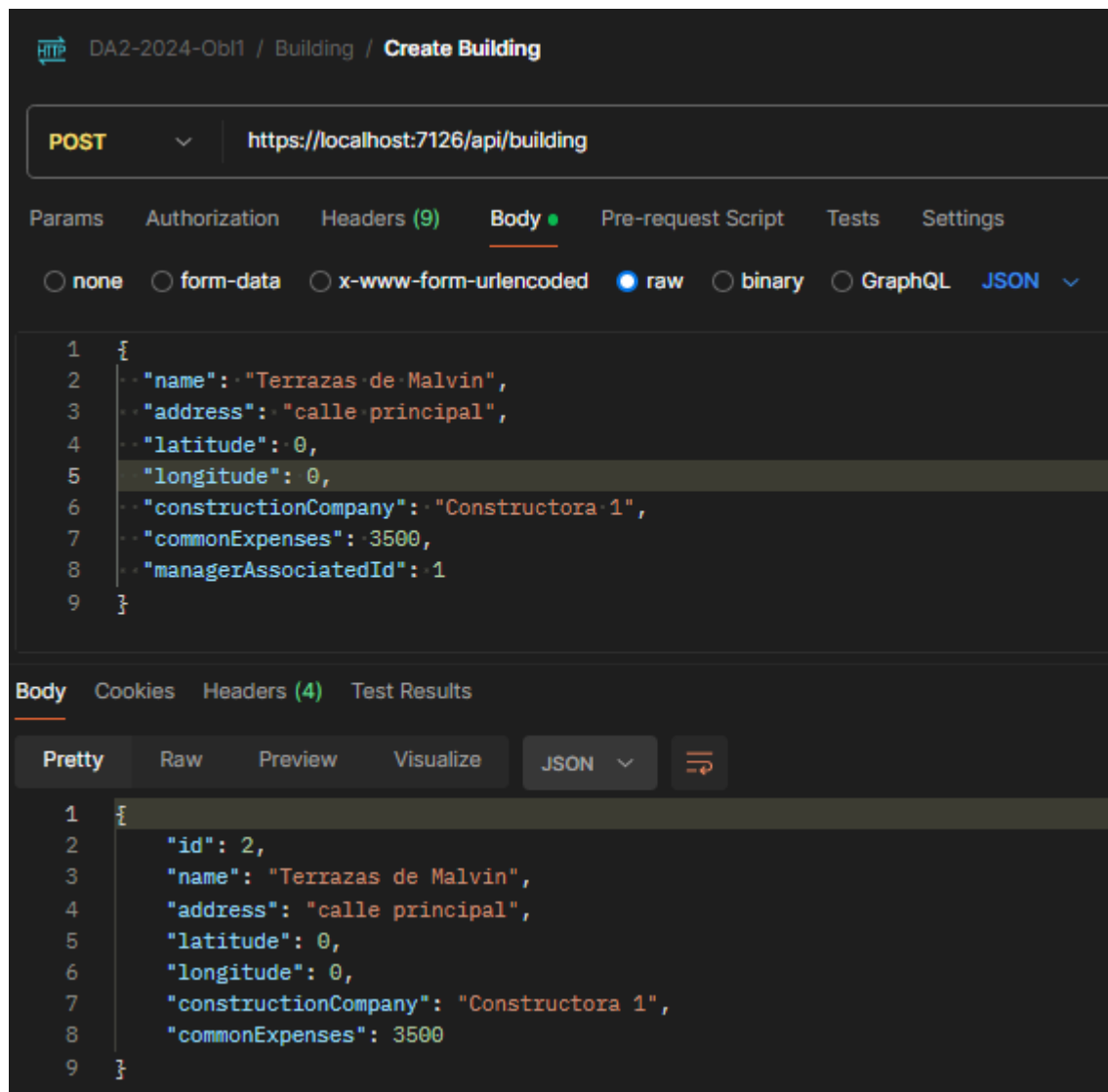
Un administrador no solo puede crear una invitación sino que también eliminar la misma. En este caso se hace por **Id**, se hace una llamada **DELETE** al siguiente endpoint (`https://localhost:7126/api/invitations/{id}`) y esta retorna un estado 204 notificando que se borro correctamente la invitación, para corroborar esto, podemos hacer una llamada **GET** al mismo endpoint pero sin la id



## Mantenimiento de edificio

Un encargado puede administrar un edificio (Crear, modificarlo y borrarlo) utilizando el recurso de building. Para crear un edificio, se hace una llamada **POST** a *https://localhost:7126/api/building* con el siguiente body

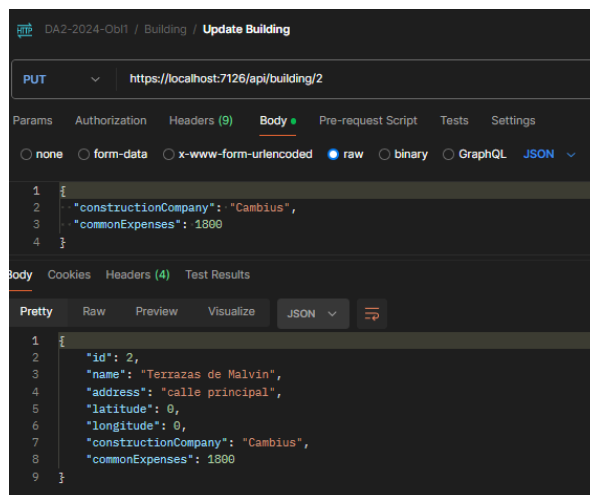
```
{
  "name": "string",
  "address": "string",
  "latitude": 0,
  "longitude": 0,
  "constructionCompany": "string",
  "commonExpenses": 0,
  "managerAssociatedId": 0
}
```



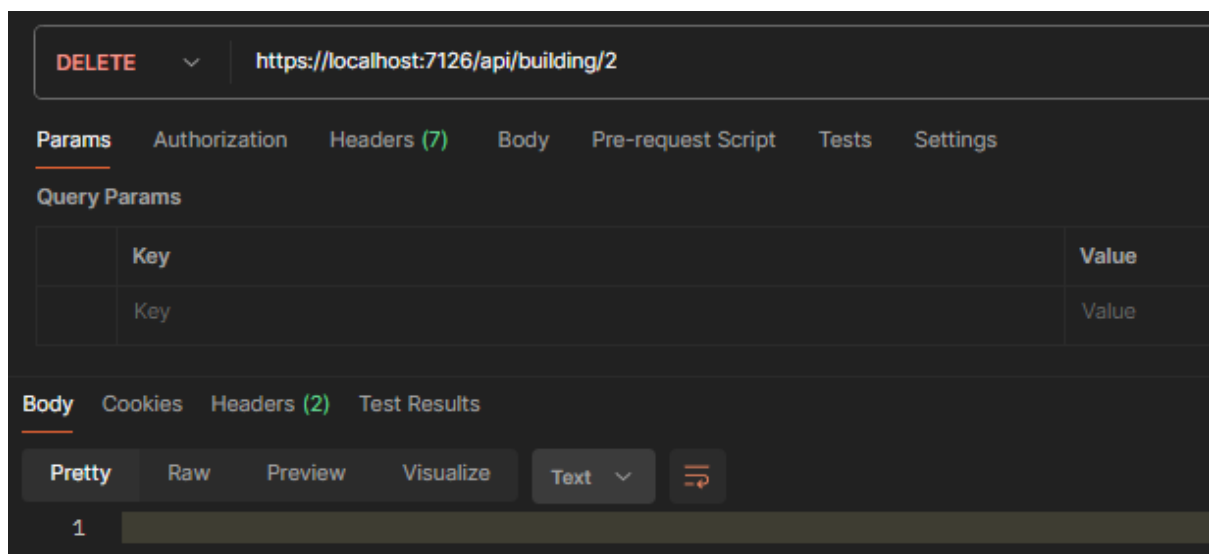


A la hora de modificar el edificio, se hace una llamada **PUT** al siguiente endpoint, *https://localhost:7126/api/building/{id}*. Se incluye en el body de la request la siguiente información:

```
{
  "constructionCompany": "string",
  "commonExpenses": 0
}
```

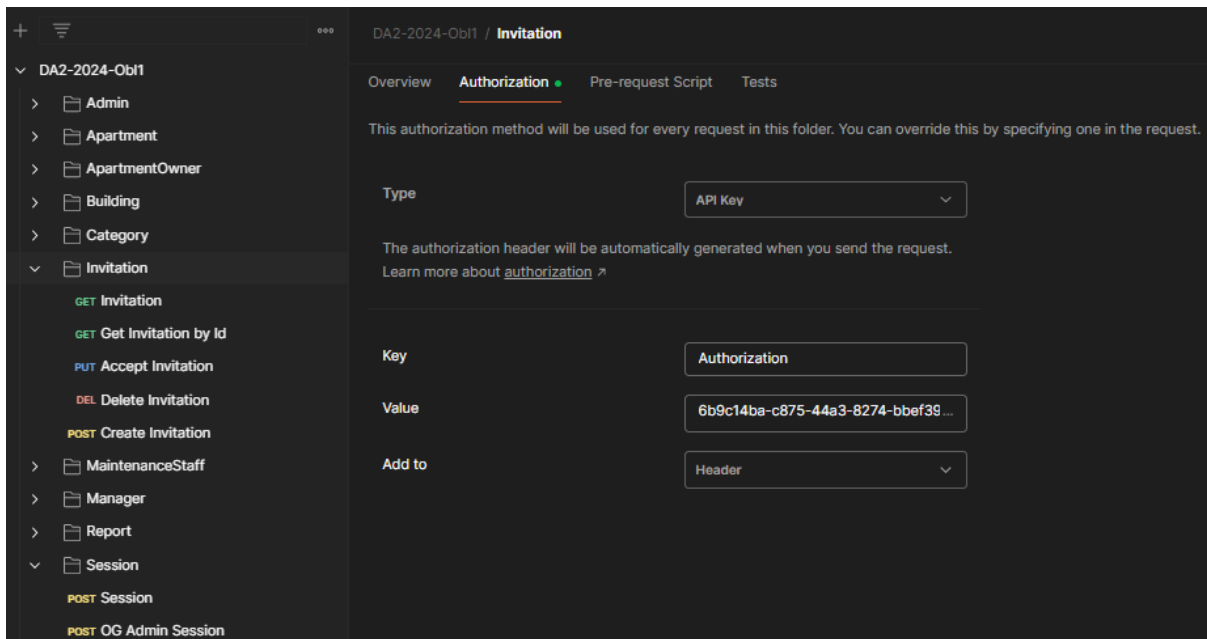


A la hora de eliminar un edificio, se hace una llamada DELETE al mismo endpoint que para modificarlo



## Detalles Importantes

Cada vez que se crea un nuevo usuario, ya sea admin, manager o maintenance y este se autentica, para poder hacer uso de los endpoints se debe agregar la **Authorization** a la carpeta asi las request utilizan la funcionalidad de **Inherit auth from parent** y puede hacer uso del token generado previamente. La imagen siguiente muestra como se debe hacer



A los endpoints de **accept** y **complete** requests se les debe agregar authorization diferente a la carpeta ya que son tareas que debe cumplir un encargado de mantenimiento pero son parte del controlador de manager encargado de las requests.

