

Universidad ORT Uruguay

Facultad de Ingeniería

Bernard Wand Polak

Diseño de aplicaciones 2

Obligatorio 1

Pruebas de la API con Postman

https://github.com/IngSoft-DA2-2023-2/268505_303433_303804

Rodrigo Conze - 268505

Sebastián Borjas – 303433

Agustina Martínez – 303804

Docente: Pablo Geymonat

Abril 2024

Declaración de Autoría

Montevideo, Uruguay, 2 de Mayo de 2024.


Nosotros, Rodrigo Conze, Sebastián Borjas y Agustina Martínez, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

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



Rodrigo Conze

02/05/2024



Sebastián Borjas

02/05/2024



Agustina Martínez

02/05/2024

Abstract

Este documento presenta la evidencia de la ejecución de las pruebas de la API utilizando Postman. Se han generado colecciones de pruebas que cubren todas las funcionalidades de la API, documentando casos de prueba para las funcionalidades marcadas con (*). Se incluyen pruebas con valores inválidos, límites, tipos de datos erróneos, datos vacíos, nulos, omisión de campos obligatorios, entre otros. Se entrega un reporte que muestra el resultado de ejecutar los casos de prueba especificados para la API con Postman, con evidencia registrada en un video publicado en YouTube.

Palabras Claves

Pruebas de API, Postman, casos de prueba, evidencia

Índice

Declaración de Autoría.....	2
Abstract.....	3
Palabras Claves.....	4
Índice.....	5
Introducción:.....	6
Collection en postman.....	6
Pruebas funcionalidades (*)......	6

Introducción:

La ejecución de pruebas de API es una parte crucial del proceso de desarrollo de software para garantizar su calidad y fiabilidad. En este documento, se presenta la evidencia de las pruebas realizadas con Postman para todas las funcionalidades de la API, con un enfoque especial en aquellas marcadas con (*). Se documentan casos de prueba que abarcan variedad de escenarios, desde valores inválidos hasta pruebas de reglas de negocio. El objetivo es proporcionar una guía clara y completa sobre la ejecución de pruebas de la API y asegurar su correcto funcionamiento.

Collection en postman

Adjunto con la documentación, encontrarán una colección de Postman que hemos preparado para facilitar la realización de pruebas de las funcionalidades de la API. Esta colección incluye una serie de solicitudes predefinidas que cubren escenarios comunes de uso de la API. Al proporcionar esta colección, esperamos facilitar y agilizar el proceso de pruebas, permitiendo a los miembros del equipo ejecutar las solicitudes de manera rápida y sencilla para validar el comportamiento de la API. Si tienen alguna pregunta o necesitan ayuda para utilizar la colección de Postman, no duden en contactarnos.

Pruebas funcionalidades (*)

Para garantizar la calidad y el correcto funcionamiento de la API, llevaremos a cabo pruebas utilizando colecciones de pruebas en Postman. Estas colecciones contendrán casos de prueba detallados para todas las funcionalidades de la API. Específicamente, para las funcionalidades marcadas con (*), documentaremos una variedad de casos de prueba, que incluirán valores inválidos, valores límite, tipos de datos erróneos, datos vacíos, datos nulos, omisión de campos obligatorios, campos redundantes, y formato de mensajes inválidos, entre otros aspectos. Nuestro enfoque se centrará en probar las

reglas del negocio, incluyendo escenarios de alta y baja de elementos existentes e inexistentes.

A continuación presentamos los casos de prueba que hemos usado en el video:

Para comenzar las pruebas, necesitamos crear un usuario semilla que actúe como administrador inicial. Dado que todas las operaciones requieren autorización, este administrador inicial será crucial para iniciar sesión y establecer las autorizaciones adecuadas. Sin embargo, la operación de aceptar invitaciones puede realizarse sin autenticación previa.

Usuario semilla que insertamos para comenzar

```
INSERT INTO BuildingAdmin.dbo.[User] (LastName, Name,
Email, Password, Discriminator, BuildingId)

VALUES ('Doe', 'John', 'johndoe@example.com', '123456',
'user', 1);
```

POST /api/v1/buildings

Caso válido:

```
{
  "name": "Edificio A",
  "address": "Cuareim, 1451, Mercedes",
  "location": "1111, 1111",
  "constructionCompany": "Constructora ABC",
  "expenses": 100000,
  "apartments": [
    {
      "floor": 1,
      "doorNumber": 101,
      "ownerName": "Juan",
      "ownerLastName": "Perez",
      "ownerEmail": "juan@example.com",
      "rooms": 2,
```

```
"bathrooms": 1,
"hasTerrace": false
}
]
```

Caso con valores inválidos:

```
{
"name": "Edificio",
"address": "Cuareim, 1451",
"location": "Ciudad",
"constructionCompany": "Constructora ABC",
"expenses": -100000,
"apartments": []
}
```

Caso con omisión de campos obligatorios:

```
{
"address": "Cuareim, 1451, Mercedes",
"location": "Ciudad XYZ",
"constructionCompany": "Constructora ABC",
"expenses": 100000
}
```

PUT /api/v1/buildings/{id}

Caso válido:

```
{
"constructionCompany": "Nueva Constructora",
"expenses": 150000,
"apartments": [
{
"id": 1,
"ownerName": "Pedro",
"ownerLastName": "Gomez",
"ownerEmail": "pedro@example.com"
}
```



```
}  
]  
}
```

Caso con omisión de campos obligatorios:

```
{  
  "constructionCompany": "Nueva Constructora"  
}
```

DELETE /api/v1/buildings/{id}

No se requiere un body para este caso.
Probar Id que existe y Id que no existe

POST /api/v1/invitations

Caso válido:

```
{  
  "email": "invitado@example.com",  
  "name": "Invitado",  
  "expirationDate": "2024-05-02T19:07:19.057Z"  
}
```

Caso con formato de fecha inválido:

```
{  
  "email": "invitado@example.com",  
  "name": "Invitado",  
  "expirationDate": "2024/05/02"  
}
```

Caso con omisión de campos obligatorios:

```
{  
  "name": "Invitado",  
  "expirationDate": "2024-05-02T19:07:19.057Z"  
}
```

```
}
```

PUT /api/v1/invitations/{id}

Caso válido:

```
{  
  "expirationDate": "2024-05-02T19:07:44.296Z"  
}
```

Caso con formato de fecha inválido:

```
{  
  "expirationDate": "2024/05/02"  
}
```

Caso con omisión de campos obligatorios:

```
{}
```

DELETE /api/v1/invitations/{id}

No se requiere un body para este caso.

Probar Id que existe y Id que no existe

Esta evidencia será proporcionada a través de un único video publicado en YouTube, donde se demostrará claramente la ejecución de las pruebas. El enlace a este video está incluido en este documento para su fácil acceso.

El siguiente es el enlace al video en YouTube:

<https://youtu.be/anjX5fFbkLY>