|  |  |
| --- | --- |
| **Número de Bug** | 01 |
| **Titulo** | Uploads an image |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

No se tiene claridad del formato de imagen que permite enviar, y no permite los formatos PNG, JPG y SVN, tampoco aclara esta situación en el Response.

**Pasos para la reproducción**

1. Verificar que el identificador petId de la mascota existe en la tabla mascotas.

2. Conectar con el endpoint: /pet/{petId}/uploadImage APIS's Postman.

3. Especificar el uso de metodo HTTP POST.

4. Especificar los encabezados con los parametros:

- petId(path): identificador de la mascota - number

- additionalMetadata(formData): Datos adicionales para pasar al servidor - string

- file(formData): Subir la imagen - archivo.

5. Realizar el envio de datos.

6. Visualizar el status code y la respuesta arrojada.

**Resultado Esperado**

200 successful operation

**Example Value**

**Model**

{

"code": 0,

"type": "string",

"message": "string"

}

**Resultado Actual**

415 Unsopported Media Type

?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<apiResponse>

<type>unknown</type>

</apiResponse>

|  |  |
| --- | --- |
| **Número de Bug** | 02 |
| **Titulo** | Add a new pet to the store |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

Se permite la creación una mascota con un Id existente en el sistema a partir del envió de un json.

**Pasos para la reproducción**

1. Especificar el uso del metodo HTTP POST para crear una nueva mascota.

2. Epecificar los encabezados o body campos: - id: Numero autogenerado - number

- category: Categoria de la mascota (gatos, perros...)

id: identificador de la categoría - number

name: nombre de la categoría - string

- name: nombre de la mascota - string

- photoUrls: urls con las fotos de las mascota - string

- tags: palabras clave para buscar las mascotas

id: identificador del tag - number

name: nombre del tag - string

- status: estado de la mascota (available - pending - sold) - string.

3. Realizar el envio de los datos.

4. Visualizar los encabezados de respuesta.

5. verificar el código de respuesta, un mensaje de respuesta y el cuerpo de la respuesta.

**Resultado Esperado**

Se debe mostrar en respuesta el status code debe ser 405 invalid input.

**Resultado Actual**

Status code 200

{

"id": 1,

"category": {

"id": 1,

"name": "string"

},

"name": "name",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "sold"

}

|  |  |
| --- | --- |
| **Número de Bug** | 03 |
| **Titulo** | Find pet by ID |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

Al ingresar el ID de una mascota que ha sigo creada varias veces con el mismo ID responde status code exitoso 200.

**Pasos para la reproducción**

1. Verificar que el identificador petId de la mascota existe en la tabla mascotas.

2. Conectar con el endpoint: /pet/{petid}.

3. Especificar el metodo HTTP POST.

4. Especificar los campos:

- petId: identificador de la mascota - number

- name: Nombre de la mascota - string

- status: estado de la mascota (available, pending o sold) - string

5. Realizar el envio de los datos.

6. verificar el código de respuesta, un mensaje de respuesta y el cuerpo de la respuesta.

**Resultado Esperado**

Se debe mostrar en respuesta el status code debe ser 405 invalid input.

**Resultado Actual**

Status code 200

|  |  |
| --- | --- |
| **Número de Bug** | 04 |
| **Titulo** | Find pet by ID |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

Al ingresar el ID de una mascota que ha sido creada, la búsqueda devuelve un 404 not Found .

**Pasos para la reproducción**

1. Verificar que el identificador petId de la mascota existe en la tabla mascotas.

2. Conectar con el endpoint: /pet/{petid}.

3. Especificar el metodo HTTP POST.

4. Especificar los campos:

- petId: identificador de la mascota - number

- name: Nombre de la mascota - string

- status: estado de la mascota (available, pending o sold) - string

5. Realizar el envio de los datos.

6. verificar el código de respuesta, un mensaje de respuesta y el cuerpo de la respuesta.

**Resultado Esperado**

Se debe mostrar en respuesta el status code debe ser 200 y un body response con un json que devuelva los datos de la mascota solicitada.

**Resultado Actual**

Status code 404 not Found.

|  |  |
| --- | --- |
| **Número de Bug** | 05 |
| **Titulo** | Updates a pet in the store with form data |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

Al ingresar el ID de una mascota que ha sigo creada varias veces con el mismo ID responde status code exitoso 200.

**Pasos para la reproducción**

1. Verificar que el identificador petId de la mascota existe en la tabla mascotas.

2. Conectar con el endpoint: /pet/{petid}.

3. Especificar el metodo HTTP POST.

4. Especificar los campos:

- petId: identificador de la mascota - number

- name: Nombre de la mascota - string

- status: estado de la mascota (available, pending o sold) - string

5. Realizar el envio de los datos.

6. verificar el código de respuesta, un mensaje de respuesta y el cuerpo de la respuesta.

**Resultado Esperado**

Se debe mostrar status code 200 con la mascota buscada..

**Resultado Actual**

Status code 415 Unsupported Media Type.

|  |  |
| --- | --- |
| **Número de Bug** | 06 |
| **Titulo** | Creates list of users withgiven input array |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

Permite crear un usuario con datos invalidos code exitoso 200.

**Pasos para la reproducción**

2. Conectar con el endpoint: /pet/{petid}.

3. Especificar el metodo HTTP POST.

4. Especificar los campos en Body:

**[**

**{**

**"id": 0,**

**"username": "string",**

**"firstName": "string",**

**"lastName": "string",**

**"email": "string",**

**"password": "string",**

**"phone": "string",**

**"userStatus": 0**

**}**

**]**

5. Realizar el envio de los datos.

6. verificar el código de respuesta, un mensaje de respuesta y el cuerpo de la respuesta.

**Resultado Esperado**

Se debe mostrar status code 405.

**Resultado Actual**

Status code 200 OK.

|  |  |
| --- | --- |
| **Número de Bug** | 07 |
| **Titulo** | Creates list of users with given input array |
| **Reportado por** | Valentina Orejuela Albornoz |
| **Fecha** | 26 07 2022 |
| **Sistema Operativo/Navegador** | Windows 11 |
| **Desarrollador Asignado:** | Pepito Perez |
| **Prioridad** | Media |

**Descripción**

Permite crear un usuario con datos invalidos code exitoso 200.

**Pasos para la reproducción**

2. Conectar con el endpoint: /pet/{petid}.

3. Especificar el metodo HTTP POST.

4. Especificar los campos en Body:

**[**

**{**

**"id": 0,**

**"username": "string",**

**"firstName": "string",**

**"lastName": "string",**

**"email": "string",**

**"password": "string",**

**"phone": "string",**

**"userStatus": 0**

**}**

**]**

5. Realizar el envio de los datos.

6. verificar el código de respuesta, un mensaje de respuesta y el cuerpo de la respuesta.

**Resultado Esperado**

Se debe mostrar status code 405.

**Resultado Actual**

Status code 200 OK.