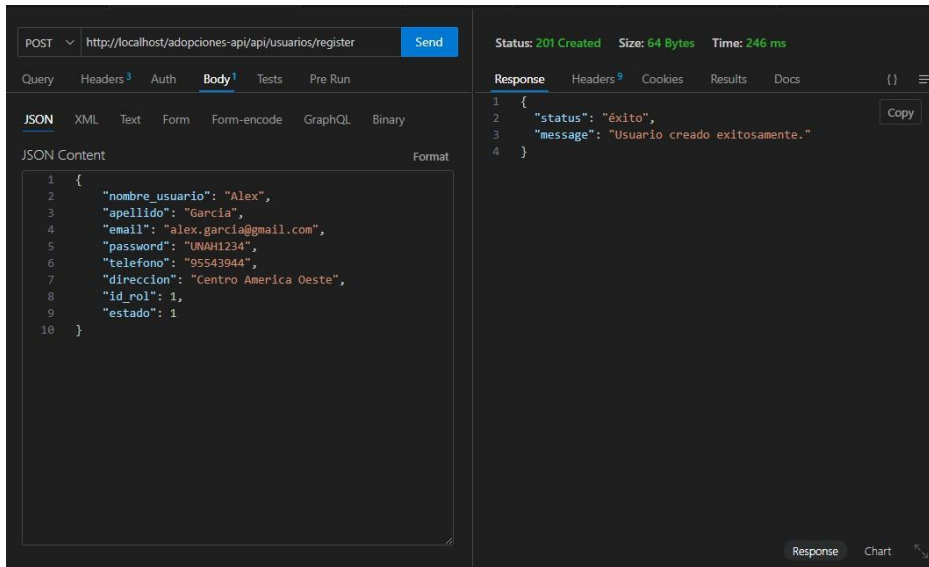


- **Descripción:** Crea un nuevo usuario en el sistema.
- **Método:** POST
- **URL:** http://localhost/adopciones-api/api/usuarios/register
- **Headers:** Content-Type: application/json
- **id\_rol:** 1 para administrador, 2 para usuario normal.
- **estado:** activo, inactivo, pendiente, bloqueado.

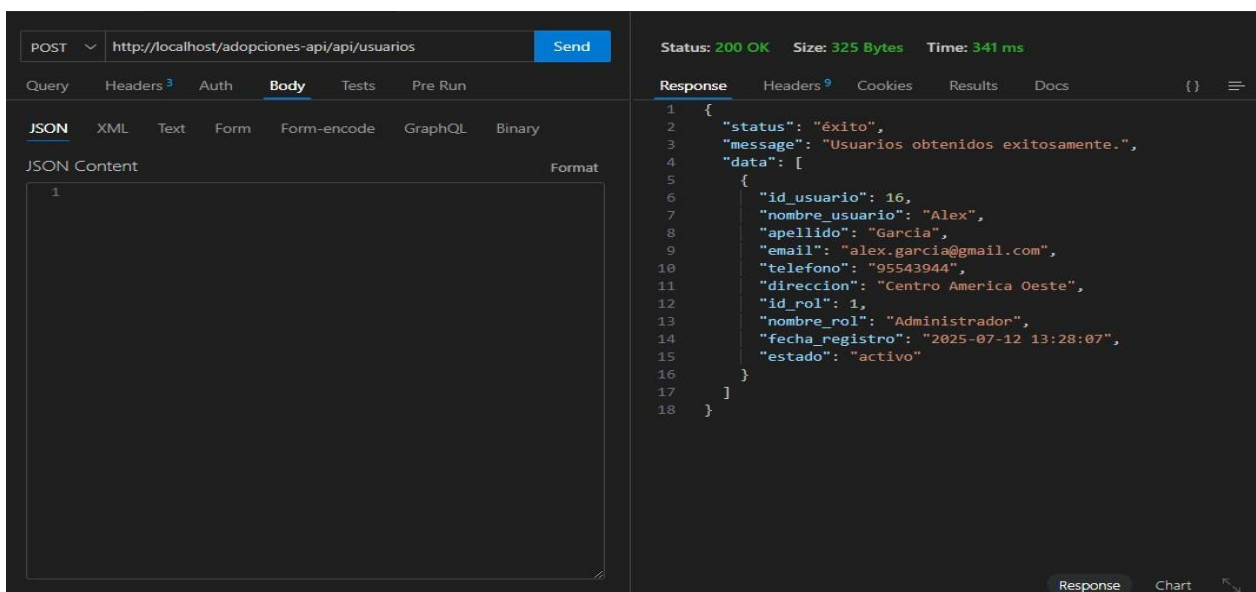
Foto del funcionamiento:



### 1.3. Obtener Todos los Usuarios

- **Descripción:** Recupera una lista de todos los usuarios registrados.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/usuarios`
- **Headers:**
  - Authorization: Bearer TU\_TOKEN\_JWT\_AQUI (Requiere autenticación. Cualquier usuario logueado puede acceder).

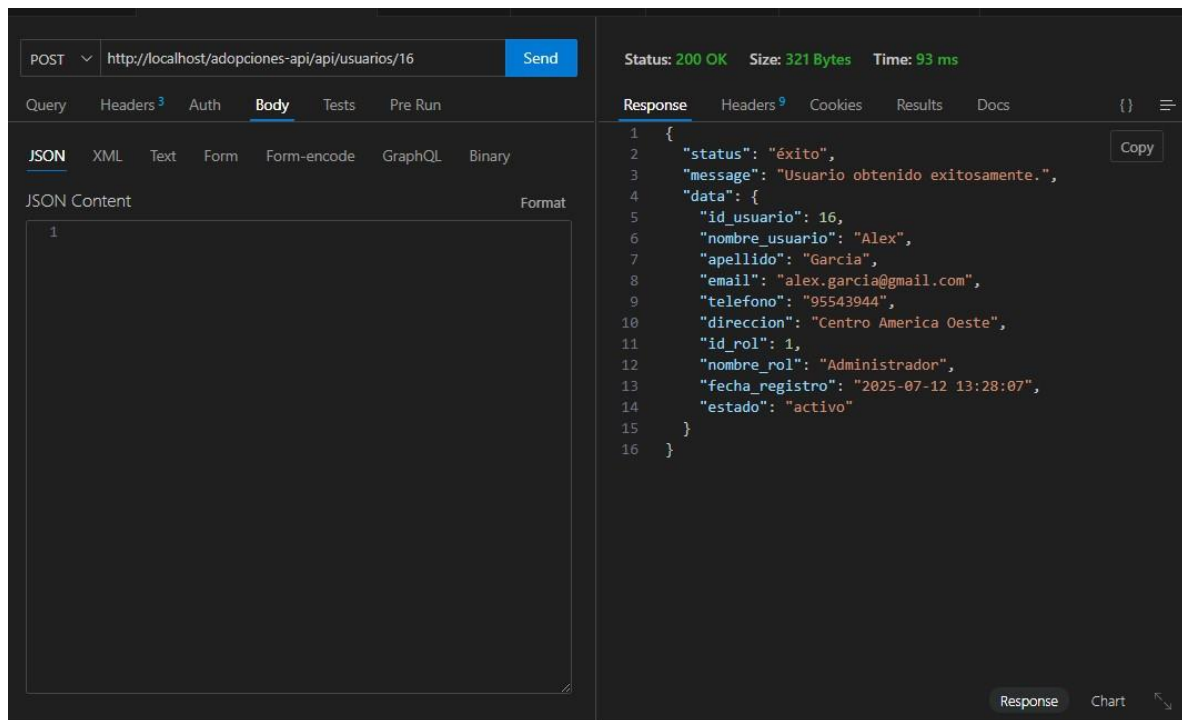
Foto del funcionamiento:



## 1.4. Obtener Usuario por ID

- **Descripción:** Recupera los detalles de un usuario específico por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/usuarios/{id_usuario}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/usuarios/1`
- **Headers:**
  - Authorization: Bearer (Con El Token)

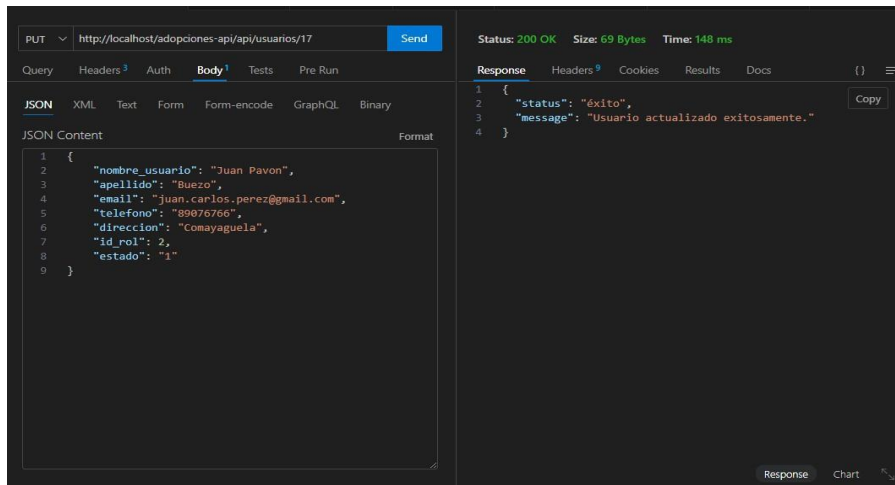
Foto del funcionamiento:



## 1.5. Actualizar Usuario

- **Descripción:** Actualiza los detalles de un usuario existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/usuarios/{id_usuario}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/usuarios/1`
- **Headers:**
  - Content-Type: application/json
  - Authorization: Bearer TU TOKEN JWT AQUI (Requiere autenticación de un **administrador** (id\_rol = 1)).

Foto del funcionamiento:



## ANTES

<input type="checkbox"/>	Editar	Copiar	Borrar	17 Nuevo	Usuario	nuevo.usuario.unico@ejemplo.com	\$2y\$10\$ZuCUi0BR1Ffm3PYjmPUwOSUhKite0Uz/HKqOnwbqK...
--------------------------	--------	--------	--------	----------	---------	---------------------------------	--

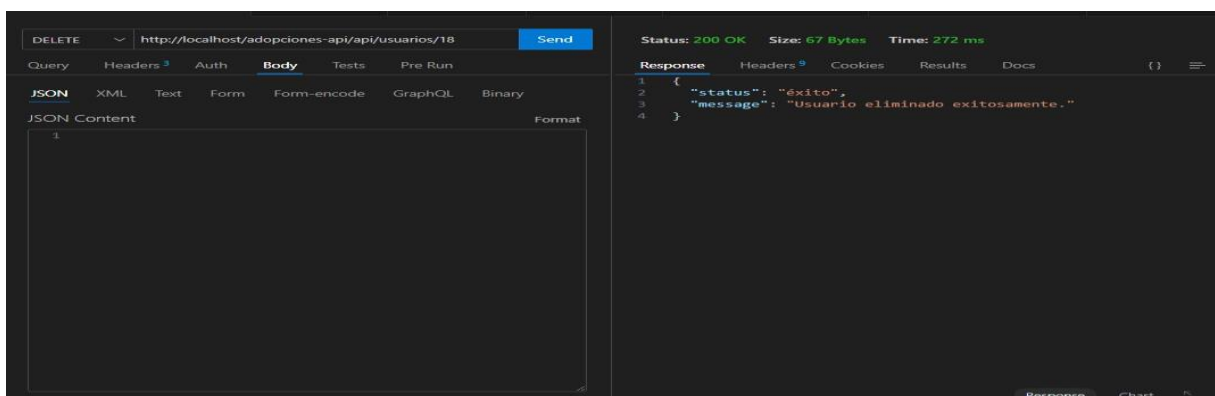
## DESPUES

<input type="checkbox"/>	Editar	Copiar	Borrar	17 Juan Pavon	Buezo	juan.carlos.perez@gmail.com	\$2y\$10\$ZuCUi0BR1Ffm3PYjmPUwOSUhKite0Uz/HKqOnwbqK...
--------------------------	--------	--------	--------	---------------	-------	-----------------------------	--

## 1.6. Eliminar Usuario

- **Descripción:** Elimina un usuario del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adoptaciones-api/api/usuarios/{id_usuario}`
  - **Ejemplo:** `http://localhost/adoptaciones-api/api/usuarios/2`
- **Headers:**
  - Authorization: Bearer (Token Del Administrador)

Foto del funcionamiento:



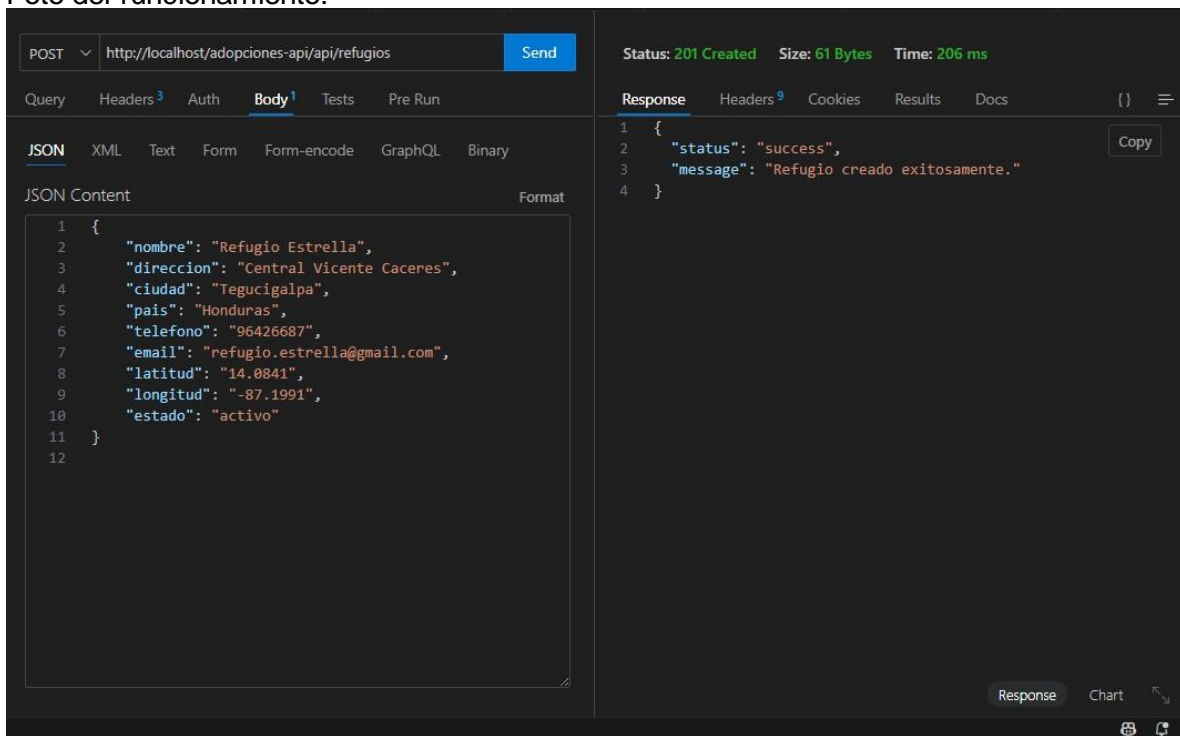
## 2.Refugios

**Base URL:** http://localhost/adopciones-api/api/refugios

### 2.1. Crear Nuevo Refugio

- **Descripción:** Crea un nuevo registro de refugio en el sistema.
- **Método:** POST
- **URL:** http://localhost/adopciones-api/api/refugios
- **Headers:**
  - Content-Type: application/json
  - Authorization: Bearer TU\_TOKEN\_JWT\_AQUI (Requiere autenticación de un **administrador** (id\_rol = 1)).

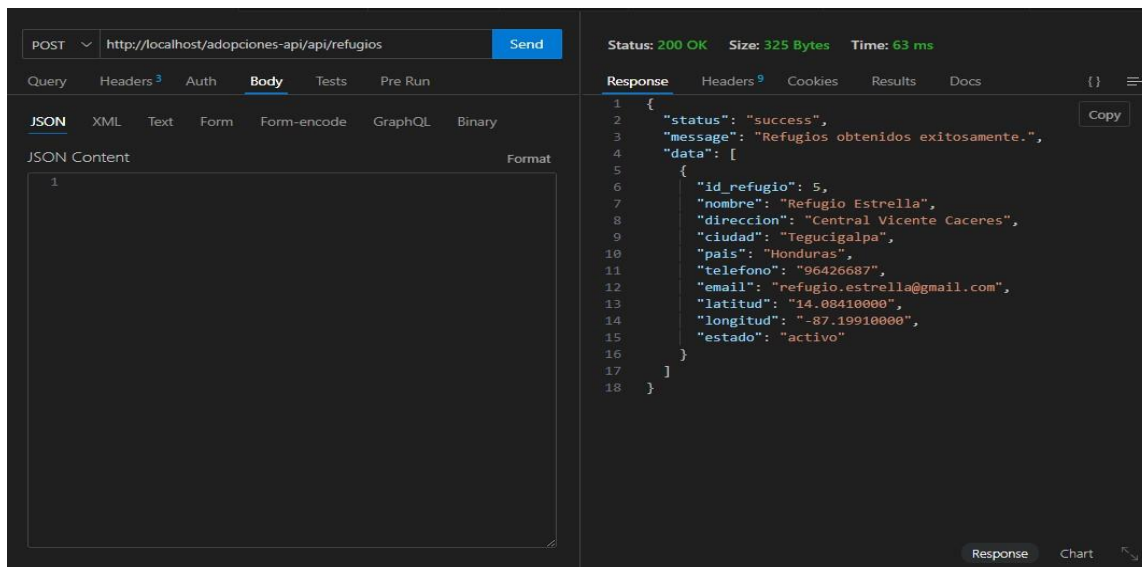
Foto del funcionamiento:



### 2.2. Obtener Todos los Refugios

- **Descripción:** Recupera una lista de todos los refugios registrados en el sistema.
- **Método:** GET
- **URL:** http://localhost/adopciones-api/api/refugios
- **Headers:**
  - Authorization: Bearer (Con Tu Token)

Foto del funcionamiento:

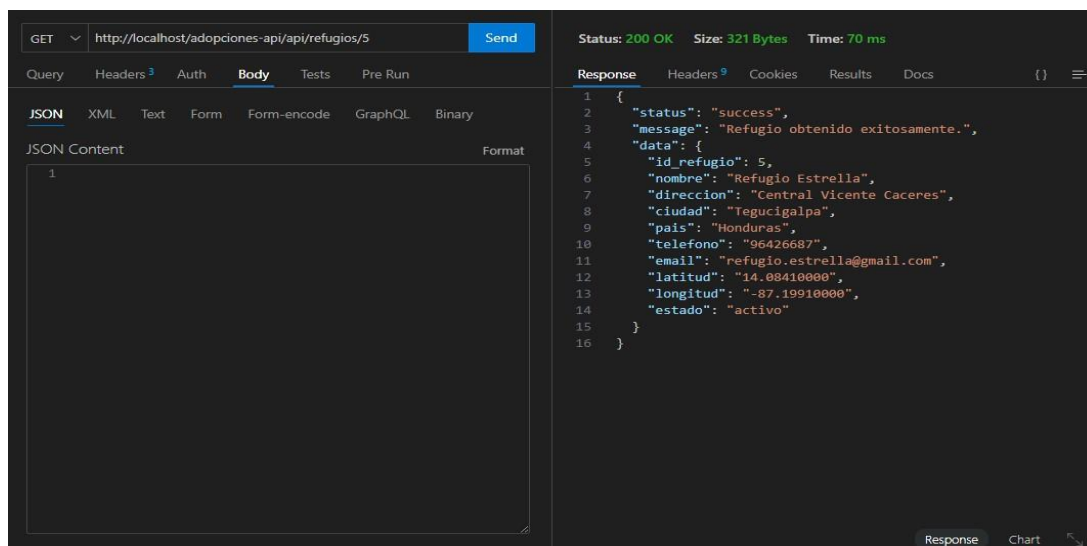


### 2.3. Obtener Refugio por ID

- **Descripción:** Recupera los detalles de un refugio específico por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/refugios/{id_refugio}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/refugios/1`
- **Headers:** (Generalmente **NO** requiere Authorization para ser público).
  - Authorization: Bearer TU\_TOKEN\_JWT\_AQUI (Si tu configuración lo requiere, cualquier usuario logueado puede acceder).

Cuerpo de json: NINGUNO

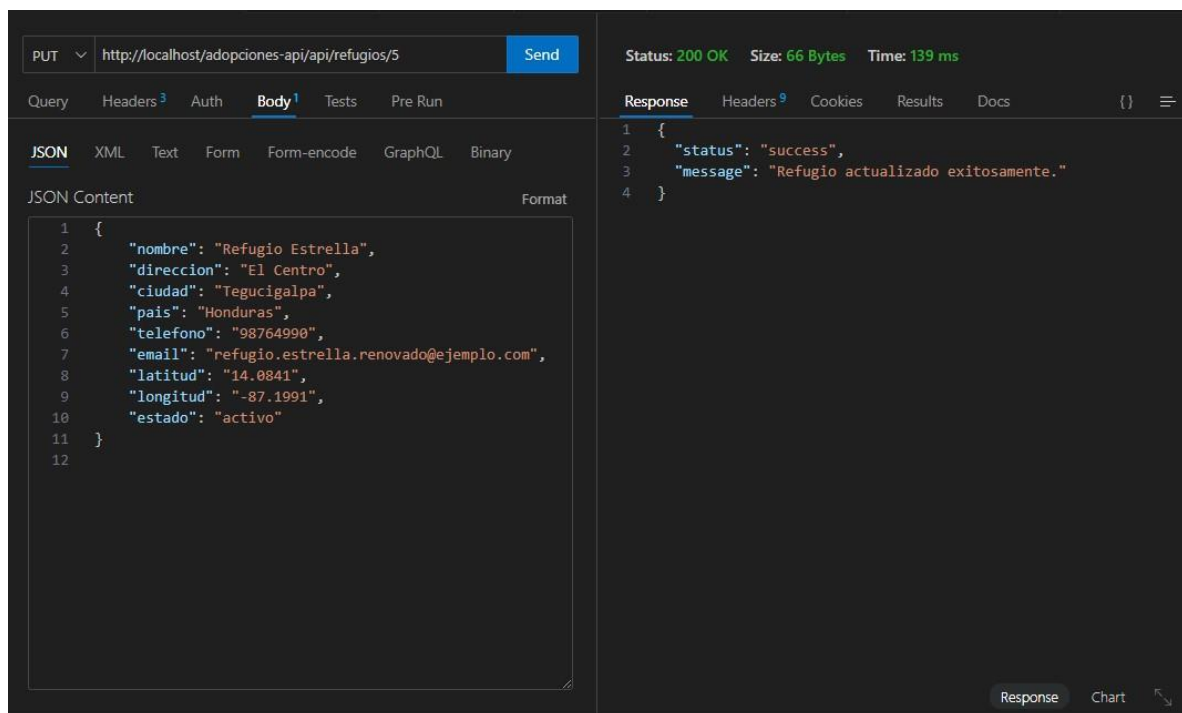
Foto del funcionamiento:



## 2.4. Update Refugio (Actualizar Refugio)

- **Descripción:** Actualiza los detalles de un refugio existente.
- **Método:** PUT (o POST a /id si tu Apache lo reescribe así y tu api.php tiene un parche para ello).
- **URL:** `http://localhost/adopciones-api/api/refugios/(id_refugio)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/refugios/1`
- **Headers:**
  - Content-Type: application/json
  - Authorization: Bearer (Tiene Que Ser Un Token Administrador)

Fotos del funcionamiento:



Antes:

<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	5	Refugio Estrella	Central Vicente Caceres	Tegucigalpa	Honduras	96426687	refugio.estrella@gmail.com	14.08410000	-87.19910000	activo
--------------------------	------------------------	------------------------	------------------------	---	------------------	-------------------------	-------------	----------	----------	----------------------------	-------------	--------------	--------

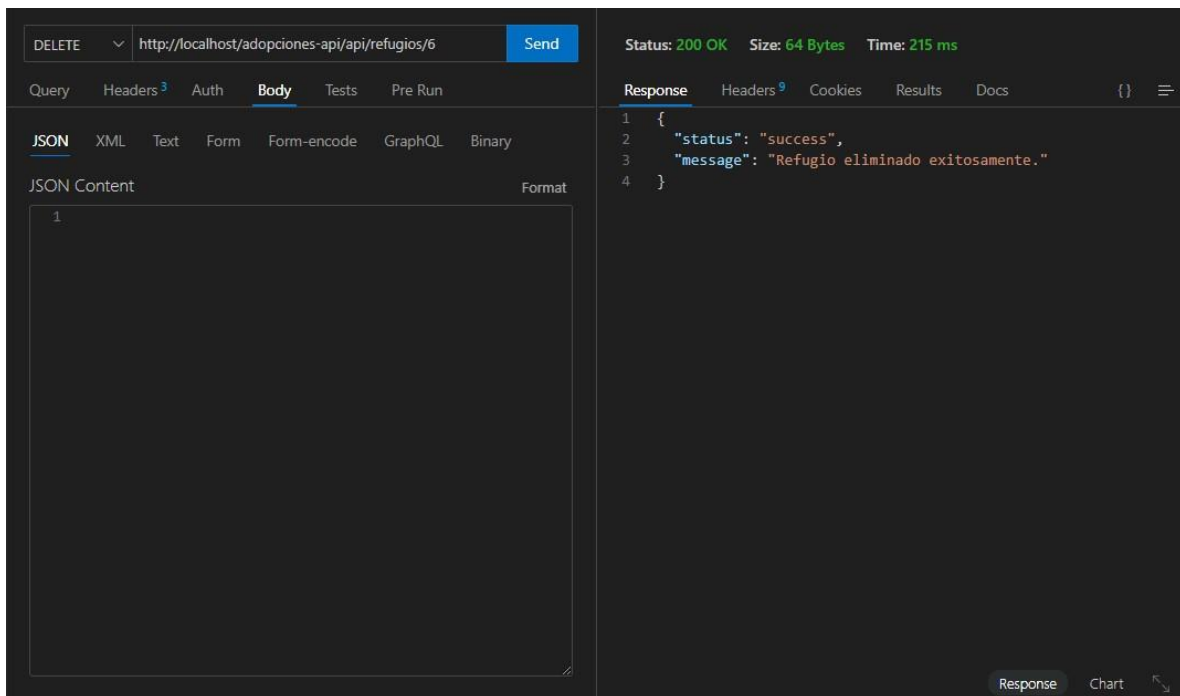
Despues:

<input type="checkbox"/>	<a href="#">Editar</a>	<a href="#">Copiar</a>	<a href="#">Borrar</a>	5	Refugio Estrella	El Centro	Tegucigalpa	Honduras	98764990	refugio.estrella.renovado@ejemplo.com	14.08410000	-87.19910000	ac
--------------------------	------------------------	------------------------	------------------------	---	------------------	-----------	-------------	----------	----------	---------------------------------------	-------------	--------------	----

## 2.5. Delete Refugio (Eliminar Refugio)

- **Descripción:** Elimina un refugio del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/refugios/(id_refugio)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/refugios/2`
- **Headers:**
  - Authorization: Bearer (Tiene Que Ser Un Token Administrador)

Foto del funcionamiento:



## 3.Actividades

**Base URL:** `http://localhost/adopciones-api/api/actividades`

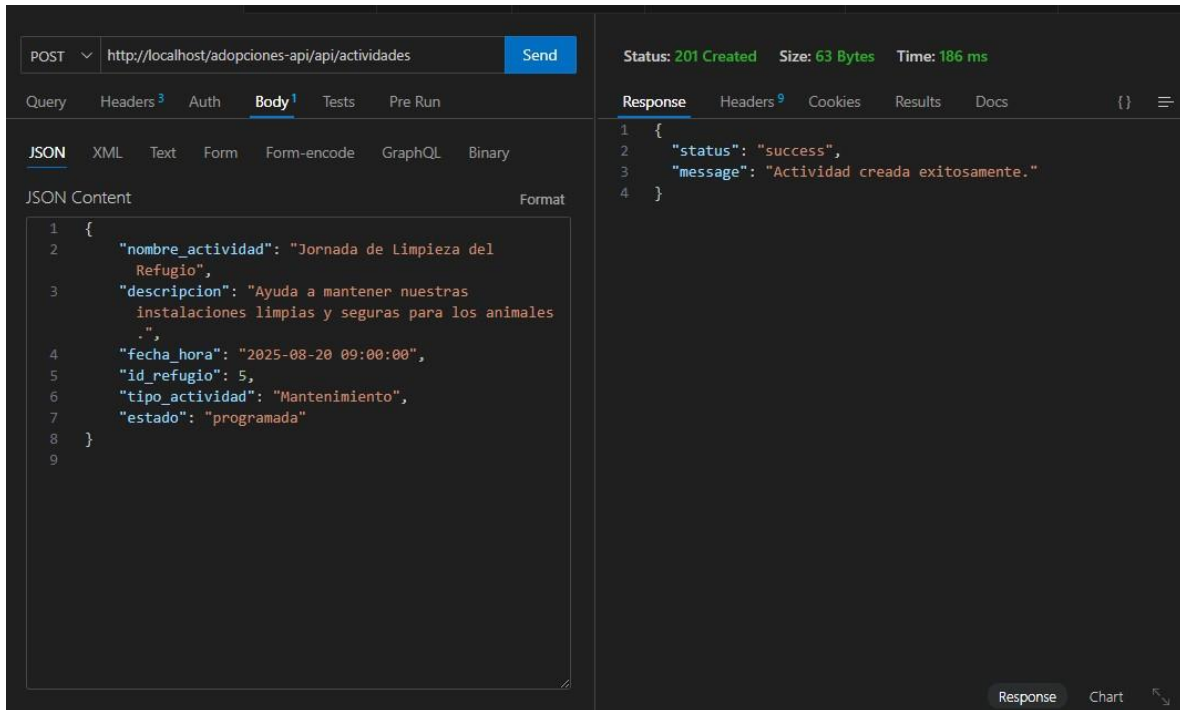
### 3.1. Crear Actividad

- **Descripción:** Crea un nuevo registro de actividad en el sistema.
- **Método:** POST
- **URL:** `http://localhost/adopciones-api/api/actividades`
- **Encabezados (Headers):**



- Content-Type: application/json
- Authorization: Bearer (Tiene que ser un Token Administrador)

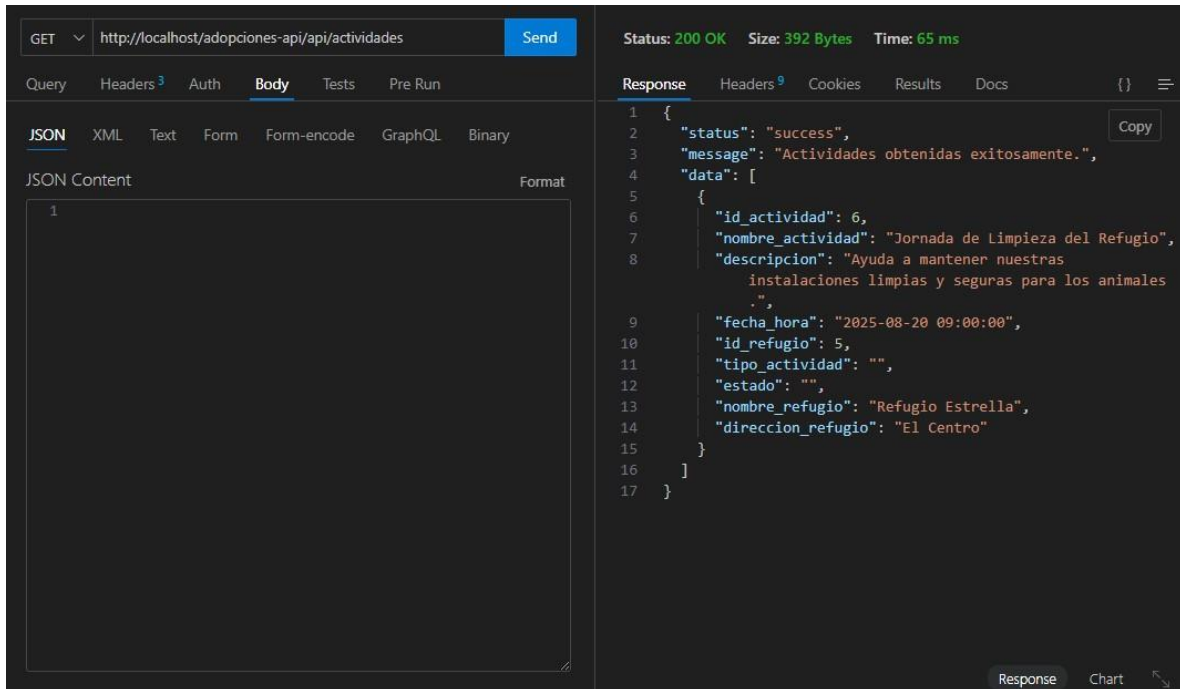
Foto del funcionamiento:



### 3.2. Obtener Todas las Actividades (GET /api/actividades)

- **Descripción:** Recupera una lista de todas las actividades registradas.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/actividades`
- **Encabezados (Headers):**
  - Authorization: Bearer (Tu Token)

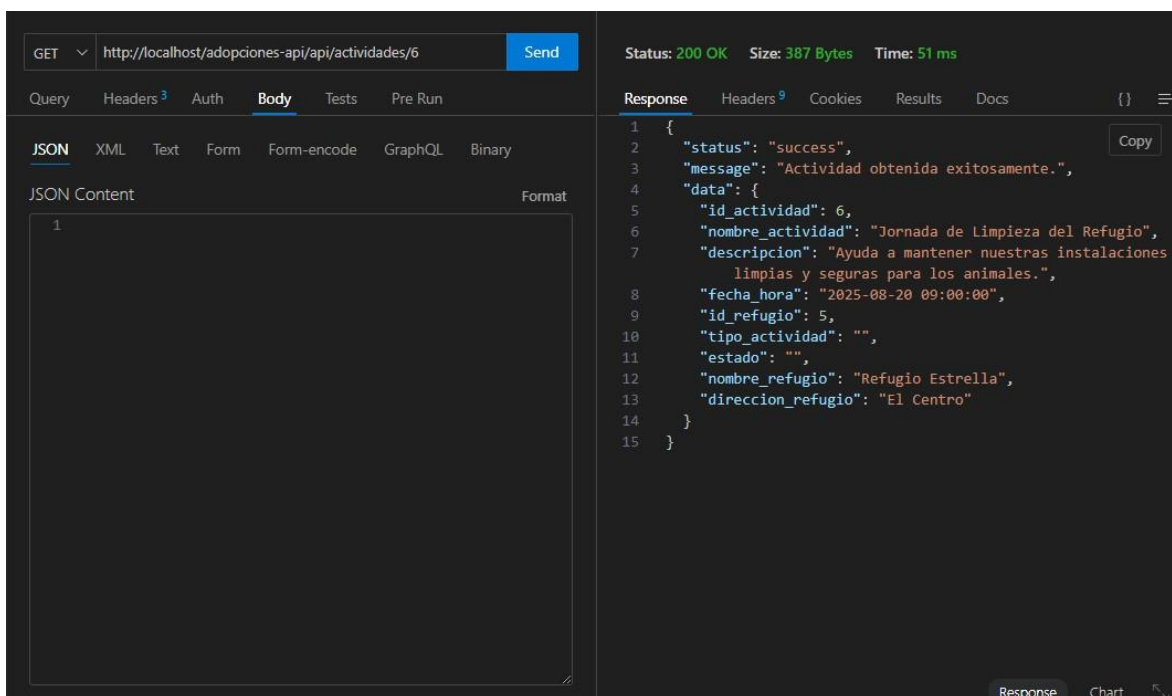
Foto del funcionamiento:



### 3.3. Obtener Actividad por ID

- **Descripción:** Recupera los detalles de una actividad específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/actividades/(id_actividad)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/actividades/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Tu Token)

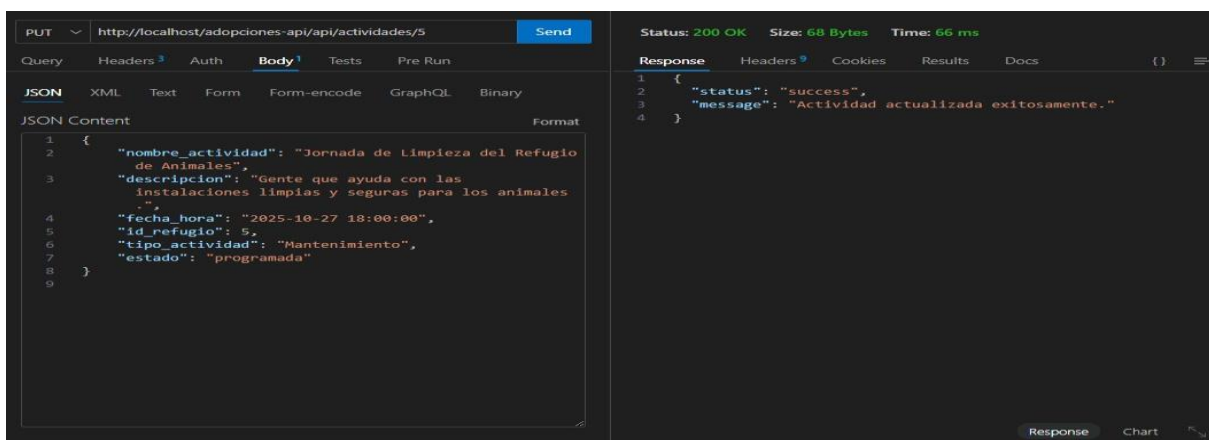
Foto del funcionamiento:



### 3.4. Actualizar Actividad

- **Descripción:** Actualiza los detalles de una actividad existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/actividades/(id_actividad)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/actividades/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

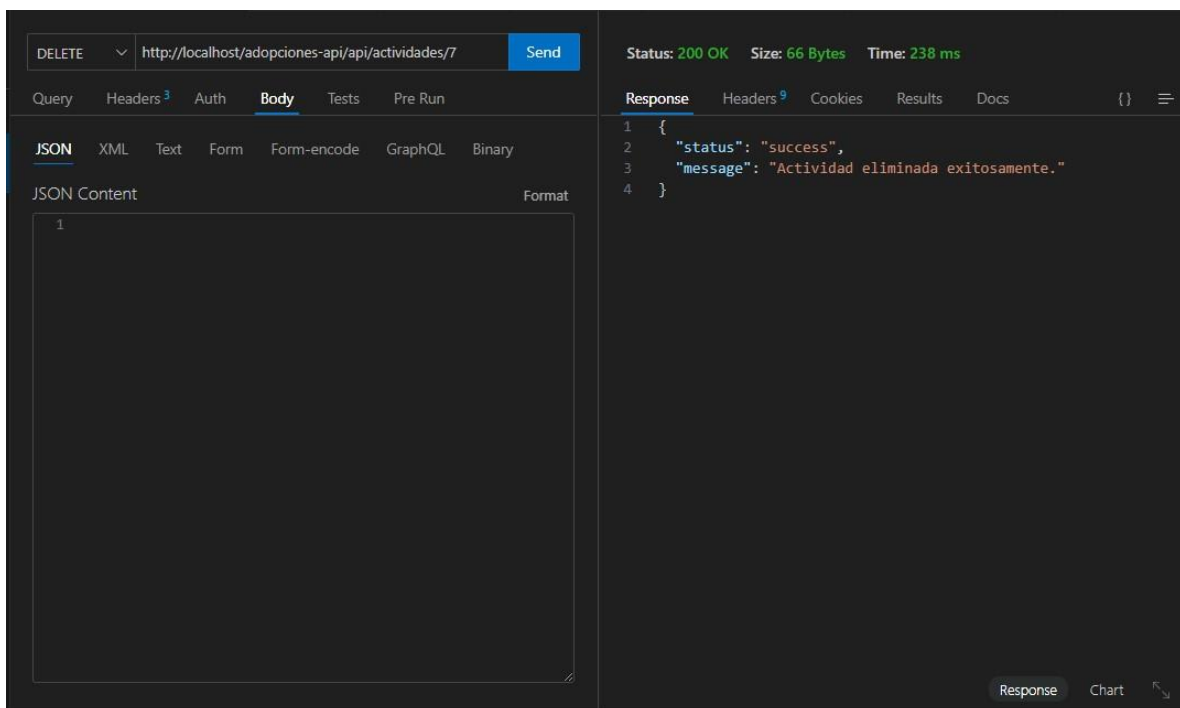
Foto del funcionamiento:



### 3.5. Eliminar Actividad

- **Descripción:** Elimina una actividad del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/actividades/(id_actividad)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/actividades/2`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



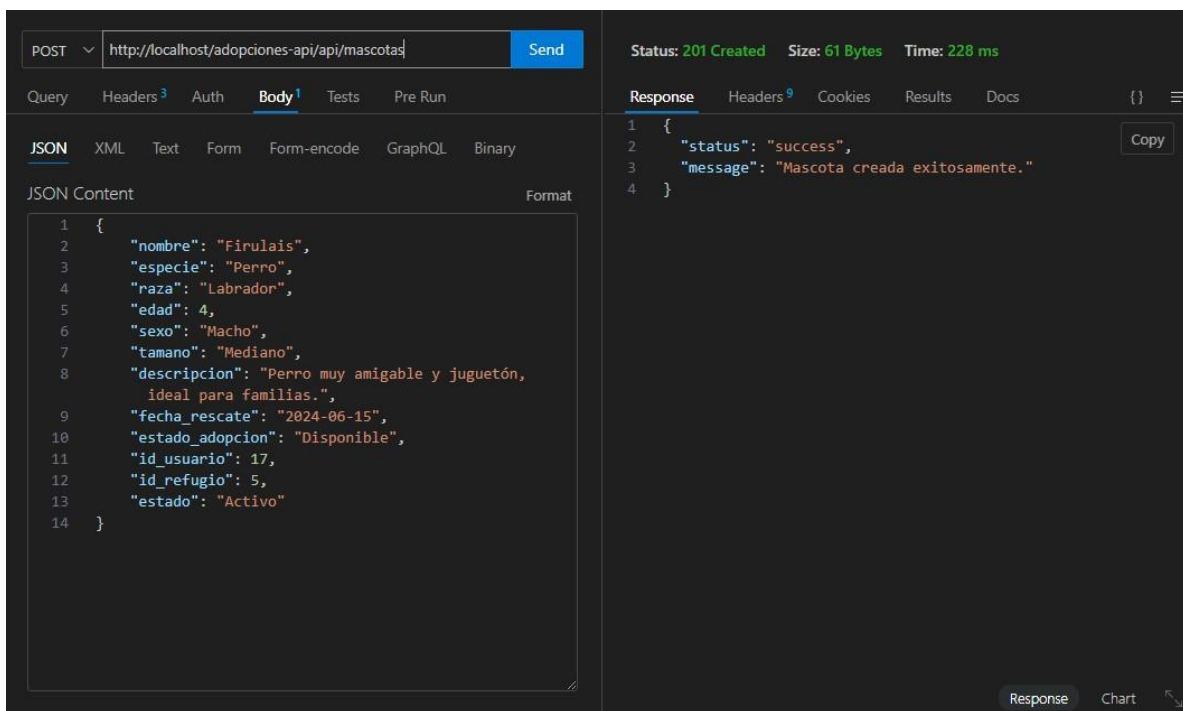
## 4.Mascotas

Base URL: <http://localhost/adopciones-api/api/mascotas>

### 4.1. Crear Mascota

- **Descripción:** Crea un nuevo registro de mascota en el sistema.
- **Método:** POST
- **URL:** <http://localhost/adopciones-api/api/mascotas>
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

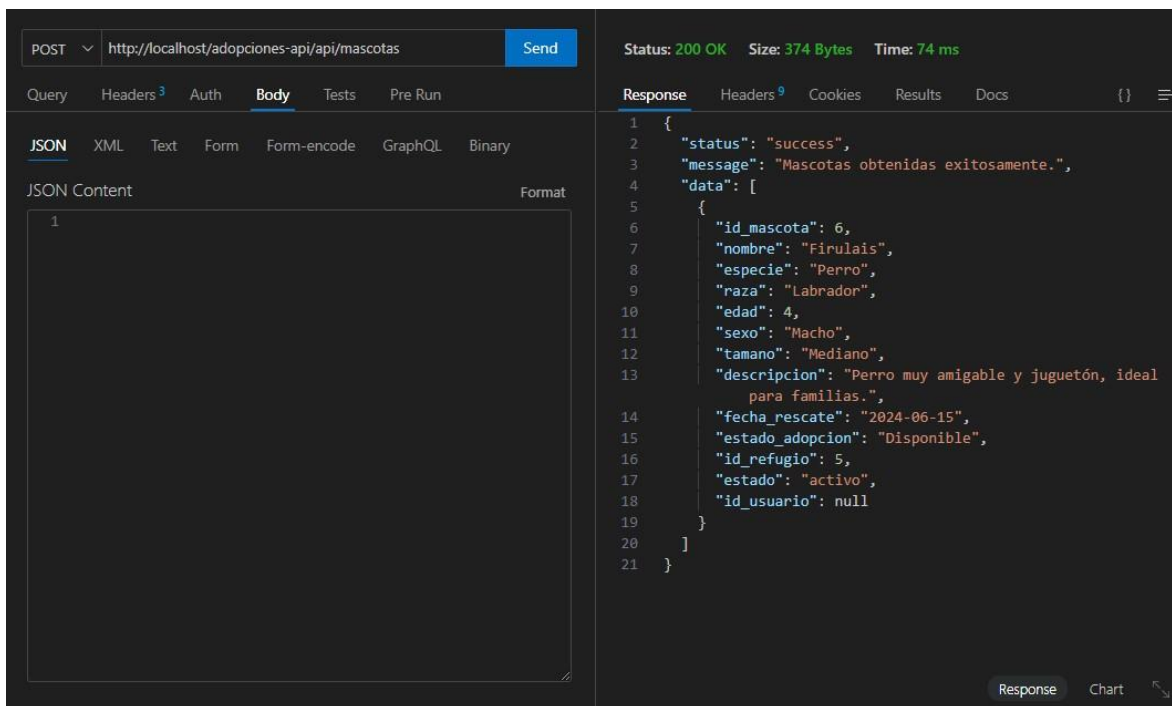
Foto de su funcionamiento:



### 4.2. Obtener Todas las Mascotas

- **Descripción:** Recupera una lista de todas las mascotas registradas.
- **Método:** GET
- **URL:** <http://localhost/adopciones-api/api/mascotas>
- **Encabezados (Headers):**
  - Authorization: Bearer (Tu Token)

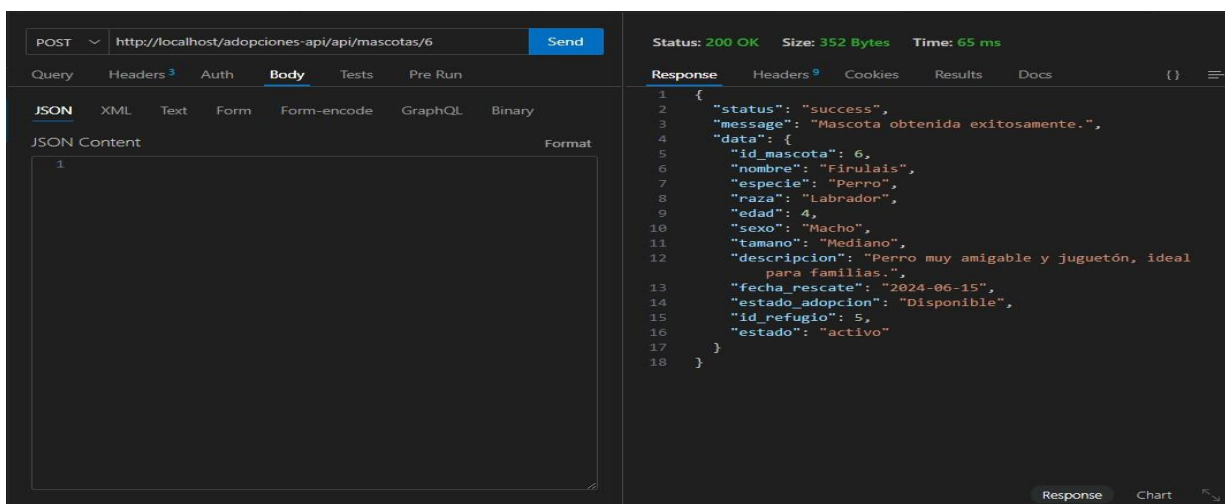
Foto el funcionamiento:



#### 4.3. Obtener Mascota por ID

- **Descripción:** Recupera los detalles de una mascota específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/mascotas/id_mascota`
  - **Ejemplo:** `http://localhost/adopciones-api/api/mascotas/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Tu Token)

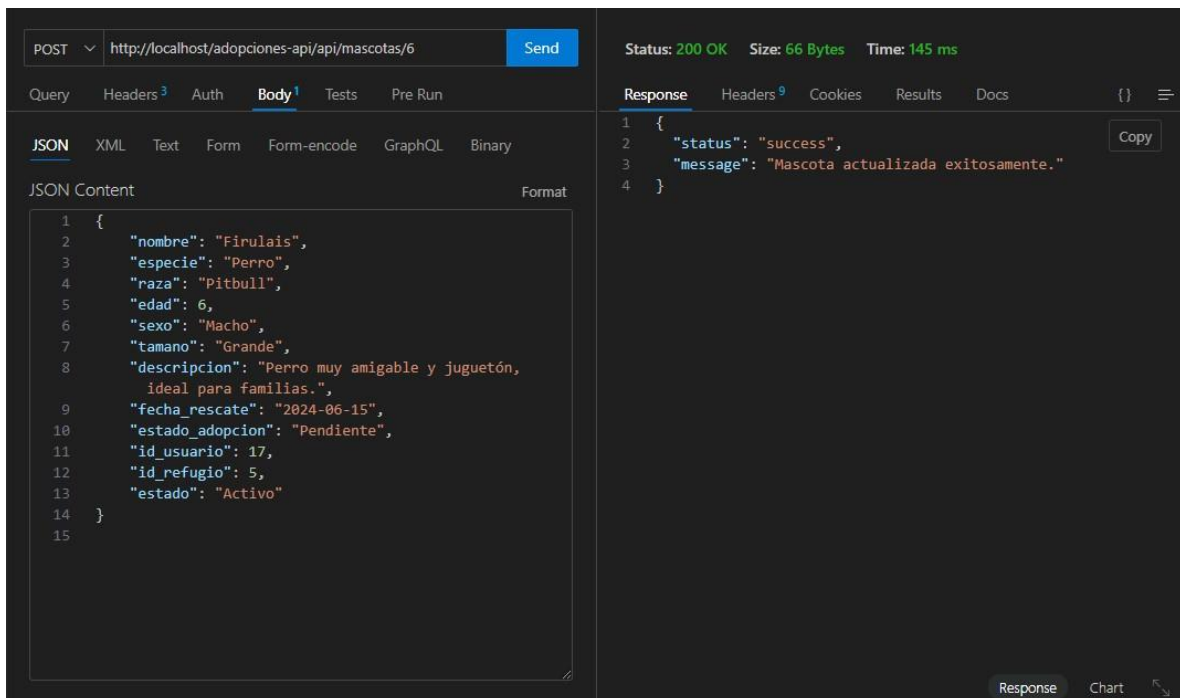
Foto del funcionamiento:



#### 4.4. Actualizar Mascota

- **Descripción:** Actualiza los detalles de una mascota existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/mascotas/(id_mascota)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/mascotas/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

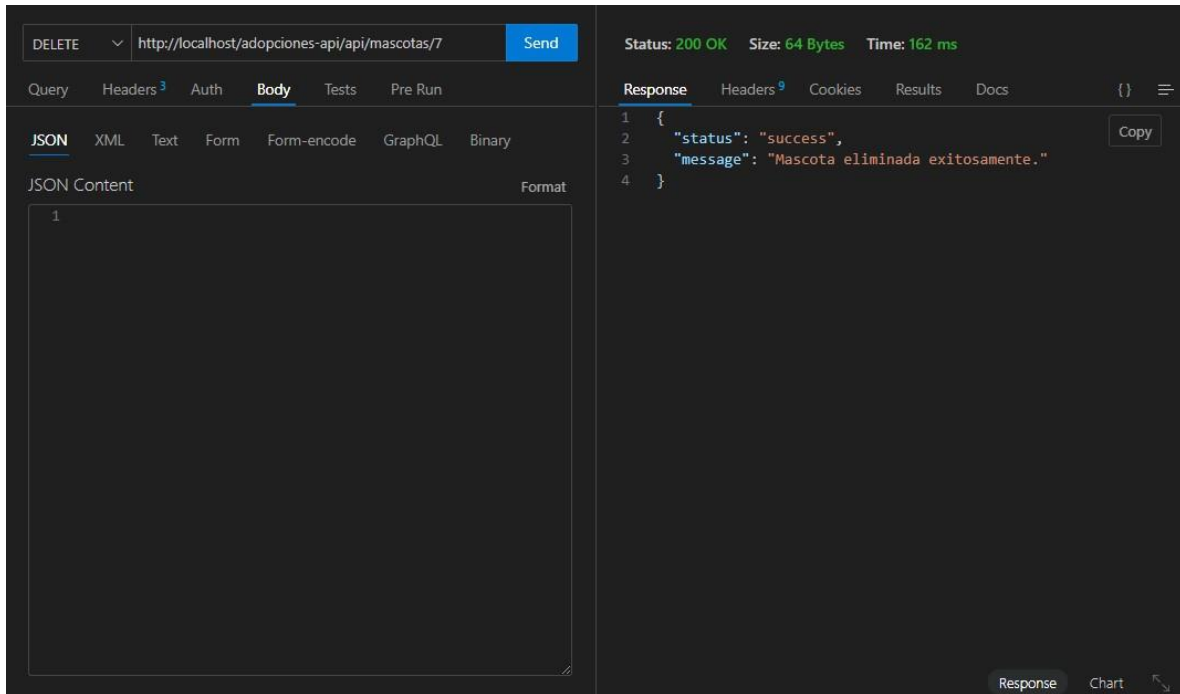
Foto del funcionamiento:



#### 4.5. Eliminar Mascota

- **Descripción:** Elimina una mascota del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/mascotas/(id_mascota)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/mascotas/2`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



## 5.Solicitudes de Adopción

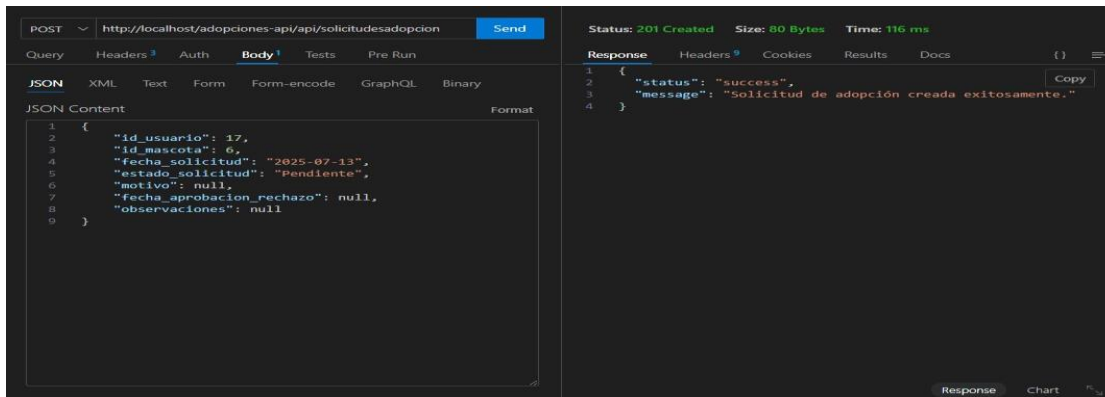
**Base URL:** http://localhost/adopciones-api/api/solicitudesadopcion

### 5.1. Crear Solicitud de Adopción

- **Descripción:** Permite a un usuario registrar una nueva solicitud para adoptar una mascota.
- **Método:** POST
- **URL:** http://localhost/adopciones-api/api/solicitudesadopcion
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Tu Token)



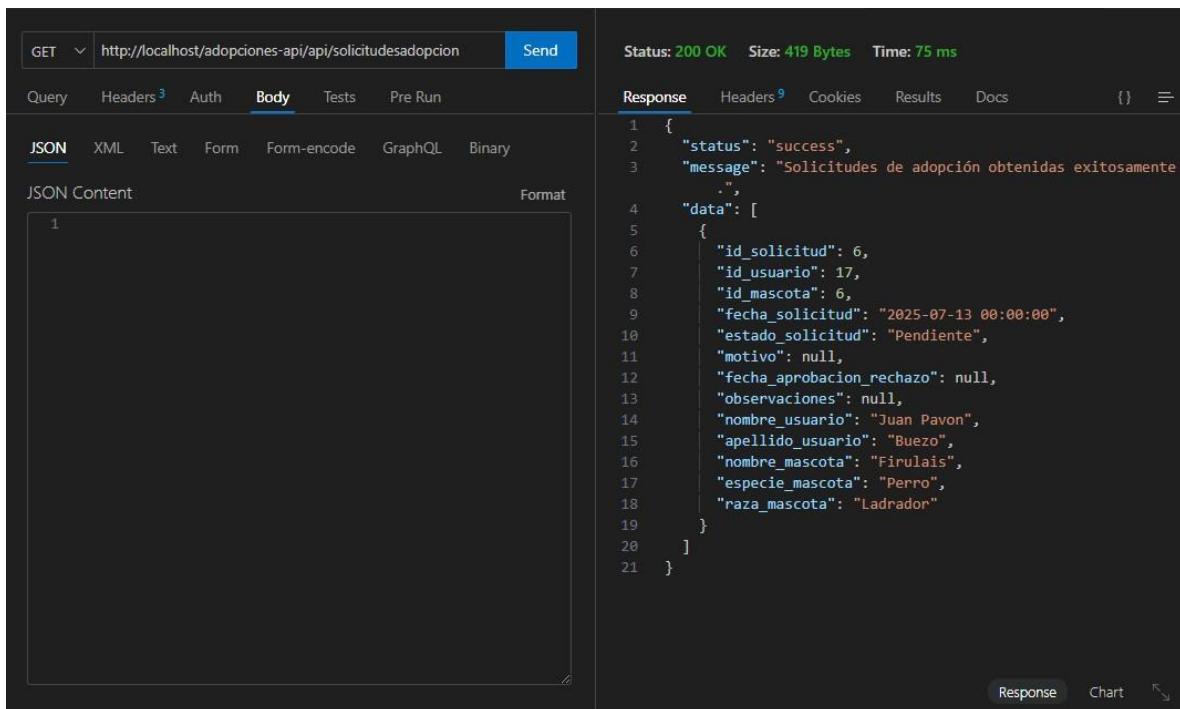
Foto del funcionamiento:



## 5.2. Obtener Todas las Solicitudes de Adopción

- **Descripción:** Recupera una lista de todas las solicitudes de adopción.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/solicitudesadopcion`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

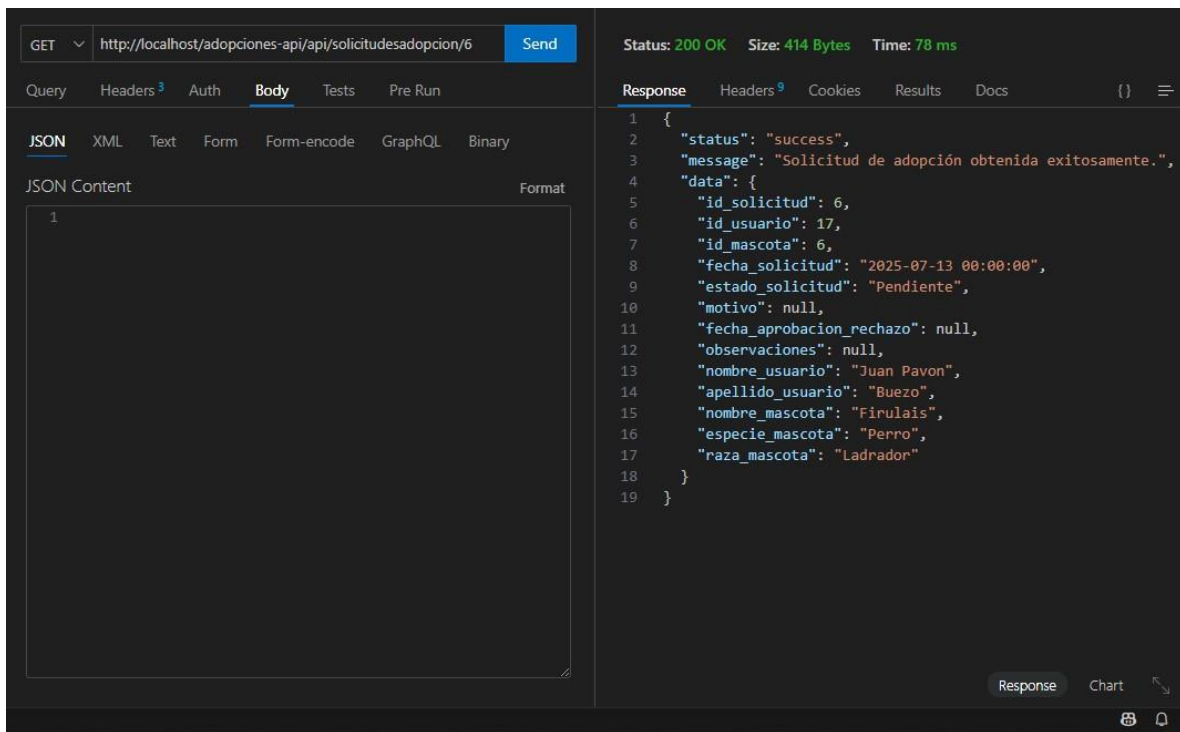
Foto del funcionamiento:



### 5.3. Obtener Solicitud por ID

- **Descripción:** Recupera los detalles de una solicitud específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/solicitudesadopcion/{id_solicitud}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/solicitudesadopcion/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Tu Token Usuario O Administrador)

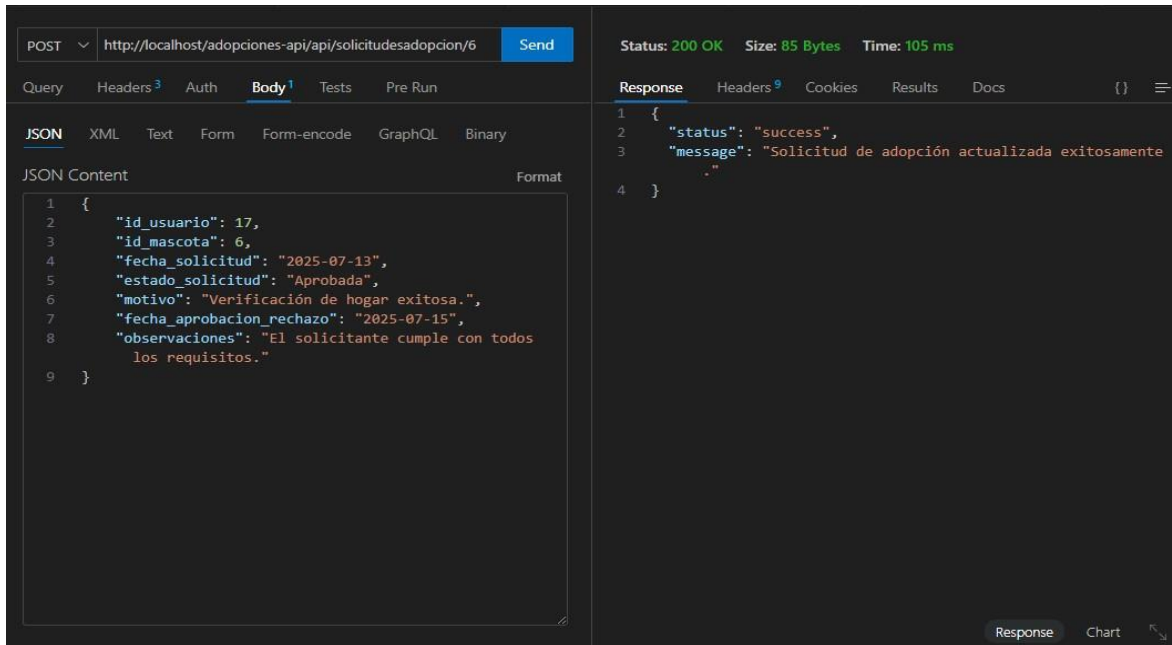
Foto del funcionamiento:



### 5.4. Actualizar Solicitud de Adopción (PUT /api/solicitudes-adopcion/{id\_solicitud})

- **Descripción:** Actualiza los detalles de una solicitud de adopción existente. Esto es usado comúnmente por administradores para cambiar el estado de la solicitud y añadir el motivo/observaciones.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/solicitudesadopcion/{id_solicitud}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/solicitudesadopcion/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:

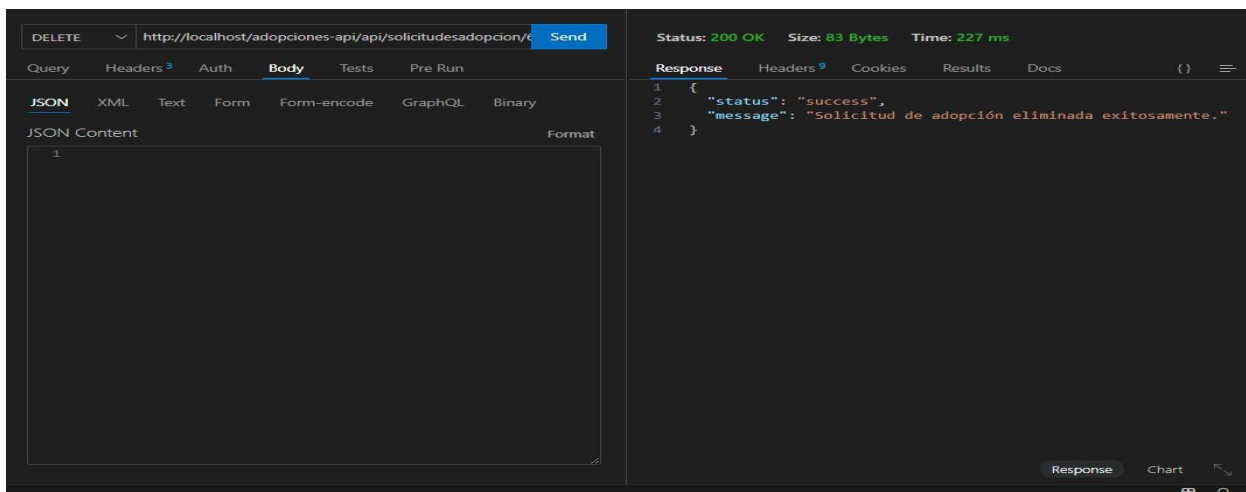


## 5.5. Eliminar Solicitud de Adopción

- **Descripción:** Elimina una solicitud de adopción del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/solicitudes-adopcion/(id_solicitud)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/solicitudes-adopcion/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Cuerpo de JSON: NINGUNO

Foto del funcionamiento:



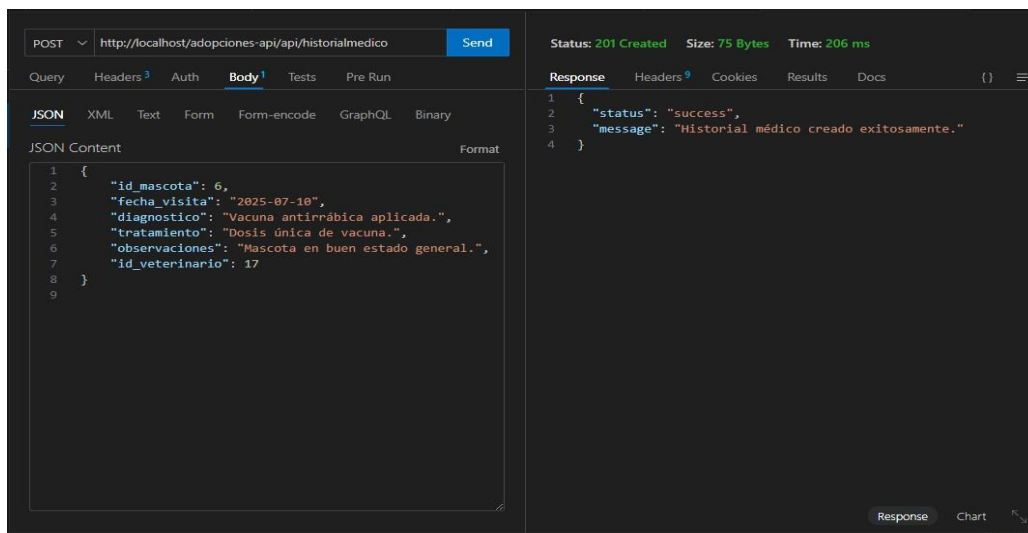
## 6. Historial Médico

**Base URL:** <http://localhost/adopciones-api/api/historialmedico>

### 6.1. Crear Registro de Historial Médico

- **Descripción:** Crea un nuevo registro en el historial médico de una mascota específica.
- **Método:** POST
- **URL:** <http://localhost/adopciones-api/api/historialmedico>
- **Encabezados (Headers):**
  - Content-Type:
  - application/json Authorization: Bearer (Token Administrador)

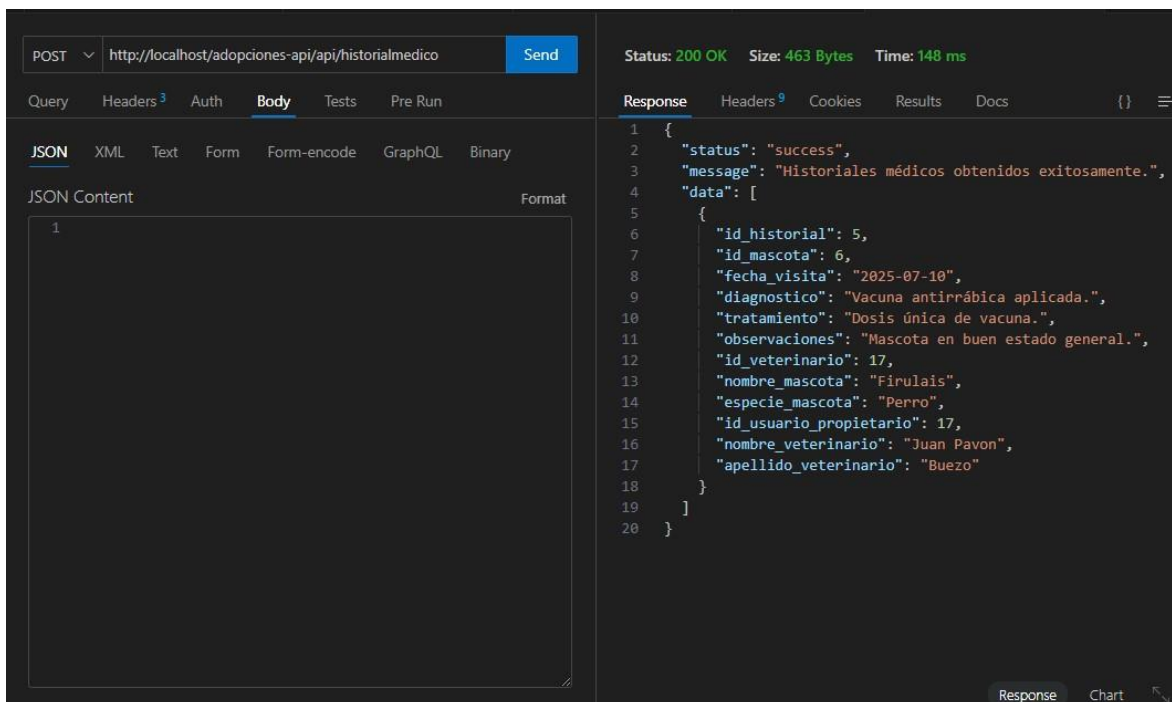
Foto del funcionamiento:



### 6.2. Obtener Todo el Historial Médico

- **Descripción:** Recupera una lista de todos los registros de historial médico de todas las mascotas.
- **Método:** GET
- **URL:** <http://localhost/adopciones-api/api/historialmedico>
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

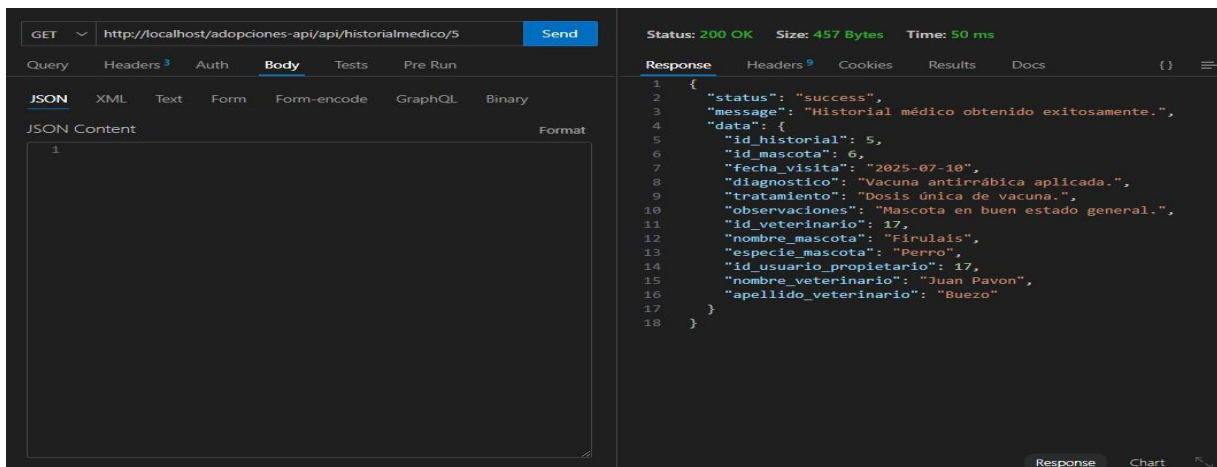
Foto del funcionamiento:



### 6.3. Obtener Historial Médico por ID

- **Descripción:** Recupera los detalles de un registro de historial médico específico por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/historialmedico/(id_historial)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/historial-medico/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

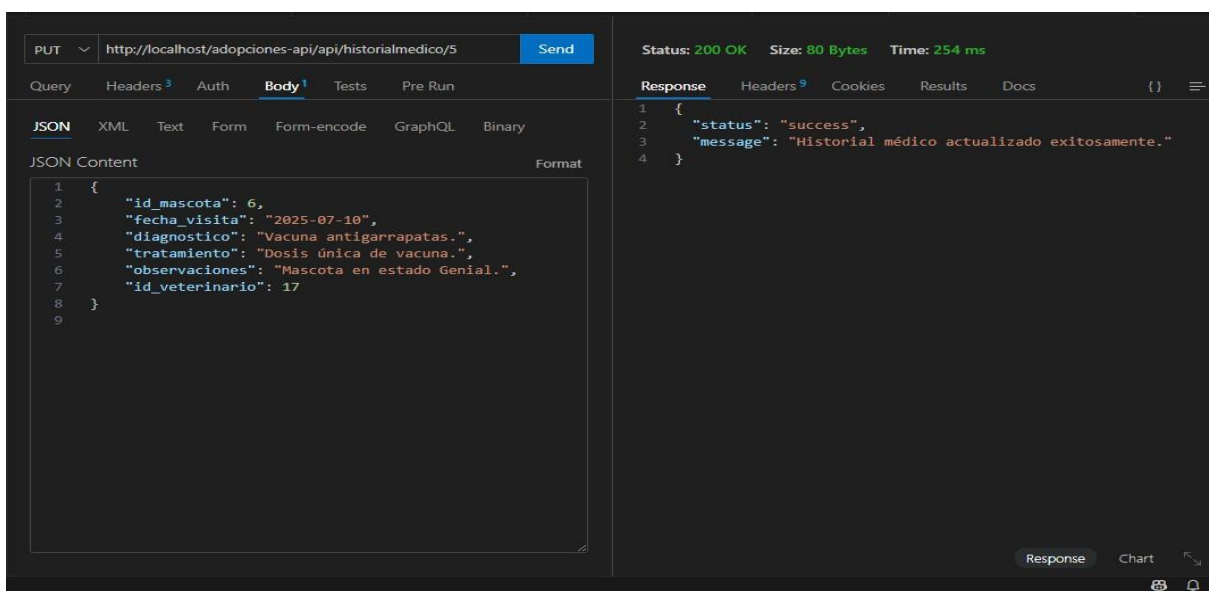
Foto del funcionamiento:



## 6.4. Actualizar Registro de Historial Médico

- **Descripción:** Actualiza los detalles de un registro de historial médico existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/historial-medico/{id_historial}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/historialmedico/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer Token Administrador

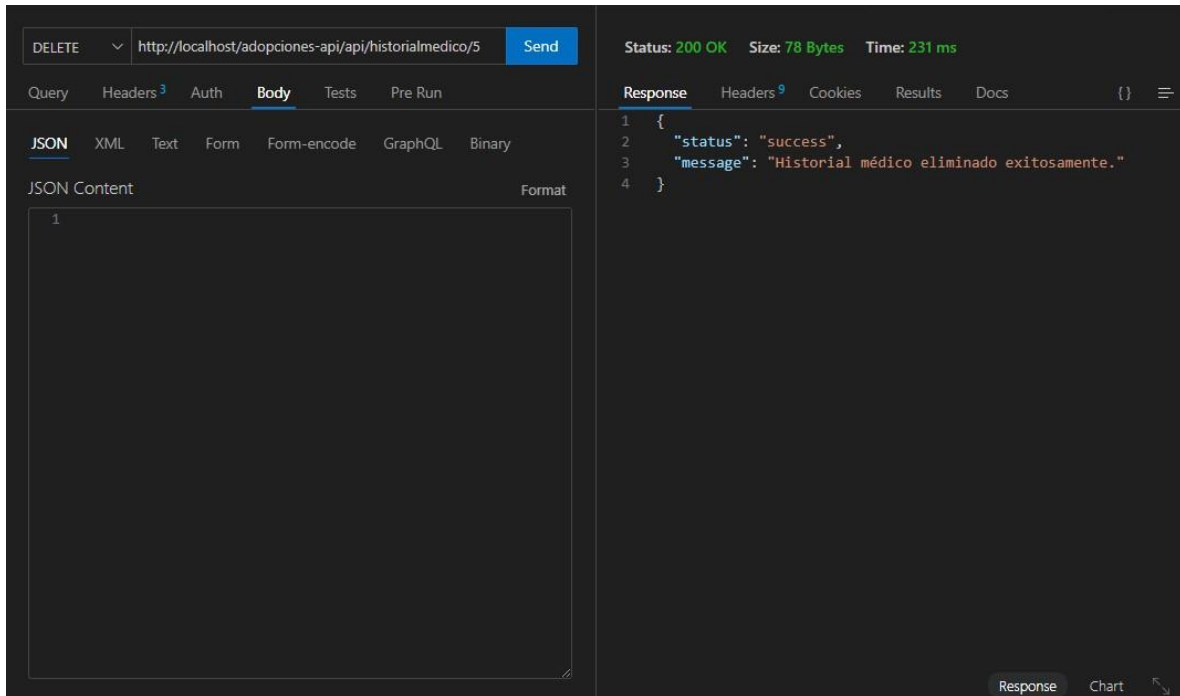
Foto del funcionamiento:



## 6.5. Eliminar Registro de Historial Médico (DELETE /api/historial-medico/{id\_historial})

- **Descripción:** Elimina un registro de historial médico del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/historial-medico/{id_historial}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/historial-medico/2`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



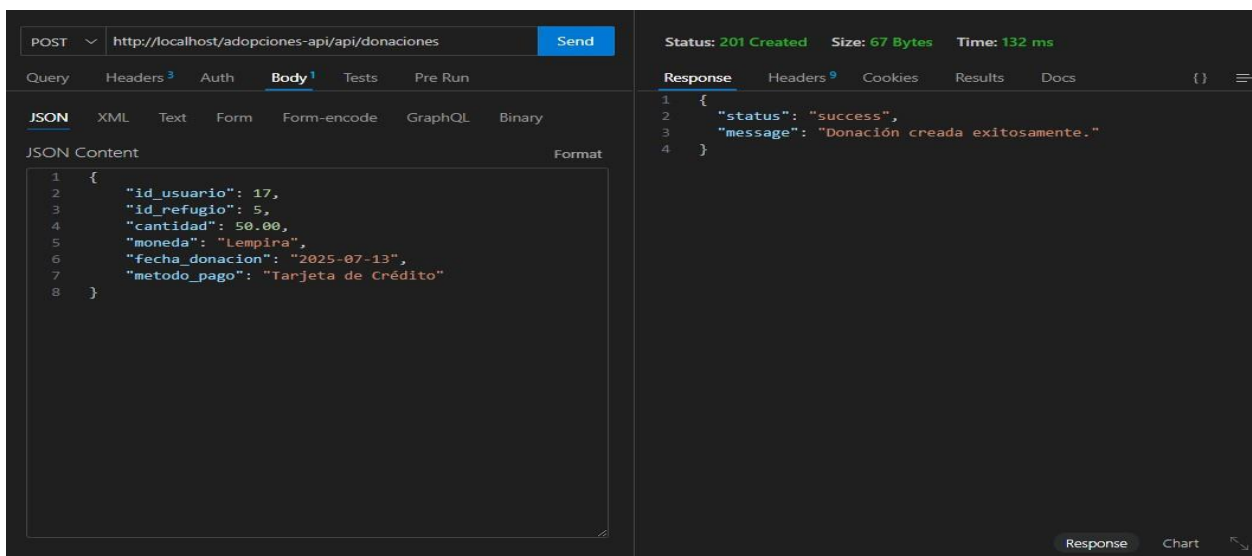
## 7.Donaciones

**Base URL:** http://localhost/adopciones-api/api/donaciones

### 7.1. Crear Donación

- Descripción: Registra una nueva donación en el sistema.
- Método: POST
- URL: http://localhost/adopciones-api/api/donaciones
- Encabezados (Headers):
- Content-Type: application/json
- Authorization: Bearer (Token Administrador)

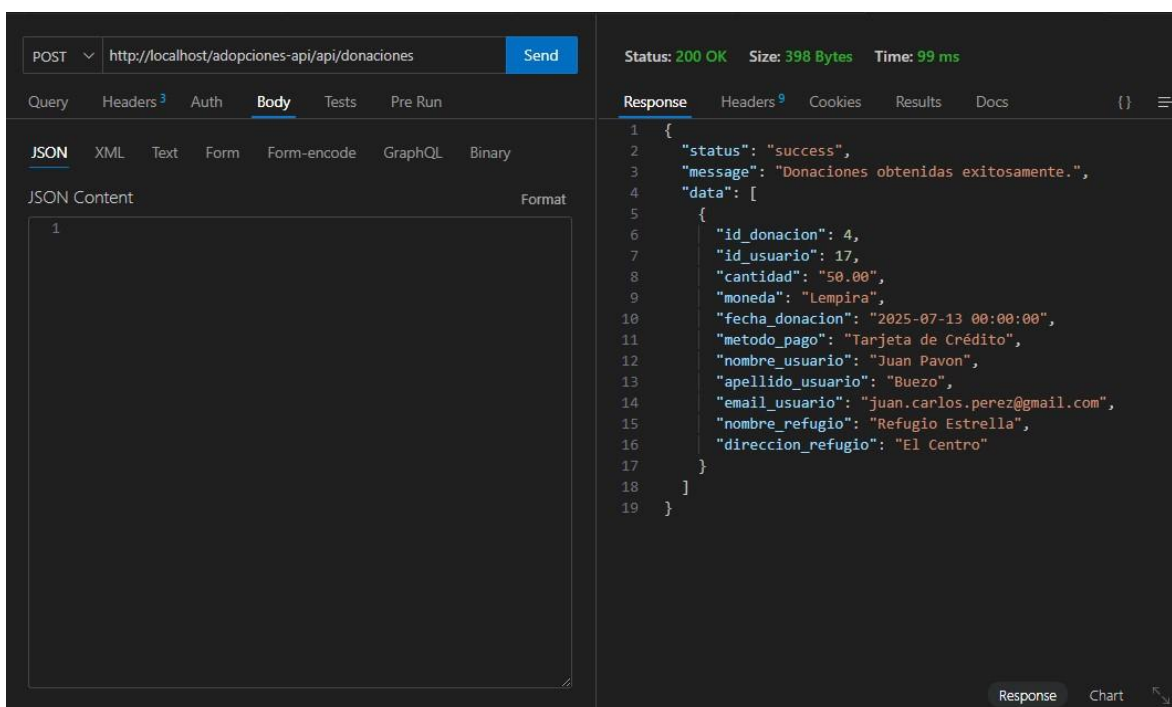
Foto del funcionamiento:



## 7.2. Obtener Todas las Donaciones

- **Descripción:** Recupera una lista de todas las donaciones registradas.
- **Método:** GET
- **URL:** `http://localhost/adoptaciones-api/api/donaciones`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:

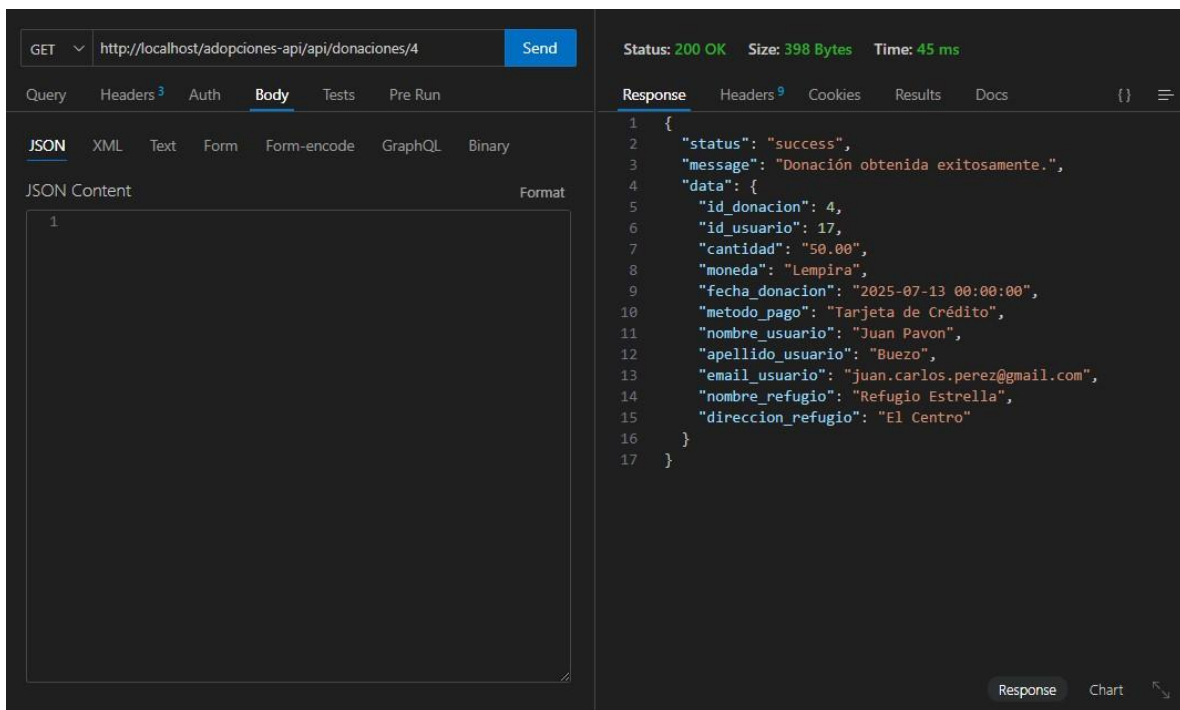




### 7.3. Obtener Donación por ID

- **Descripción:** Recupera los detalles de una donación específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/donaciones/(id_donacion)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/donaciones/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

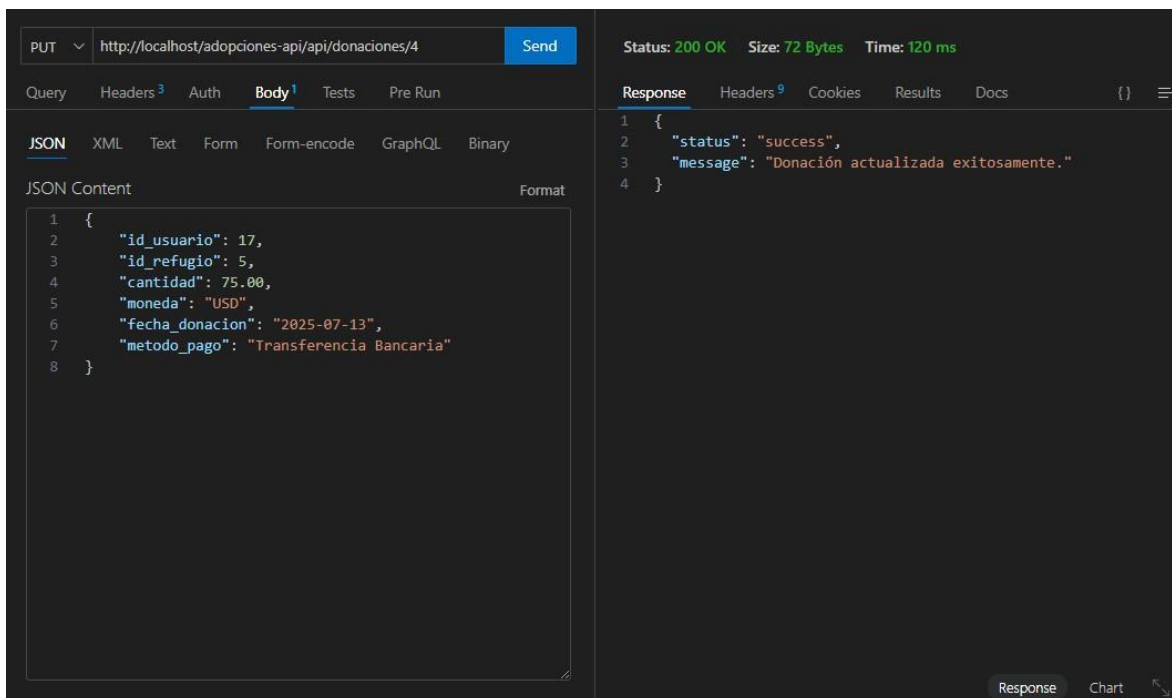
Foto del funcionamiento:



### 7.4. Actualizar Donación

- **Descripción:** Actualiza los detalles de una donación existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/donaciones/(id_donacion)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/donaciones/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

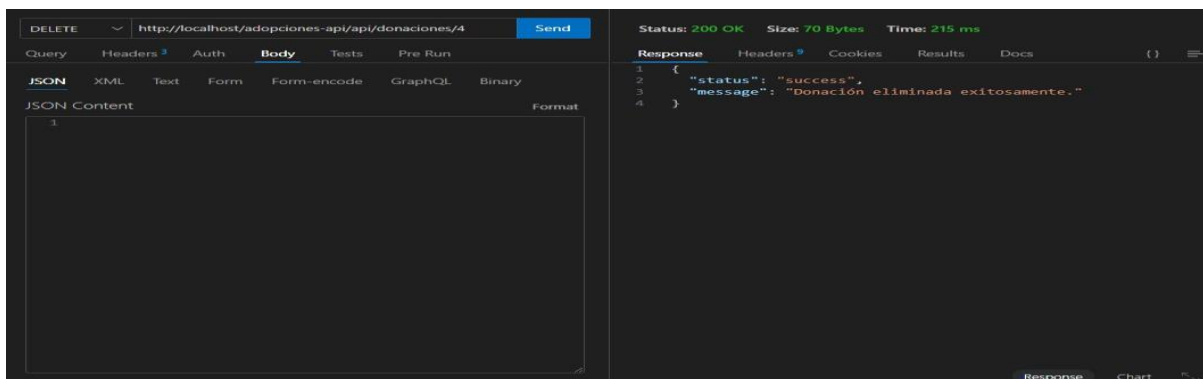
Foto del funcionamiento:



## 7.5. Eliminar Donación

- **Descripción:** Elimina una donación del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adoptaciones-api/api/donaciones/(id_donacion)`
  - **Ejemplo:** `http://localhost/adoptaciones-api/api/donaciones/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



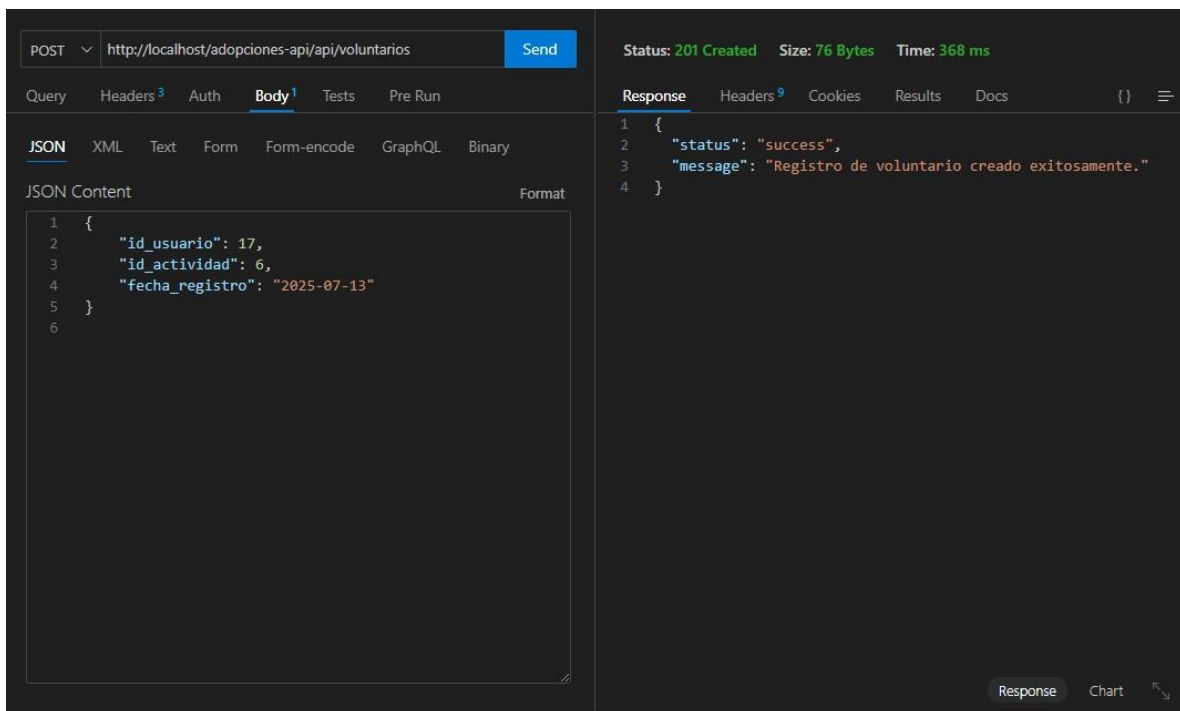
## 8.Voluntarios

**Base URL:** http://localhost/adopciones-api/api/voluntarios

### 8.1. Crear Voluntario

- **Descripción:** Registra a una nueva persona como voluntario en el sistema.
- **Método:** POST
- **URL:** http://localhost/adopciones-api/api/voluntarios
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

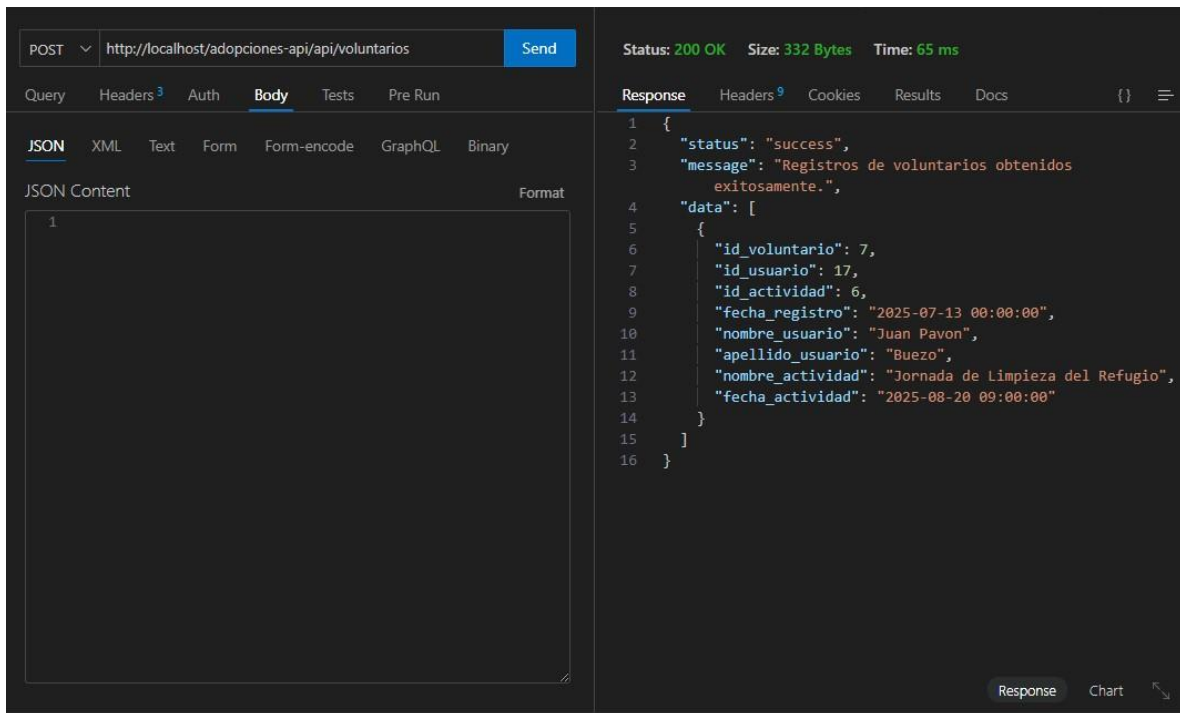
Foto del funcionamiento:



### 8.2. Obtener Todas las Participaciones de Voluntarios

- **Descripción:** Recupera una lista de todas las participaciones de voluntarios registradas.
- **Método:** GET
- **URL:** http://localhost/adopciones-api/api/voluntarios
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

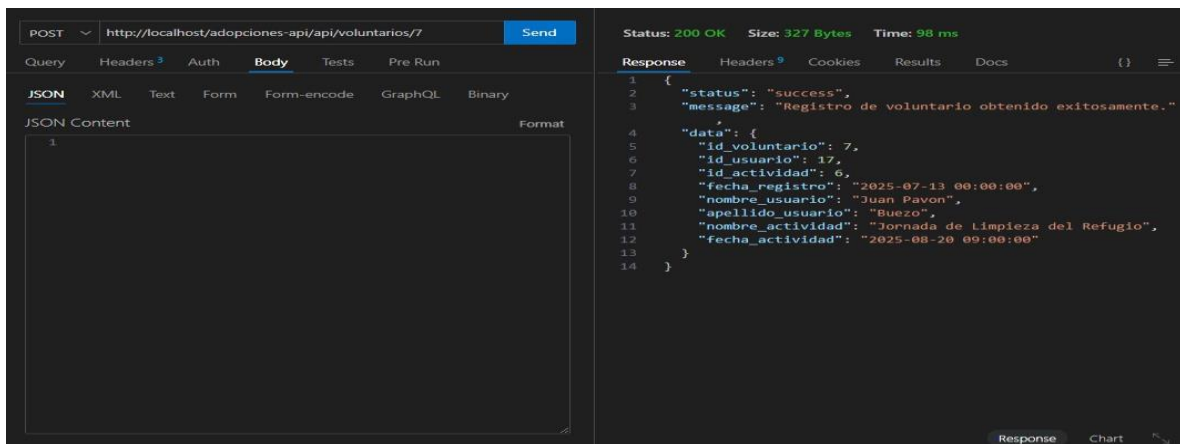
Foto del funcionamiento:



### 8.3. Obtener Participación de Voluntario por ID

- **Descripción:** Recupera los detalles de una participación de voluntario específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adoptaciones-api/api/voluntarios/(id_voluntario)`
  - **Ejemplo:** `http://localhost/adoptaciones-api/api/voluntarios/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:

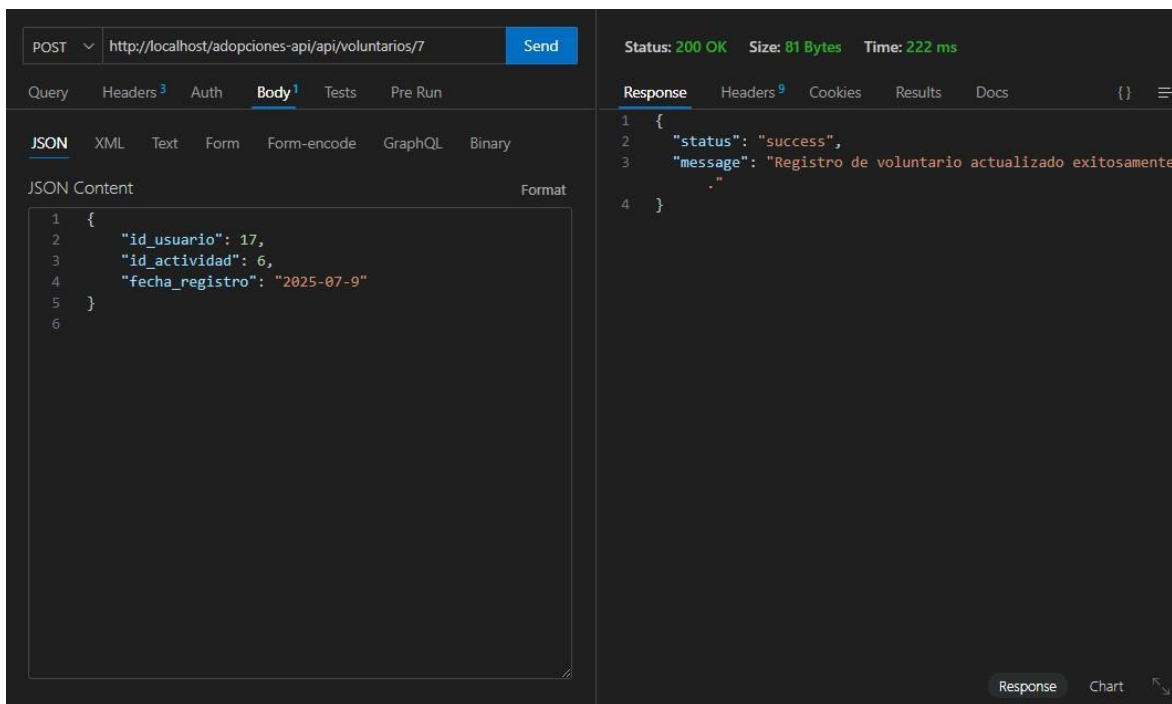


## 8.4. Actualizar Participación de Voluntario

- **Descripción:** Actualiza los detalles de una participación de voluntario existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/voluntarios/(id_voluntario)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/voluntarios/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

Cuerpo de JSON:

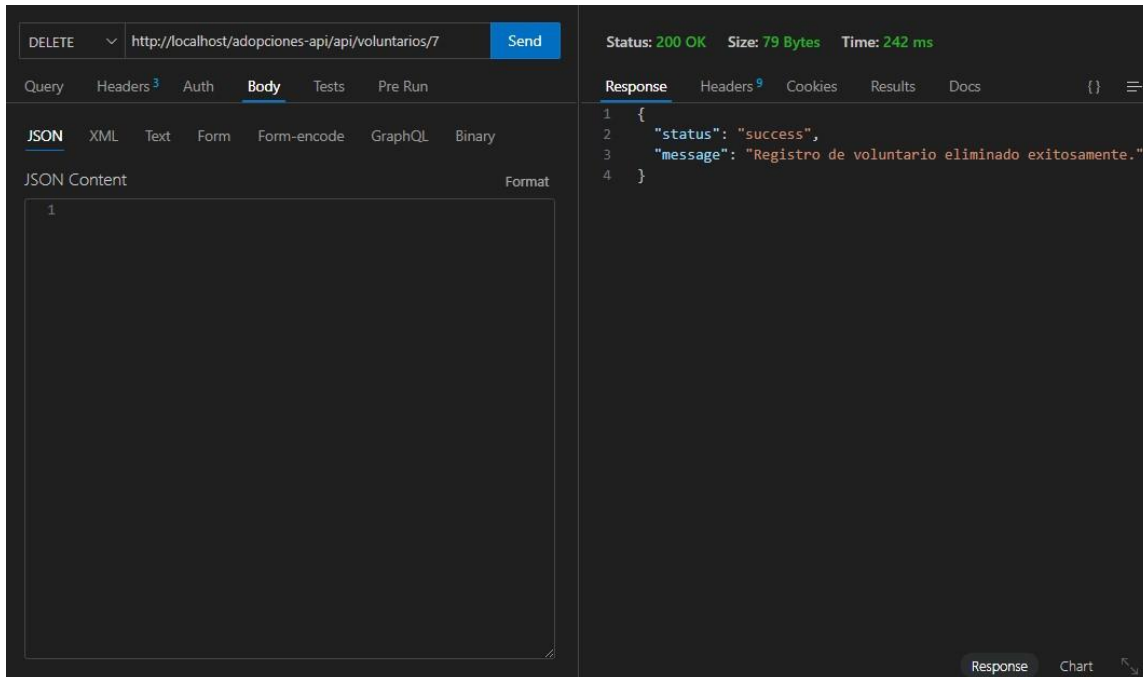
Foto del funcionamiento:



## 8.5. Eliminar Participación de Voluntario

- **Descripción:** Elimina un registro de participación de voluntario del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/voluntarios/(id_voluntario)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/voluntarios/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



## 9. Galería Multimedia

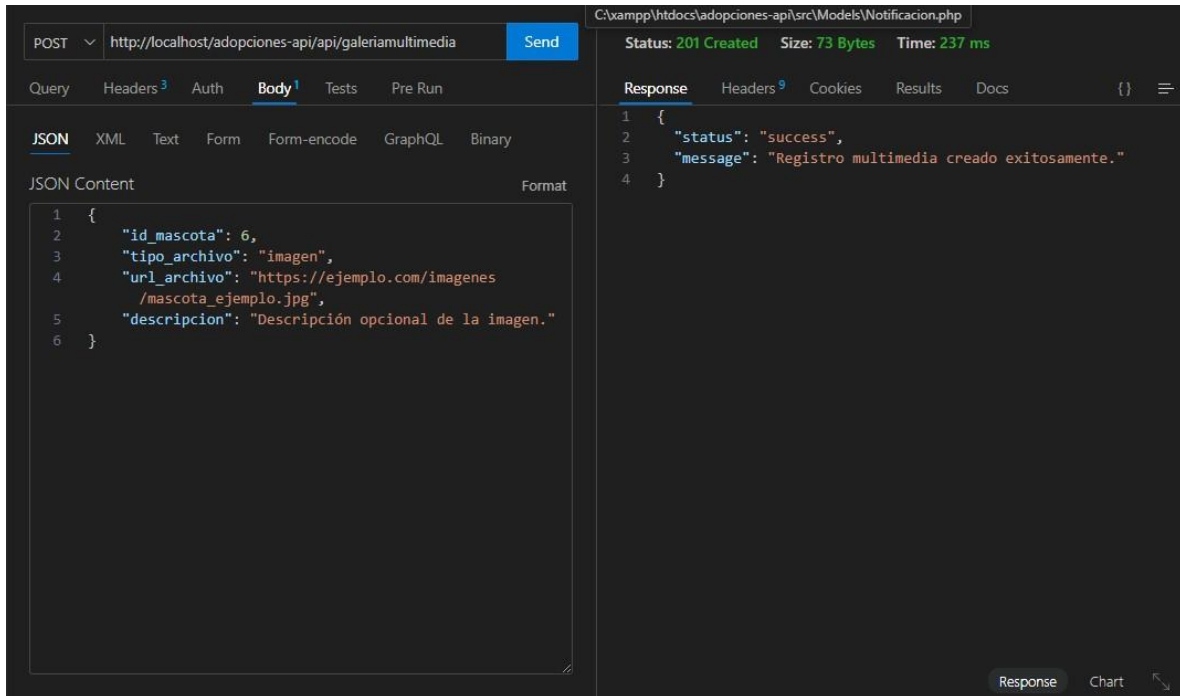
Este controlador (GaleriaMultimediaController) gestiona la asociación de imágenes y otros archivos multimedia a las mascotas.

**Base URL:** <http://localhost/adopciones-api/api/galeriamultimedia>

### 9.1. Añadir Archivo Multimedia

- **Descripción:** Añade un nuevo archivo multimedia (ej. una imagen) a una mascota específica.
- **Método:** POST
- **URL:** <http://localhost/adopciones-api/api/galeriamultimedia>
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer Token Administrador

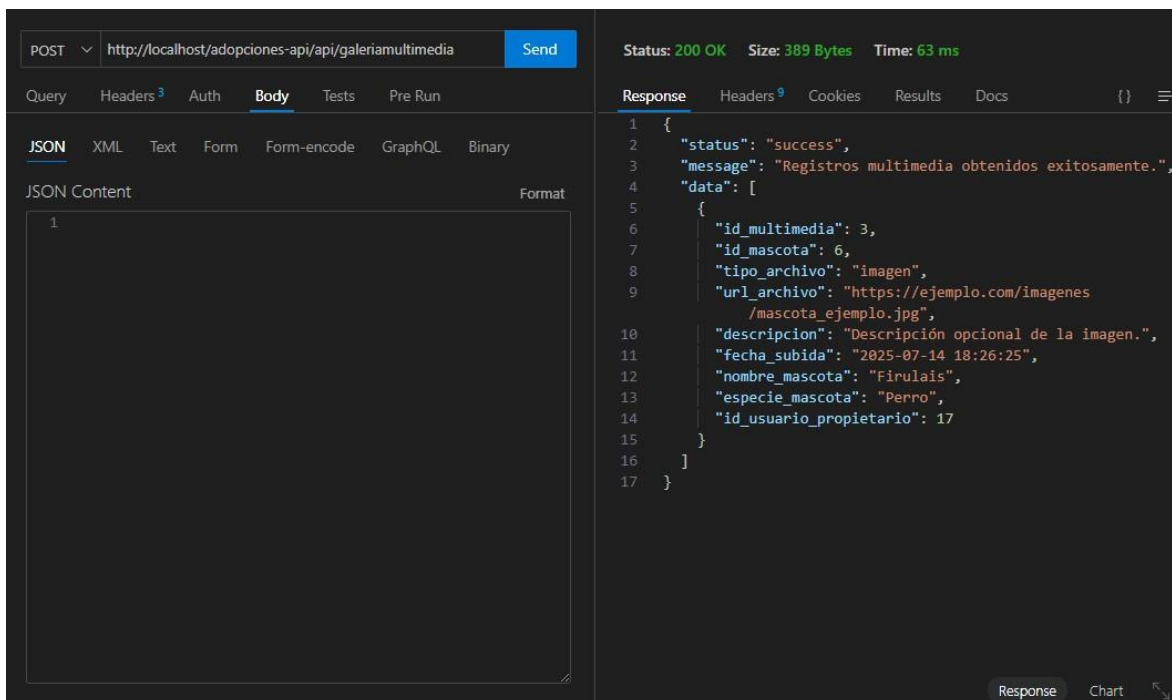
Foto del funcionamiento:



## 9.2. Obtener Todos los Archivos Multimedia

- **Descripción:** Recupera una lista de todos los archivos multimedia registrados para todas las mascotas.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/galeriamultimedia`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Usuario)

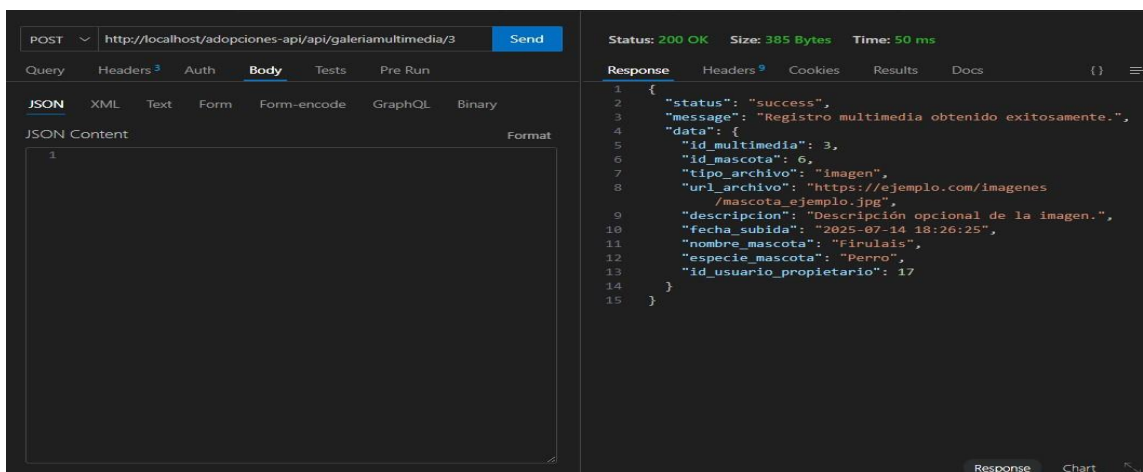
Foto del funcionamiento:



### 9.3. Obtener Archivo Multimedia por ID

- **Descripción:** Recupera los detalles de un archivo multimedia específico por su ID.
- **Método:** GET
- **URL:** http://localhost/adopciones-api/api/galeriamultimedia/(id\_multimedia)
  - **Ejemplo:** http://localhost/adopciones-api/api/galeriamultimedia/1
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Usuario)

Foto del funcionamiento;

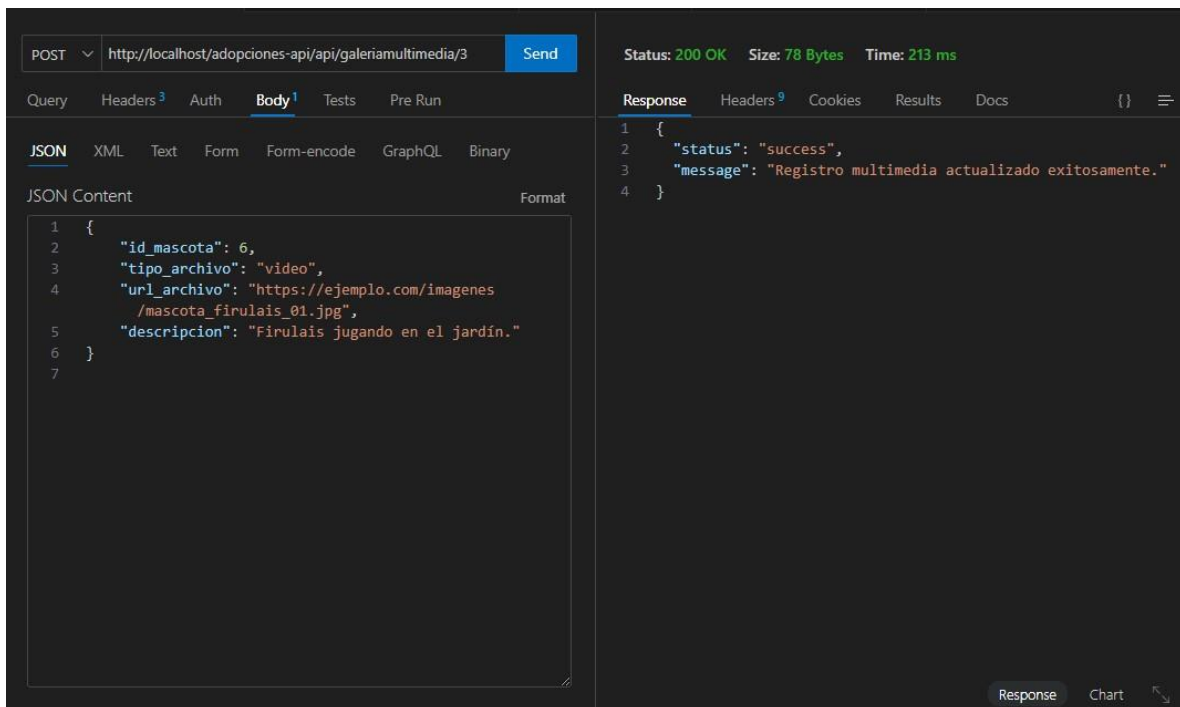




## 9.4. Actualizar Archivo Multimedia

- **Descripción:** Actualiza los detalles de un archivo multimedia existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/galeriamultimedia/(id_multimedia)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/galeriamultimedia/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administardor)

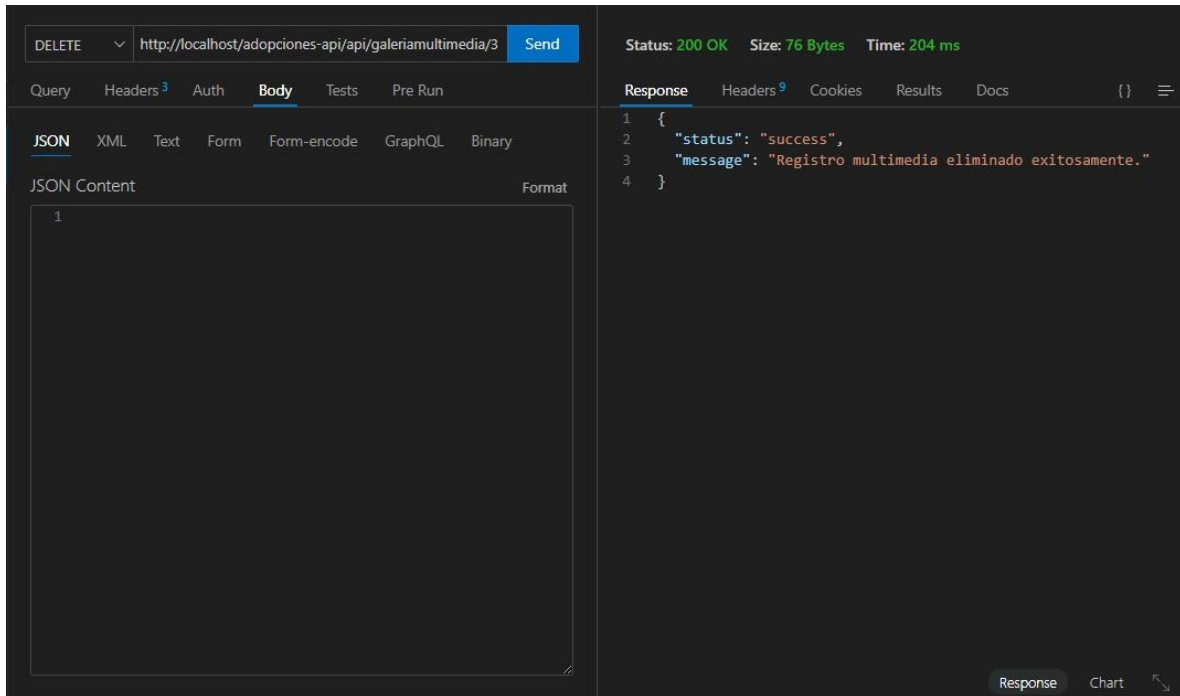
Foto del funcionamiento:



## 9.5. Eliminar Archivo Multimedia

- **Descripción:** Elimina un registro de archivo multimedia del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/galeriamultimedia/(id_multimedia)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/galeria-multimedia/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



## 10. Notificaciones

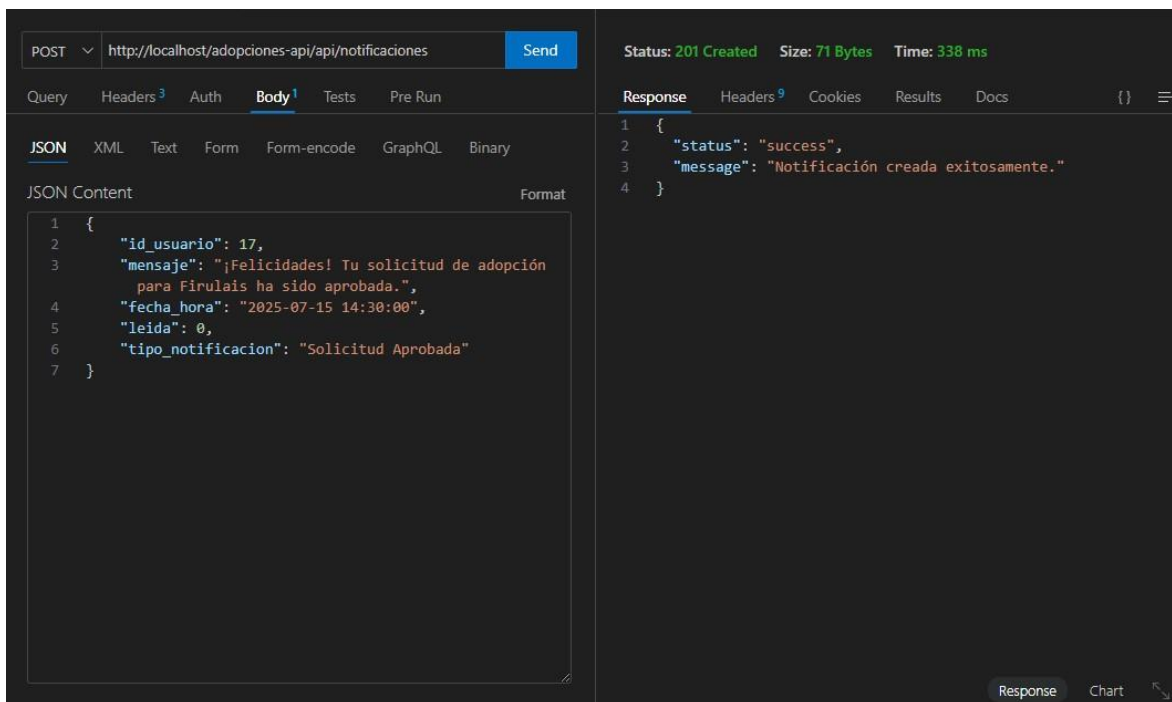
Este controlador (NotificacionesController) gestiona el envío y la visualización de notificaciones para los usuarios del sistema.

**Base URL:** `http://localhost/adopciones-api/api/notificaciones`

### 10.1. Crear Notificación (POST /api/notificaciones)

- Descripción: Crea una nueva notificación para un usuario específico.
- Método: POST
- URL: `http://localhost/adopciones-api/api/notificaciones`
- Encabezados (Headers):
- Content-Type: `application/json`
- Authorization: Bearer (Token Administrador)

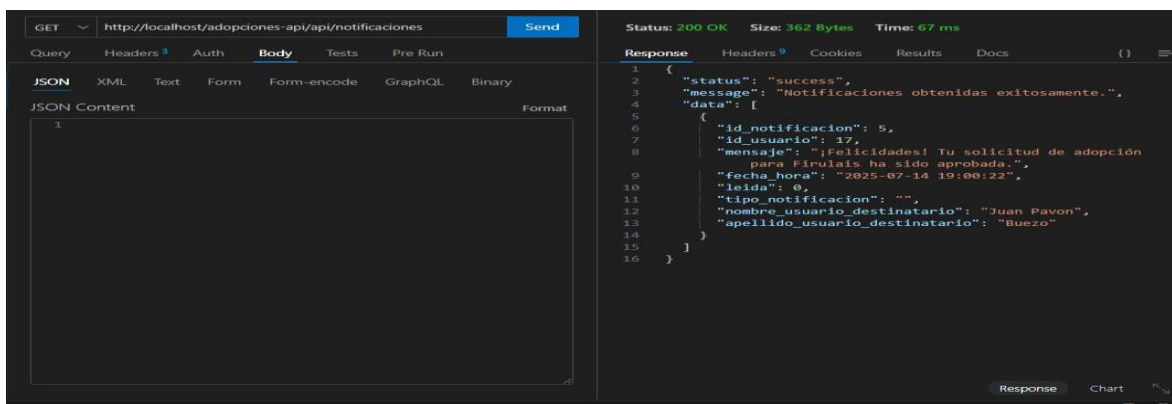
Foto del funcionamiento:



## 10.2. Obtener Todas las Notificaciones (para un usuario específico o todas para administradores)

- **Descripción:** Recupera una lista de notificaciones. Los administradores pueden ver todas, los usuarios normales solo las suyas.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/notificaciones`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador o Usuario)

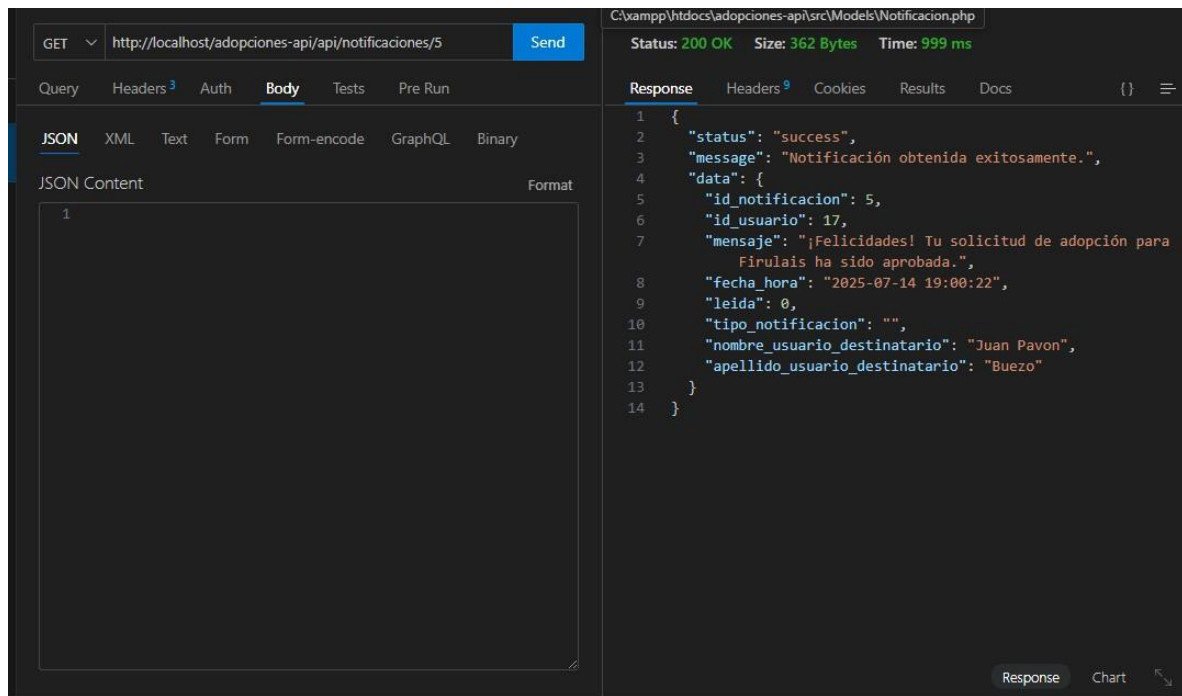
Foto del funcionamiento:



### 10.3. Obtener Notificación por ID

- **Descripción:** Recupera los detalles de una notificación específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/notificaciones/(id_notificacion)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/notificaciones/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador o Usuario)

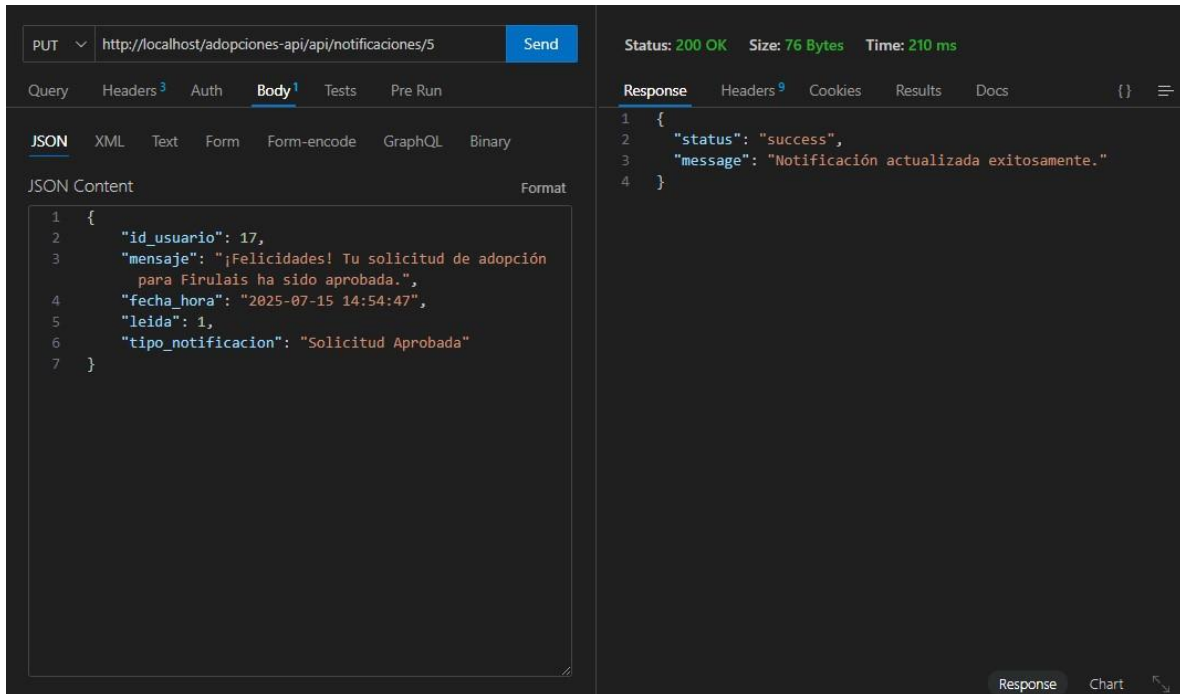
Foto del funcionamiento:



### 10.4. Actualizar Notificación

- **Descripción:** Actualiza los detalles de una notificación existente (comúnmente para marcarla como leída).
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/notificaciones/(id_notificacion)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/notificaciones/1`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador O Usuario)

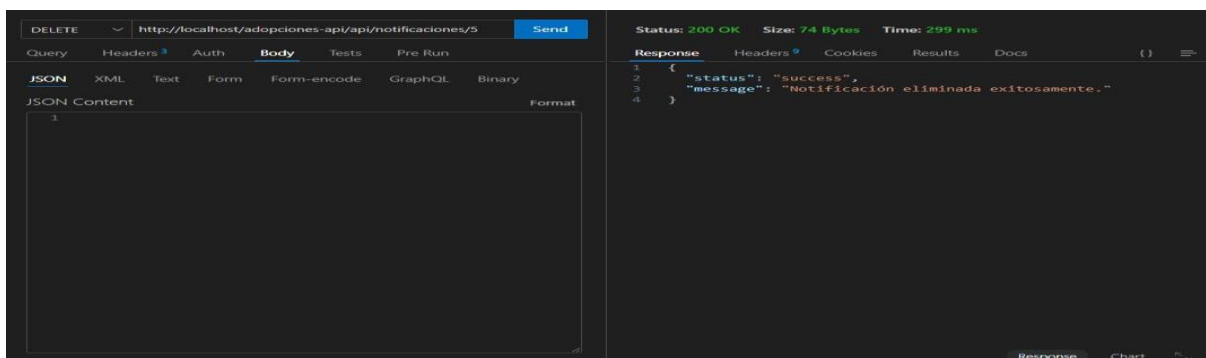
Foto del funcionamiento:



## 10.5. Eliminar Notificación

- **Descripción:** Elimina una notificación del sistema.
- **Método:** DELETE
- **URL:** [http://localhost/adopciones-api/api/notificaciones/{id\\_notificacion}](http://localhost/adopciones-api/api/notificaciones/{id_notificacion})
  - **Ejemplo:** <http://localhost/adopciones-api/api/notificaciones/1>
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



## 11.Evento

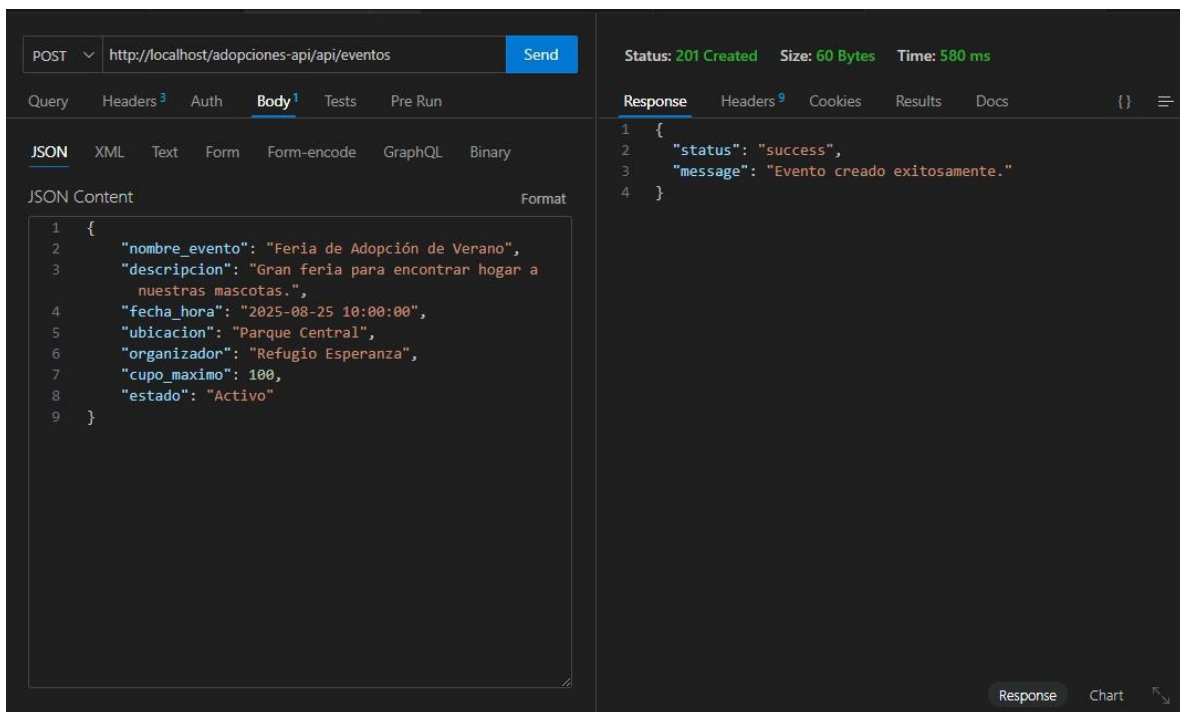
Este controlador (EventoController) gestiona la creación, lectura, actualización y eliminación de los eventos organizados por el sistema (que pueden ser actividades, ferias, etc.).

**Base URL:** <http://localhost/adopciones-api/api/evento>

### 11.1. Crear Evento (POST /api/evento)

- **Descripción:** Crea un nuevo registro de evento en el sistema.
- **Método:** POST
- **URL:** <http://localhost/adopciones-api/api/eventos>
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

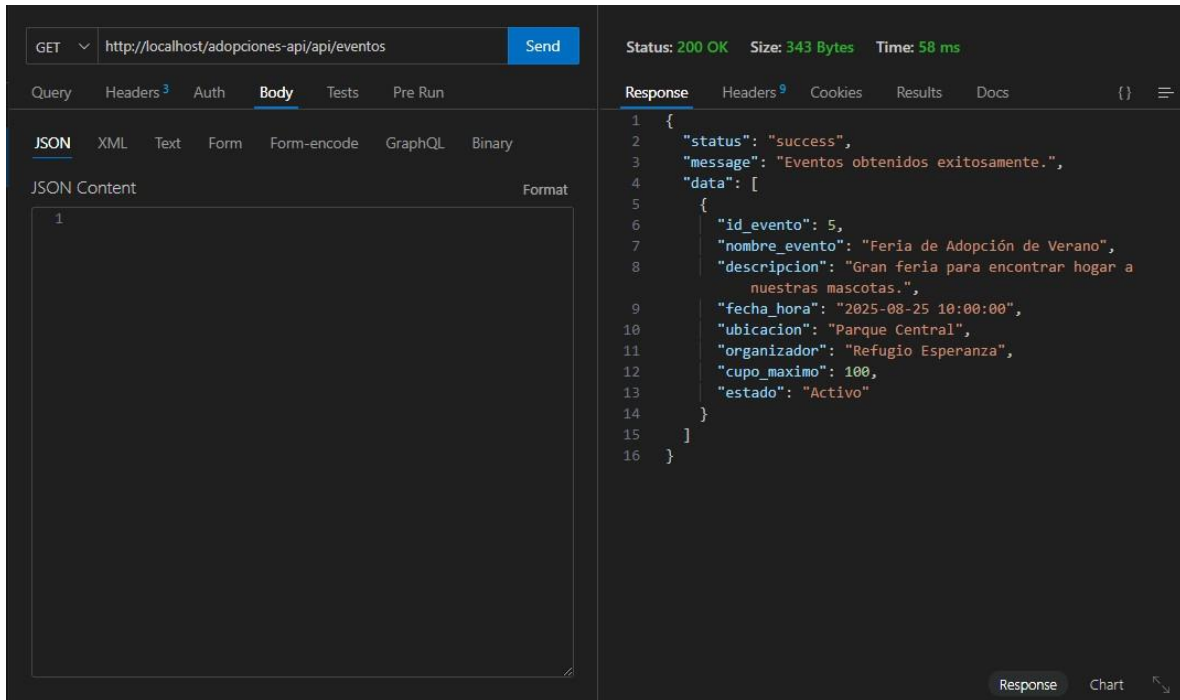
Foto del funcionamiento:



## 11.2. Obtener Todos los Eventos

- **Descripción:** Recupera una lista de todos los eventos registrados.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/eventos`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador o Usuario)

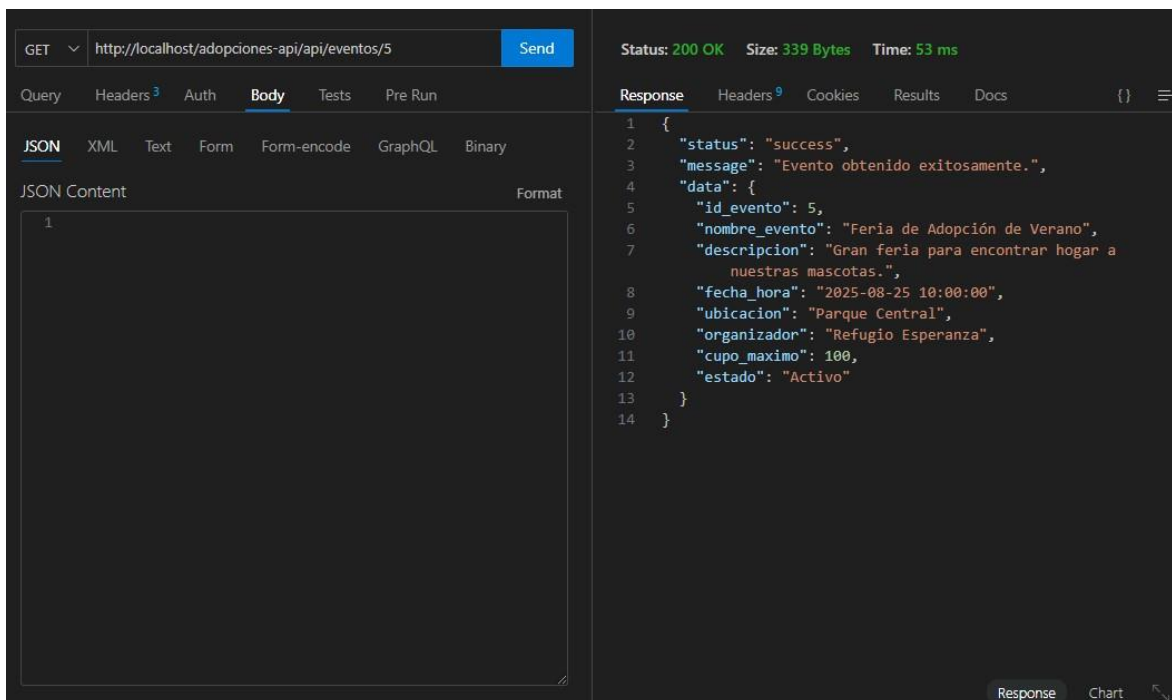
Foto del funcionamiento:



## 11.3. Obtener Evento por ID

- **Descripción:** Recupera los detalles de un evento específico por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/eventos/(id_evento)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/evento/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador o Usuario)

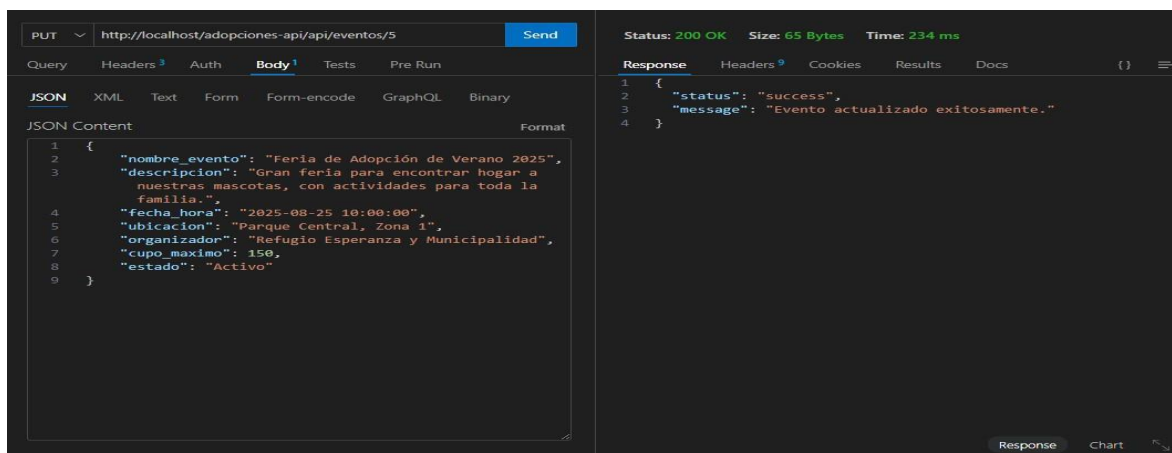
Foto del funcionamiento:



## 11.4. Actualizar Evento

- **Descripción:** Actualiza los detalles de un evento existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/eventos/(id_evento)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/evento/1`
- **Encabezados (Headers):**
  - Content-Type: `application/json`
  - Authorization: `Bearer (Token Administrador)`

Foto del funcionamiento:

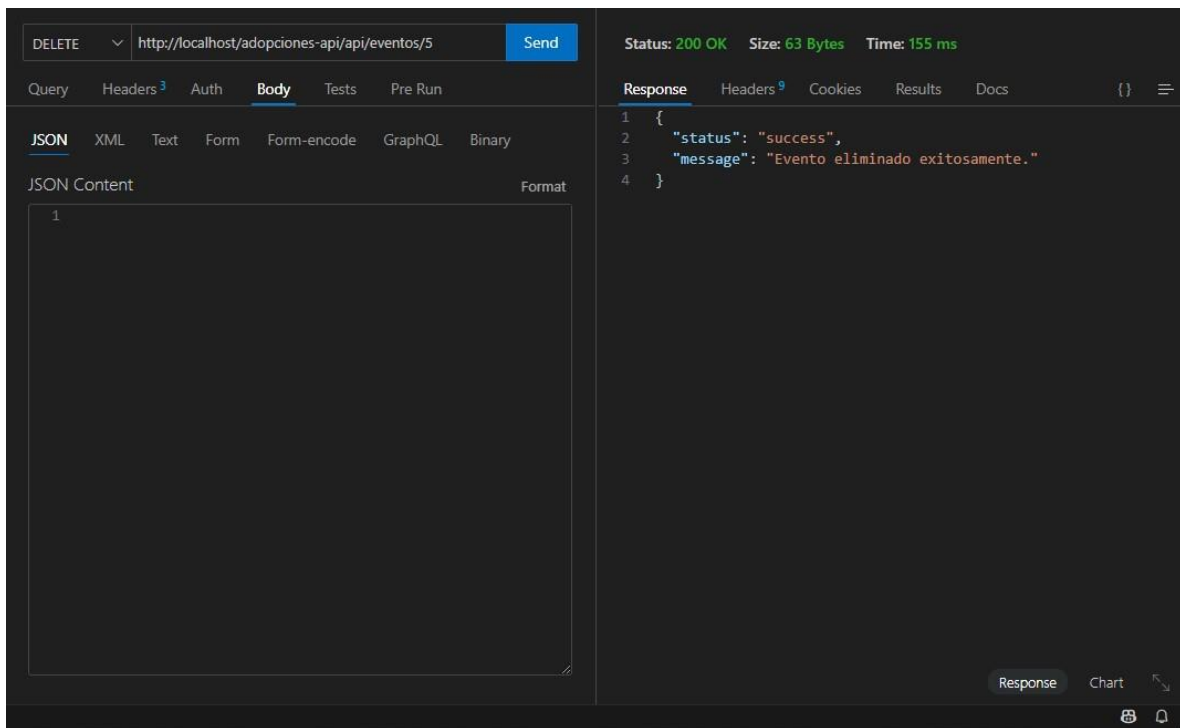




## 11.5. Eliminar Evento

- **Descripción:** Elimina un evento del sistema.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/eventos/(id_evento)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/evento/2`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



## 12. Participación en Eventos

Este controlador (ParticipacionEventoController) gestiona el registro de la vinculación de usuarios con los eventos.

**Base URL:** `http://localhost/adopciones-api/api/participacionevento`

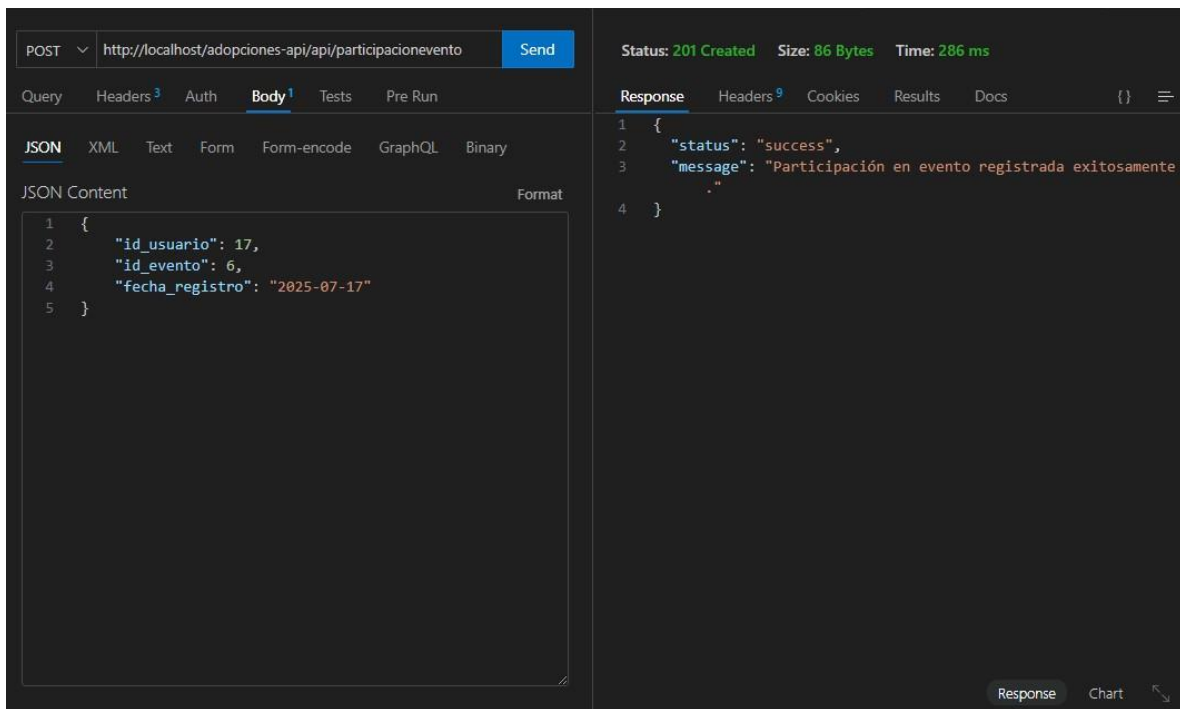
### 12.1. Crear Participación en Evento

**Descripción:** Registra la participación de un usuario en un evento específico.

- **Método:** POST
- **URL:** `http://localhost/adopciones-api/api/participacionevento`
- **Encabezados (Headers):**
  - Content-Type: application/json

- Authorization: Bearer (Token Administrador o Usuario)

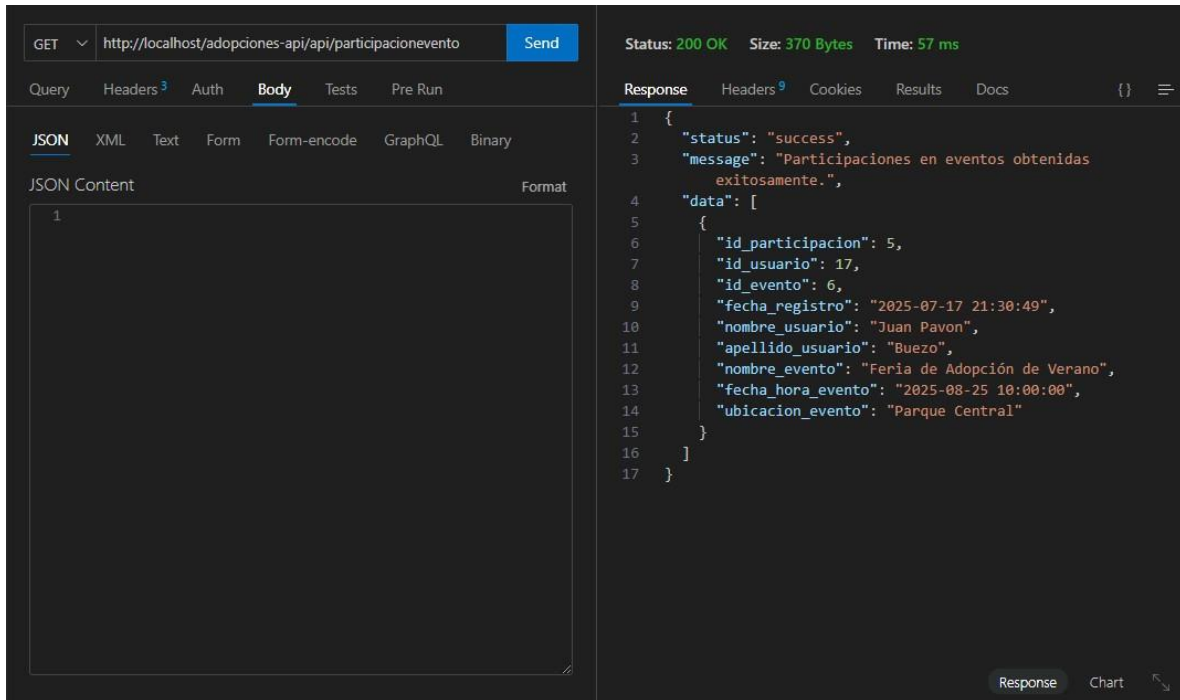
Foto del funcionamiento:



## 12.2. Obtener Todas las Participaciones

- **Descripción:** Recupera una lista de todas las participaciones en eventos. Puede ser filtrada por id\_evento o id\_usuario.
- **Método:** GET
- **URL:** http://localhost/adopciones-api/api/participacionevento
- **Parámetros de Consulta (Query Parameters - Opcional):**
  - id\_evento: Filtra las participaciones por un evento específico.
  - id\_usuario: Filtra las participaciones por un usuario específico.
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador o Usuario)

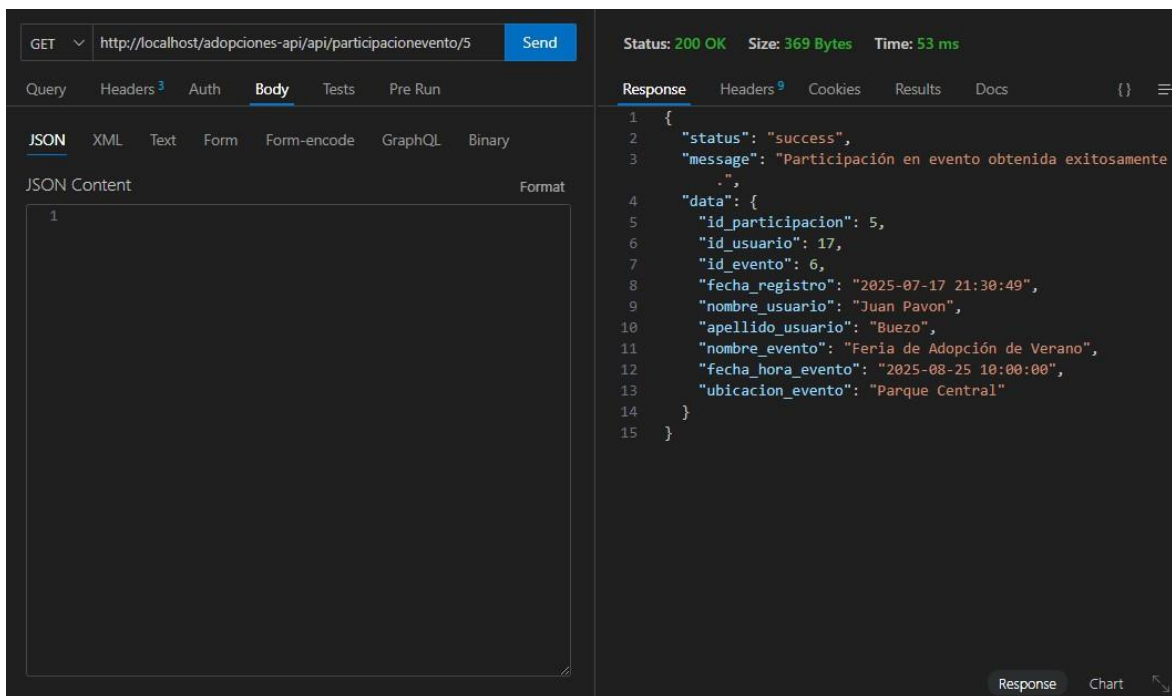
Foto del funcionamiento:



### 12.3. Obtener Participación por ID

- **Descripción:** Recupera los detalles de una participación específica por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/participacionevento/(id_participacion)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/participacionevento/1`
- **Encabezados (Headers):**
  - Authorization: Bearer TU\_TOKEN\_JWT\_AQUI (Requiere autenticación. Un **administrador** puede ver cualquier participación; un **usuario regular** solo puede ver *sus propias* participaciones).

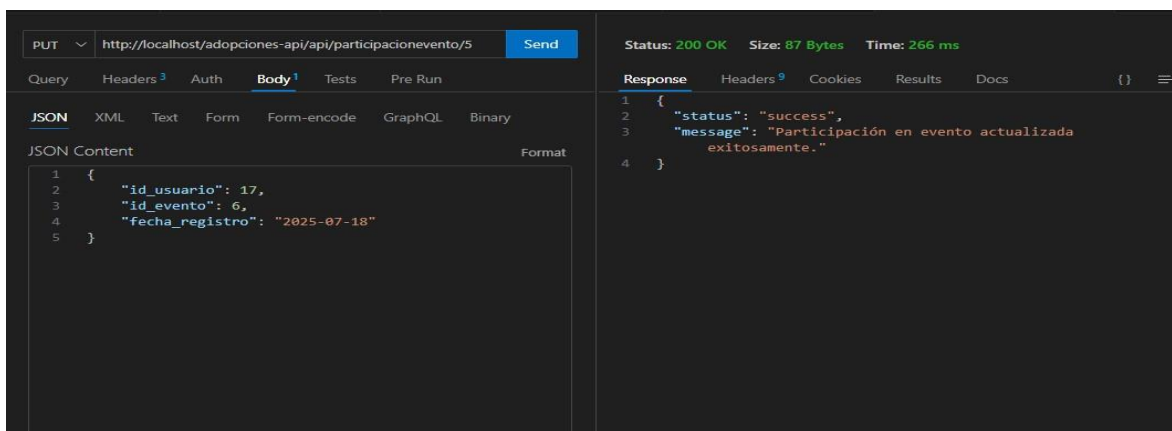
Foto del funcionamiento:



## 12.4. Actualizar Participación en Evento

- **Descripción:** Actualiza los detalles de una participación existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/participacionevento/(id_participacion)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/participacionevento/1`
- **Encabezados (Headers):**
  - Content-Type: `application/json`
  - Authorization: Bearer (Token Administrador o Usuario)

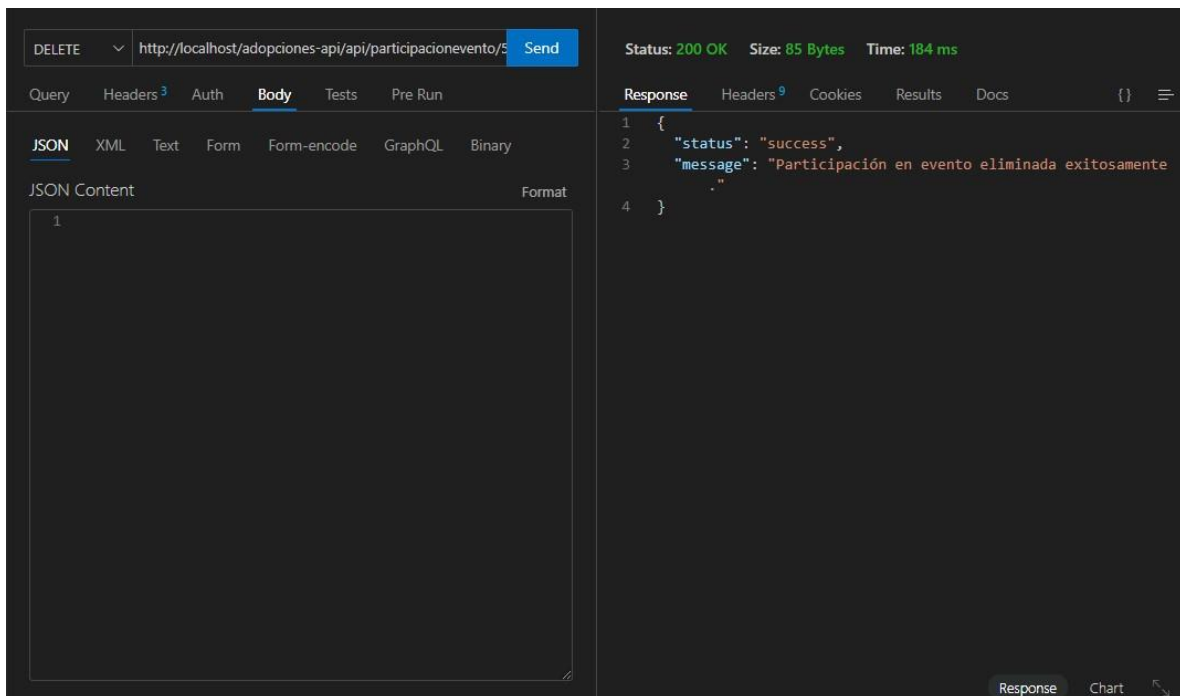
Foto del funcionamiento:



## 12.5. Eliminar Participación en Evento

- **Descripción:** Elimina un registro de participación en evento.
- **Método:** DELETE
- **URL:** `http://localhost/adopciones-api/api/participacionevento/{id_participacion}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/participacionevento/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



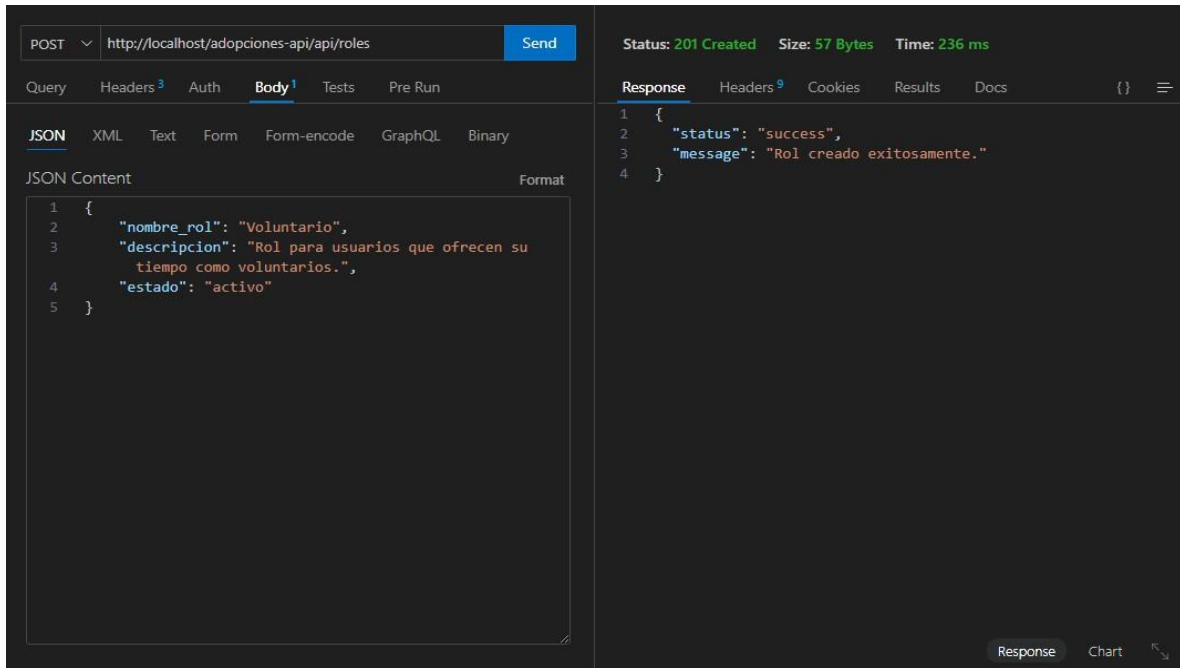
## 13. Roles

**Base URL:** `http://localhost/adopciones-api/api/roles`

### 13.1. Crear Rol

- **Descripción:** Crea un nuevo rol de usuario en el sistema.
- **Método:** POST
- **URL:** `http://localhost/adopciones-api/api/roles`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

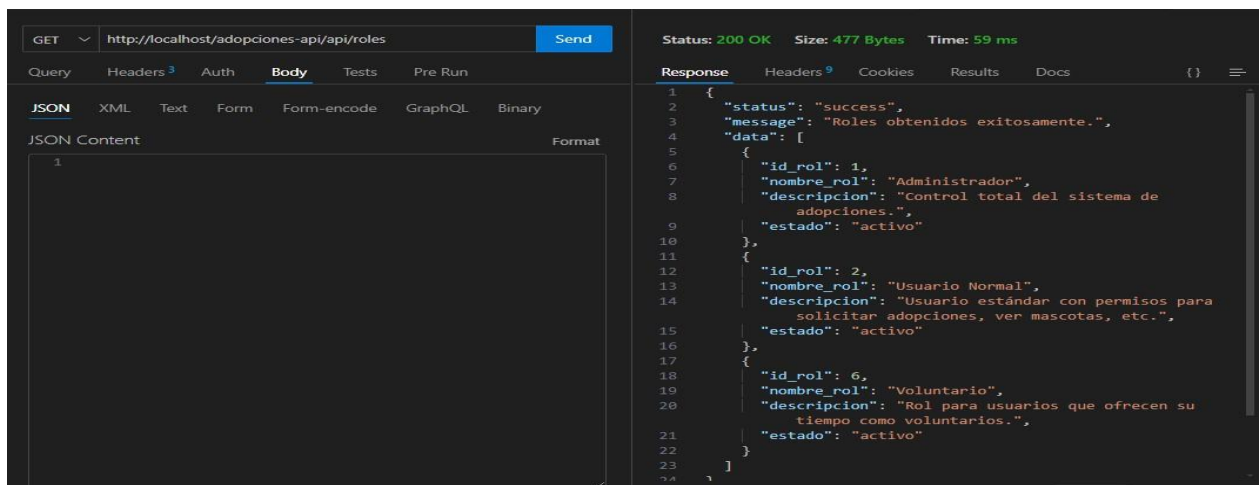
Foto del funcionamiento:



### 13.2. Obtener Todos los Roles

- **Descripción:** Recupera una lista de todos los roles de usuario registrados.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/roles`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

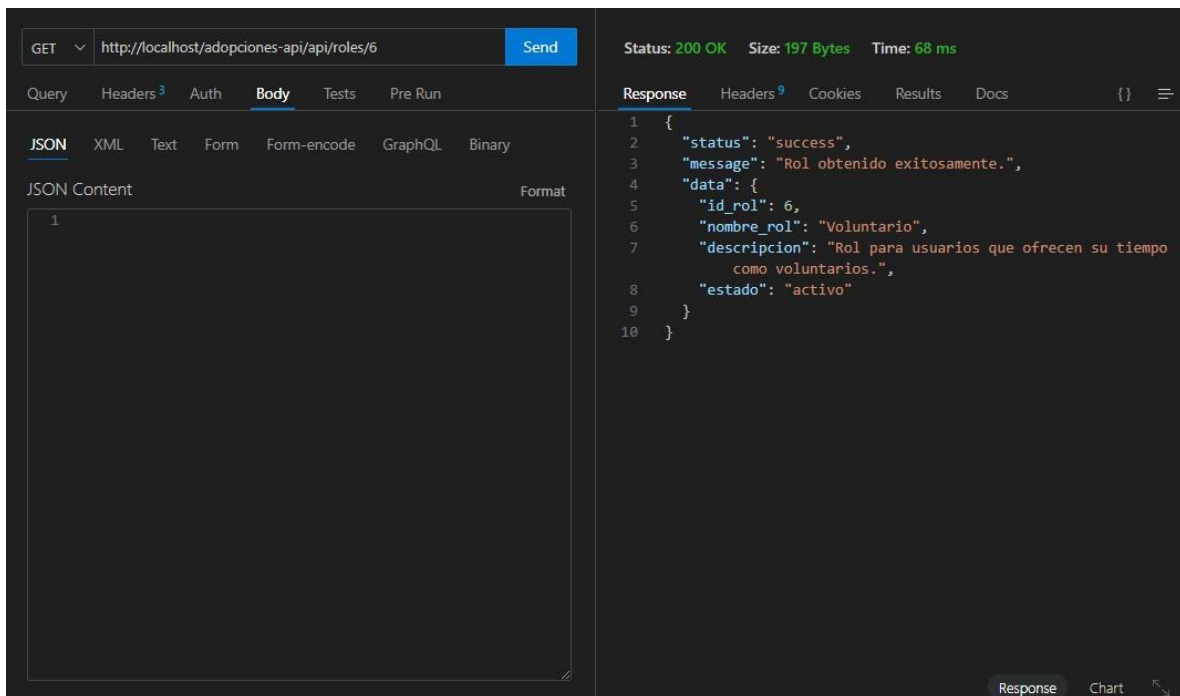
Foto del funcionamiento:



### 13.3. Obtener Rol por ID

- **Descripción:** Recupera los detalles de un rol específico por su ID.
- **Método:** GET
- **URL:** `http://localhost/adopciones-api/api/roles/(id_rol)`
  - **Ejemplo:** `http://localhost/adopciones-api/api/roles/1`
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administrador)

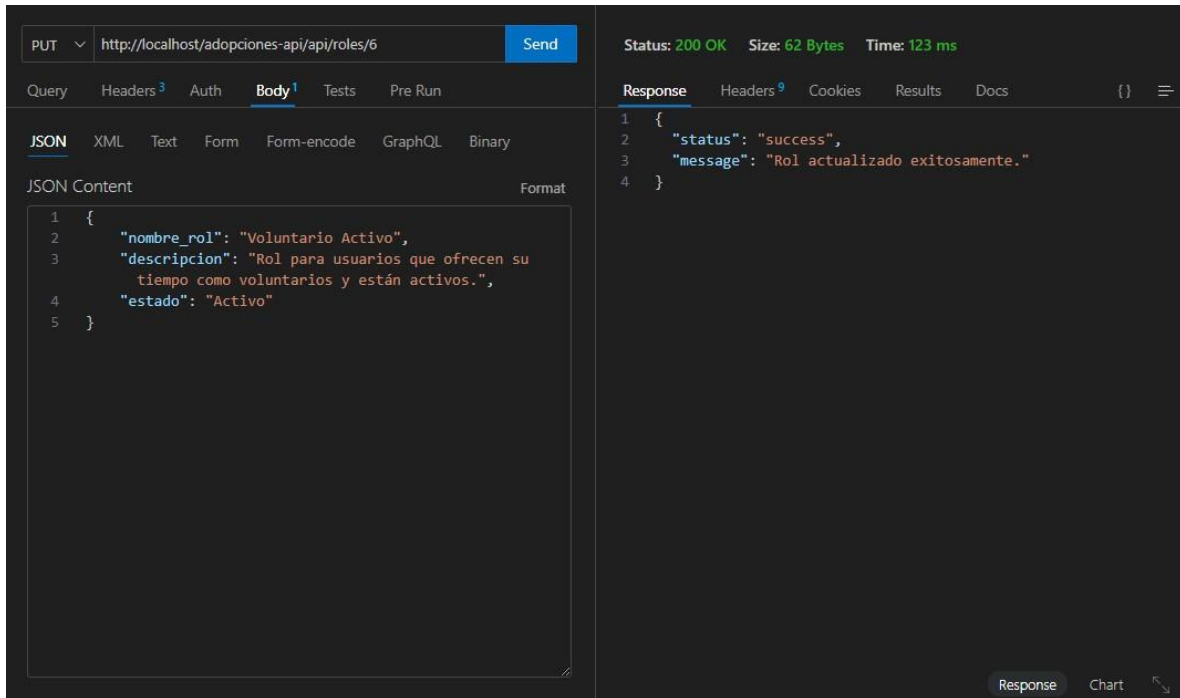
Foto del funcionamiento:



### 13.4. Actualizar Rol

- **Descripción:** Actualiza los detalles de un rol existente.
- **Método:** PUT
- **URL:** `http://localhost/adopciones-api/api/roles/{id_rol}`
  - **Ejemplo:** `http://localhost/adopciones-api/api/roles/3`
- **Encabezados (Headers):**
  - Content-Type: application/json
  - Authorization: Bearer (Token Administrador)

Foto del funcionamiento:



### 13.5. Eliminar Rol

- **Descripción:** Elimina un rol del sistema.
- **Método:** DELETE
- **URL:** http://localhost/adopciones-api/api/roles/{id\_rol}
  - **Ejemplo:** http://localhost/adopciones-api/api/roles/3
- **Encabezados (Headers):**
  - Authorization: Bearer (Token Administardor)

Foto del funcionamiento:

