

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Lenguajes de programación**

**2S-2019**

Integrantes:

* Iliana Estefania Bolaños Fajardo
* Carlos Anthony Sesme Vera
* Tony Freddy Veas Cervantes

Evidencias BACKEND

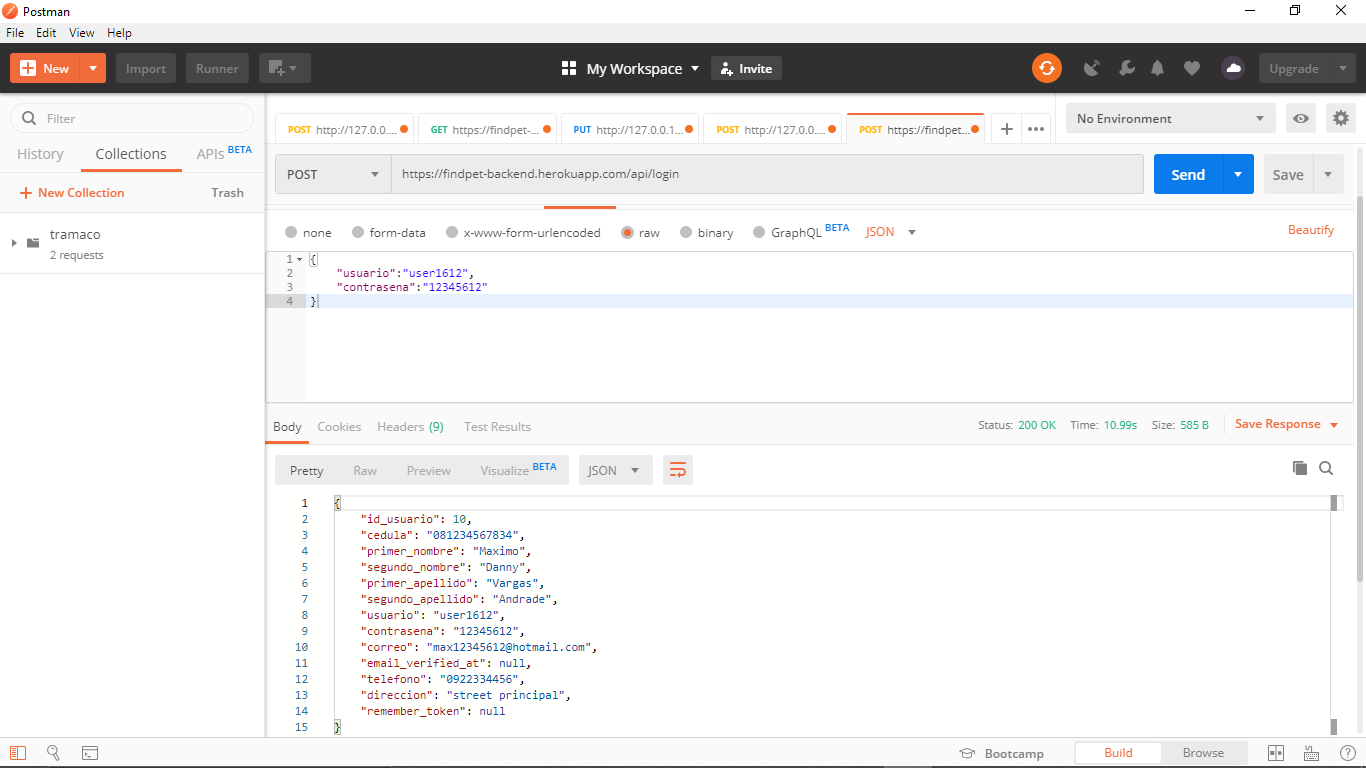
El backend se desarrolló usando el lenguaje de programación PHP y el Framework Laravel, para realizar el consumo de los servicios desde la app móvil se desplegó el Rest API en Heroku y de igual forma la Base de datos (MySQL) fue alojada en Clever cloud, que nos provee un alojamiento gratuito.

Para mostrar el funcionamiento de cada uno de los servicios que se implementaron se usará Postman

Servicios de BackEnd (API REST):

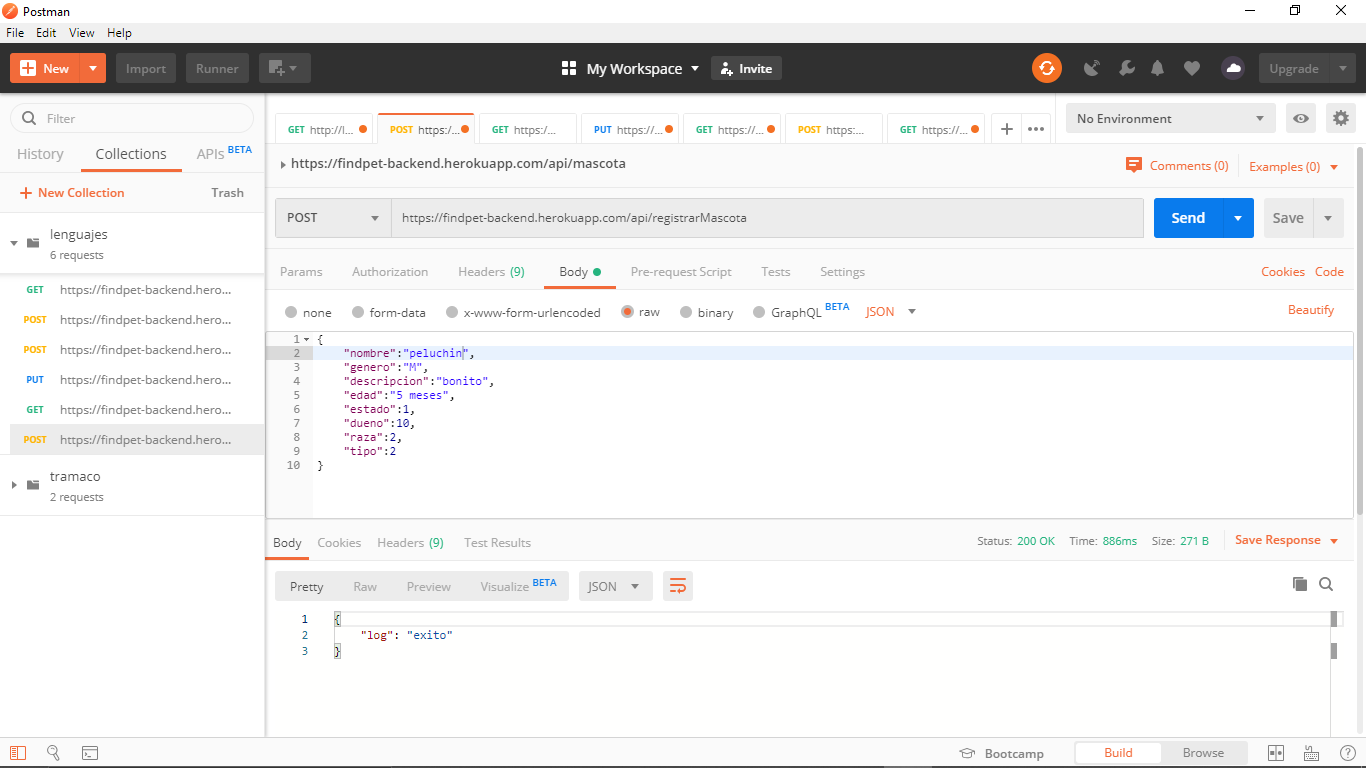
* **Login (POST)**

El servicio recibe como parámetros del request el usuario y la contraseña, si estas credenciales existen en la base de datos el servicio retorna como response la data más importante del usuario la cual se registra en el app móvil para que el usuario quede logeado y no tenga que iniciar sesión cada vez que entra al app.

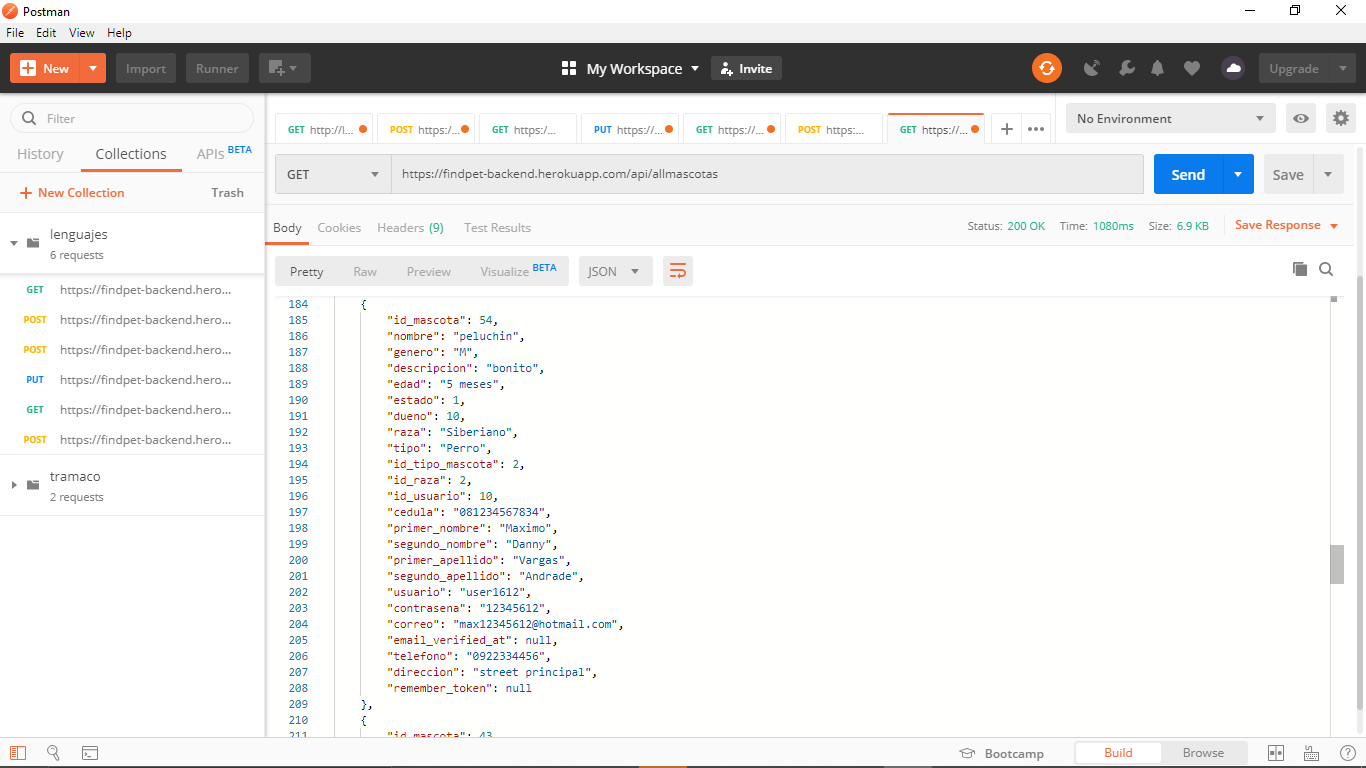


* **RegistrarMascotas (POST)**

El servicio de registro de mascotas recibe los datos de la mascota dentro del Body del request, si el registro es exitoso el servicio retorna como response un código de status 200 junto con un mensaje de “éxito”.

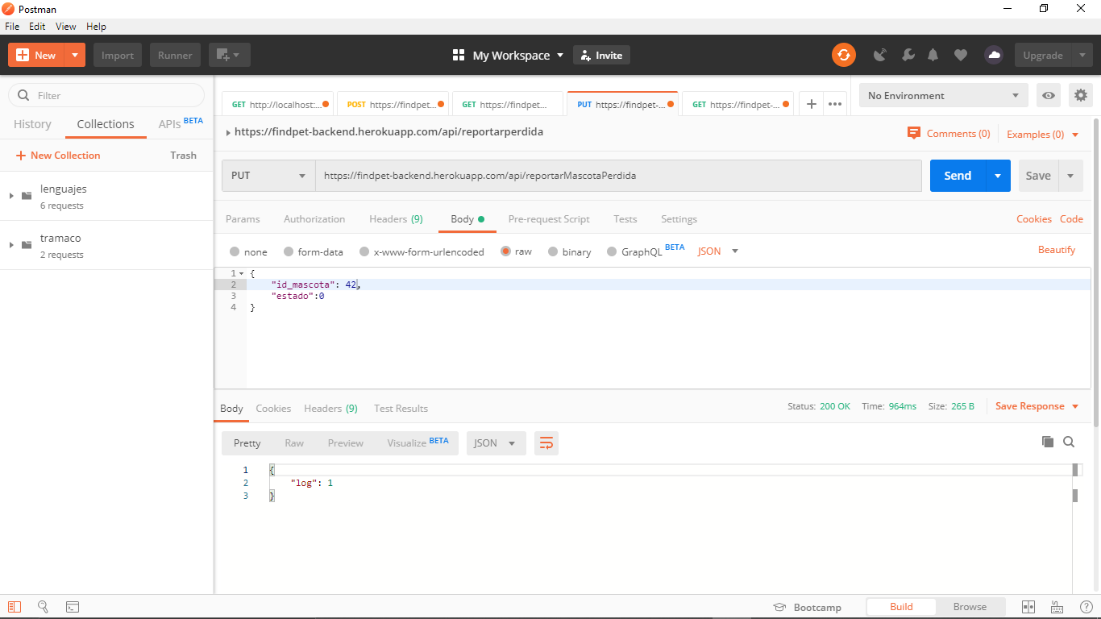


También se creó un servicio que retorna todas las mascotas registradas, en el response de este servicio se puede observar la mascota registrada en la captura anterior con sus respectivos datos.

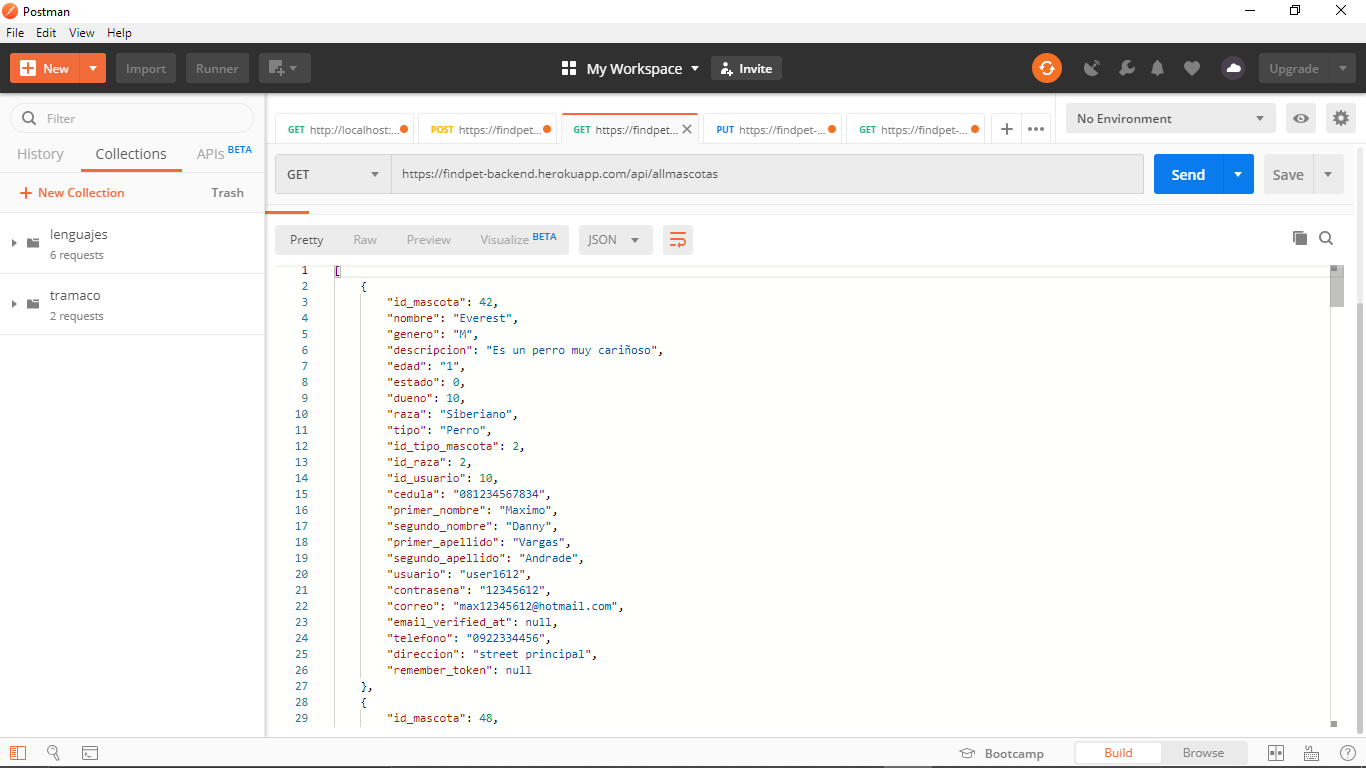


* **ReportarMascotaPerdida** **(PUT)**

Para reportar una mascota perdida lo que se realiza es un “UPDATE” en el atributo “estado” de la mascota, para esto, el servicio recibe en el body del request el ID de la mascota y el nuevo estado de la misma. El response muestra el número de registros actualizados durante la ejecución del servicio.



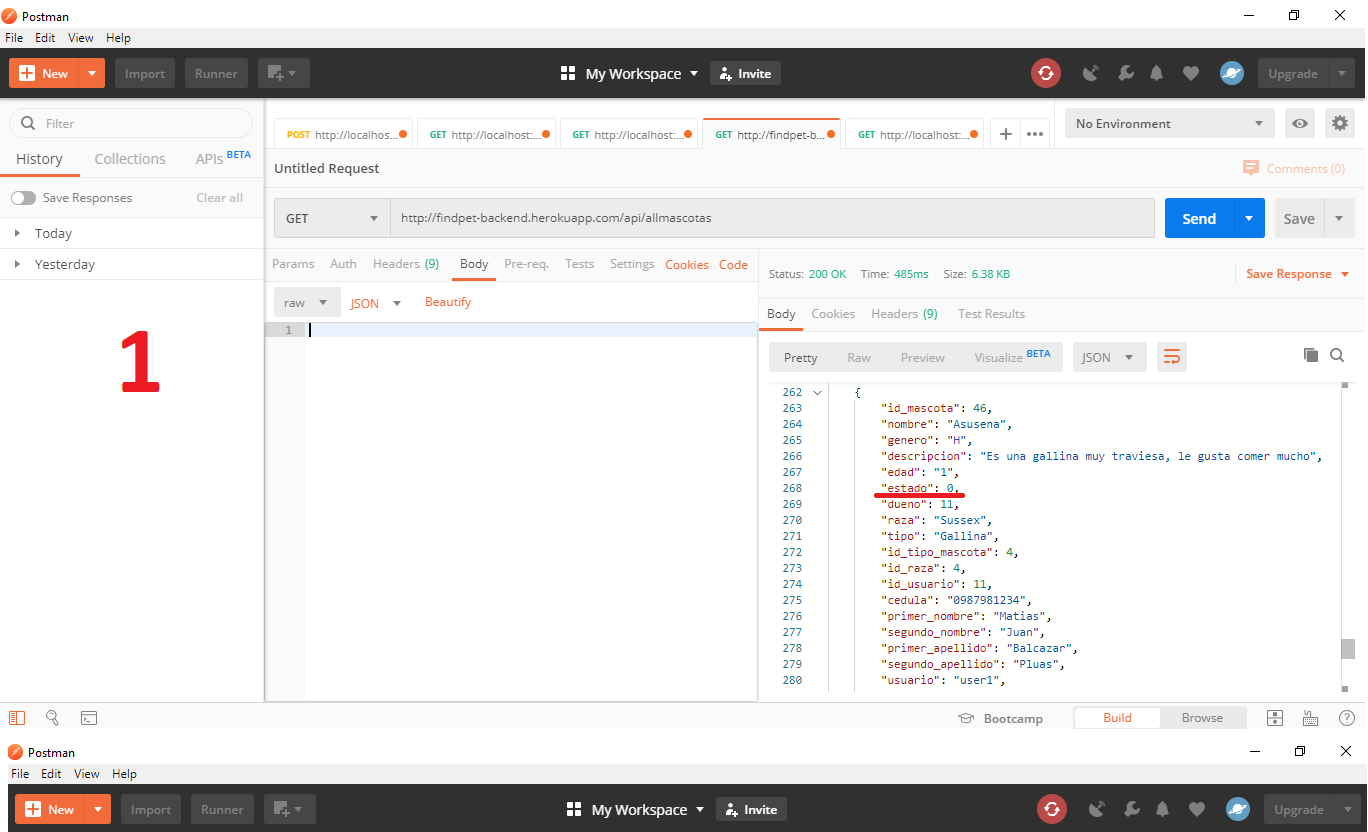
Al consultar el servicio que permite ver todas las mascotas registradas, podemos ver el registro actualizado. La mascota posee el nuevo estado.



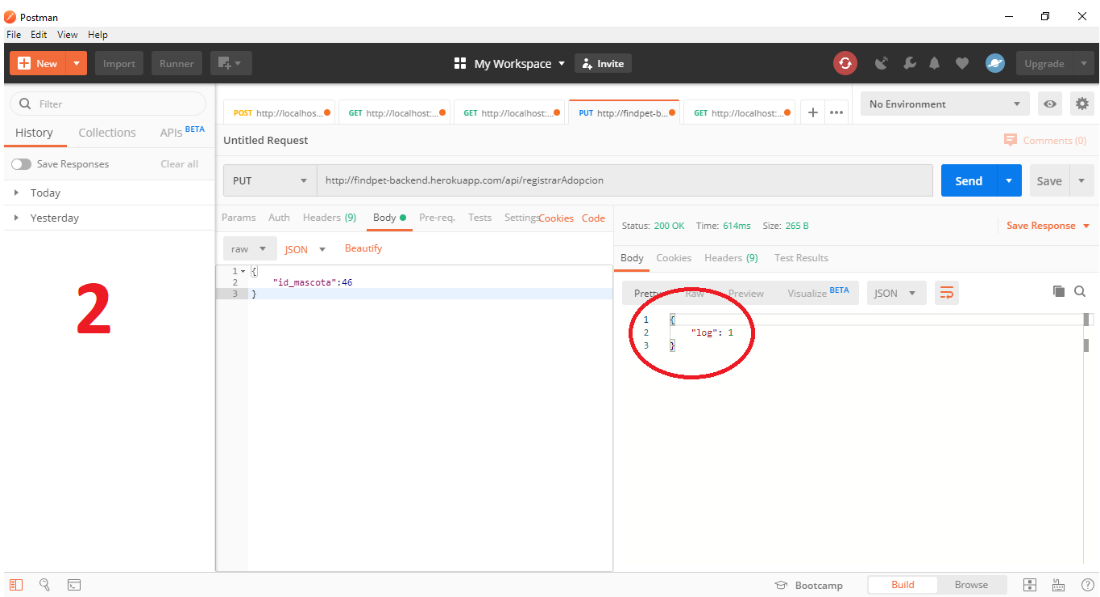
* **RegistrarAdopcion(PUT)**

Para poner una mascota en adopción, lo que se realiza es un “UPDATE” en el atributo “estado” de la mascota, para esto, el servicio recibe en el body del request el ID de la mascota a poner en adopción (hay que tener en cuenta que se definió un estado igual a 1 para identificar a una mascota como “en adopción”). El response muestra el número de registros actualizados durante la ejecución del servicio.

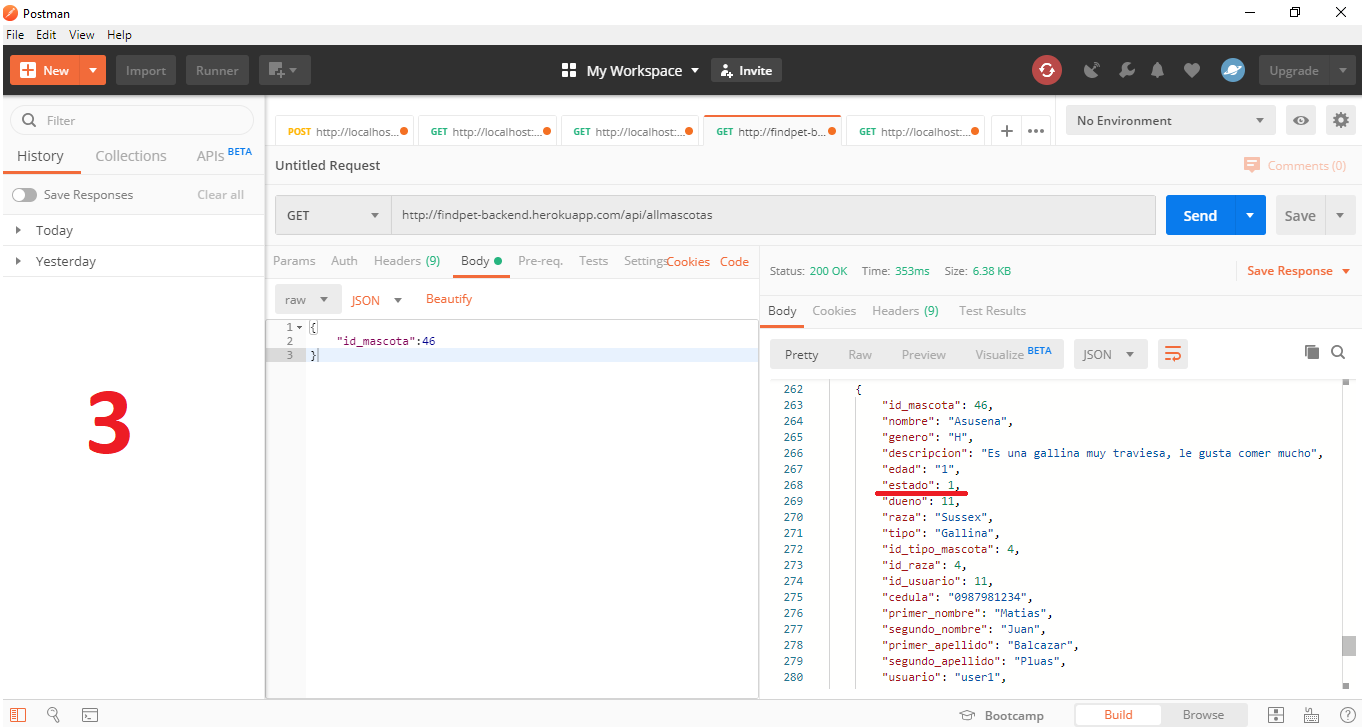
A continuación mostramos todas las mascotas con un servicio que permite ver las mascotas registradas, en este caso observamos que hay una mascota con id\_mascota:46 con un estado: 0.



Luego, usando el servicio registrarAdopcion que recibe en el body del request el ID de la mascota a poner en adopción (hay que tener en cuenta que se definió un estado igual a 1 para identificar a una mascota como “en adopción”).

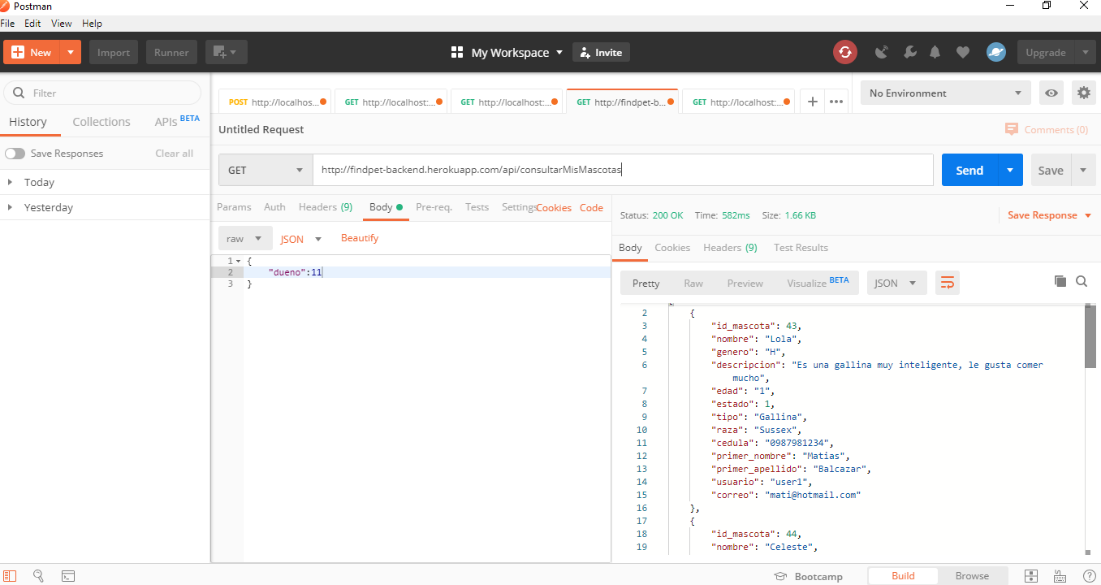


Finalmente podemos comprobar mediante el servicio que permite ver las mascotas registradas, como cambió el estado de la mascota.



* **MostrarMisMascotas(GET)**

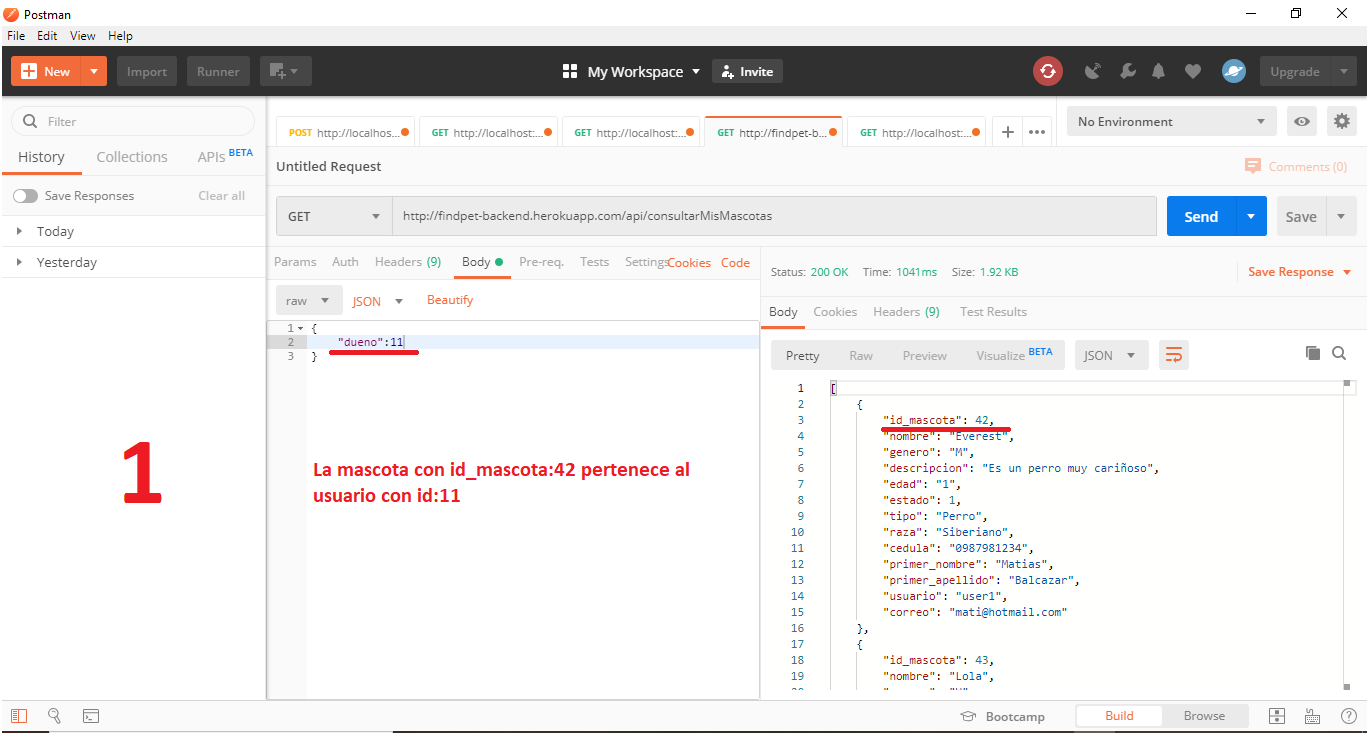
Este servicio nos permite consultar todas las mascotas que pertenecen a un determinado dueño pasando en el body del request el ID del dueño.



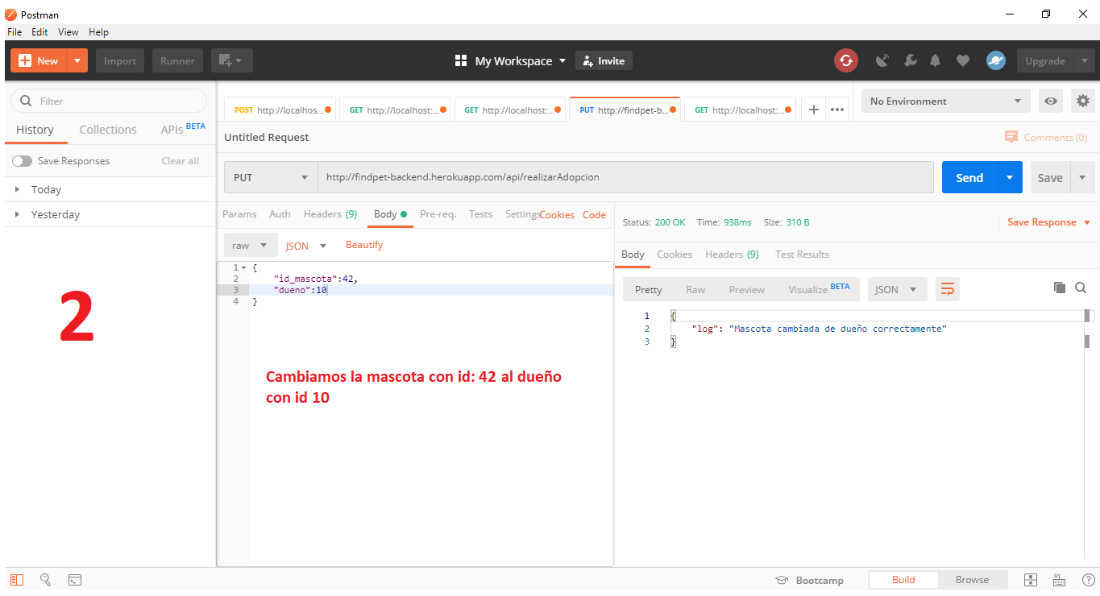
* **RealizarAdopción(PUT)**

Realizar una adopción consiste en cambiar de dueño para lo cual se realiza un “UPDATE” en el atributo “dueno” de la mascota, para esto, el servicio recibe en el body del request el ID de la mascota y el ID del nuevo dueño. El response muestra un mensaje diciendo Mascota cambiada correctamente de dueño.

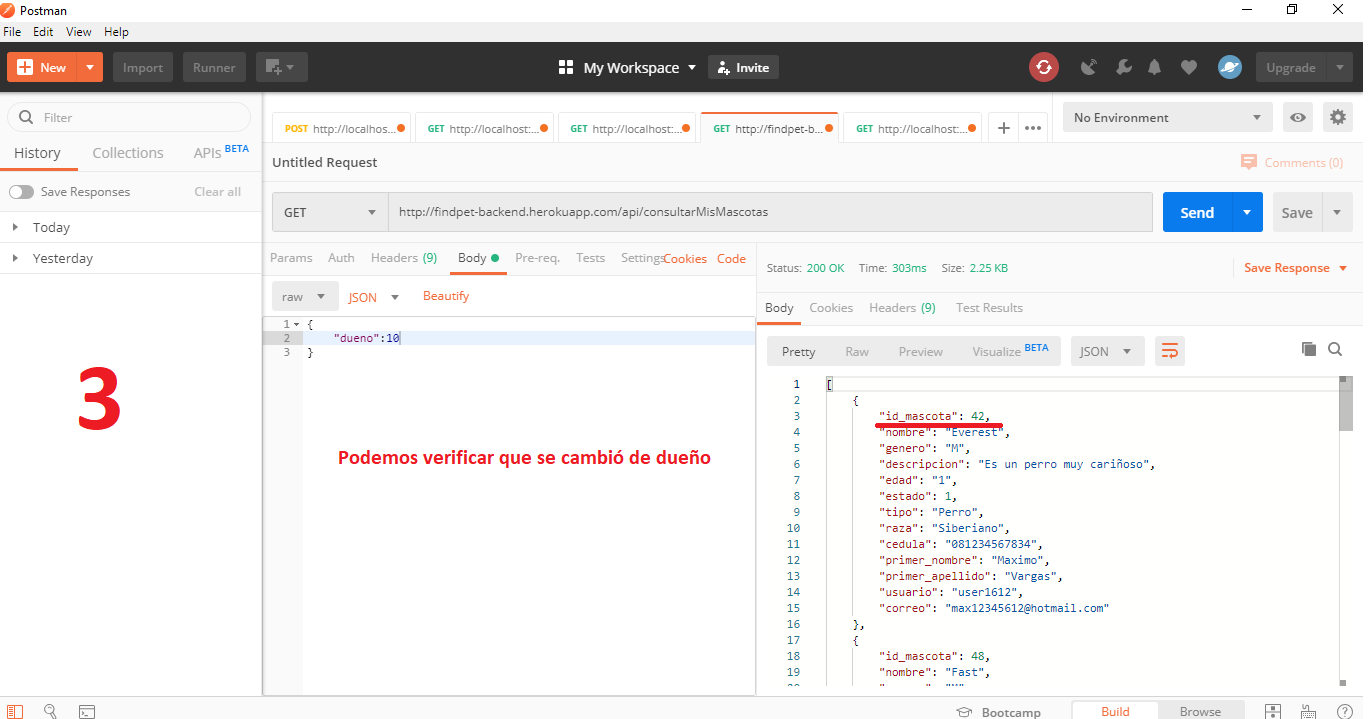
A continuación con el servicio consultarMisMascotas, podemos ver que la mascota con ID 42 pertenece al usuario con ID 11.



Luego usuamos el servicio realizarAdopcion para cambiar de dueño, donde este servicio recibe en el Body del request el ID de la mascota y el ID del nuevo dueño.

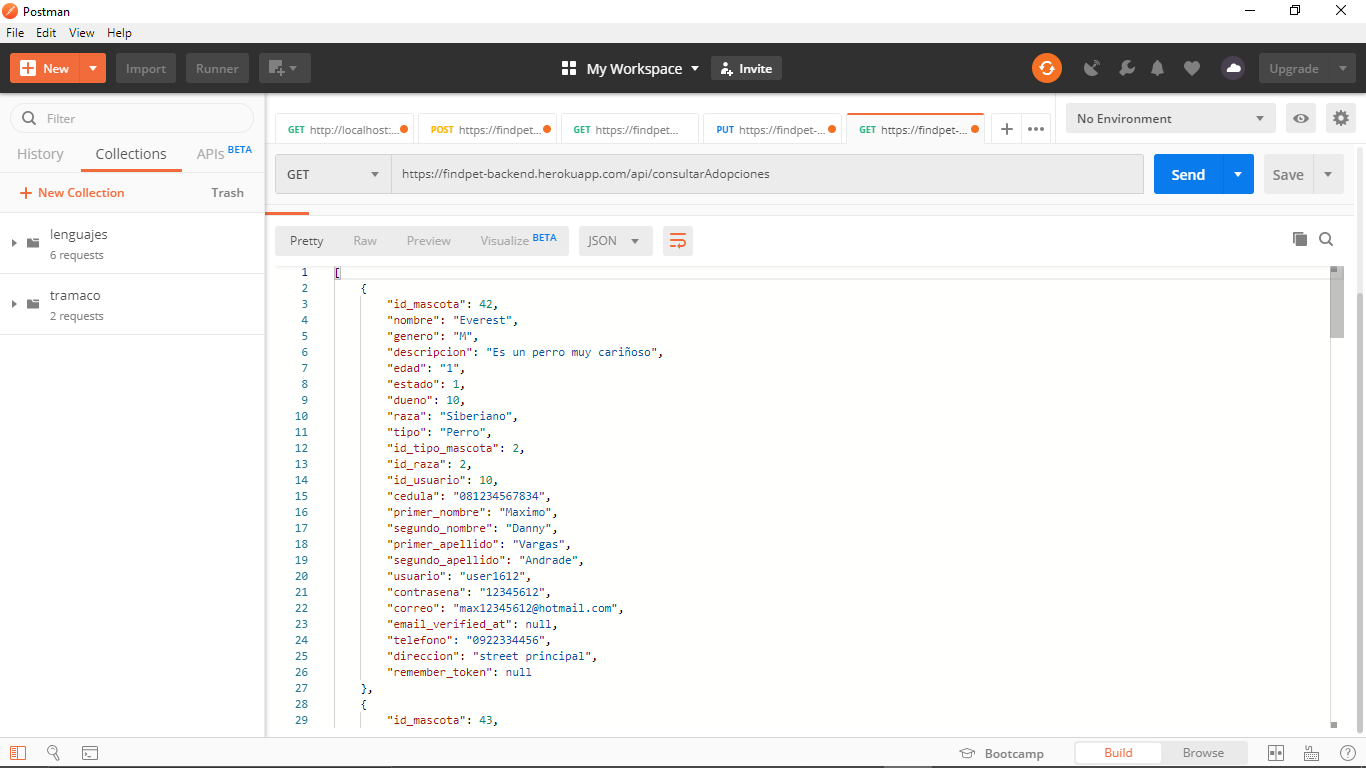


Finalmente podemos usar el servicio consultarMisMascota para verificar que ahora la mascota con ID 42 ahora pertenece al dueño con ID 10.



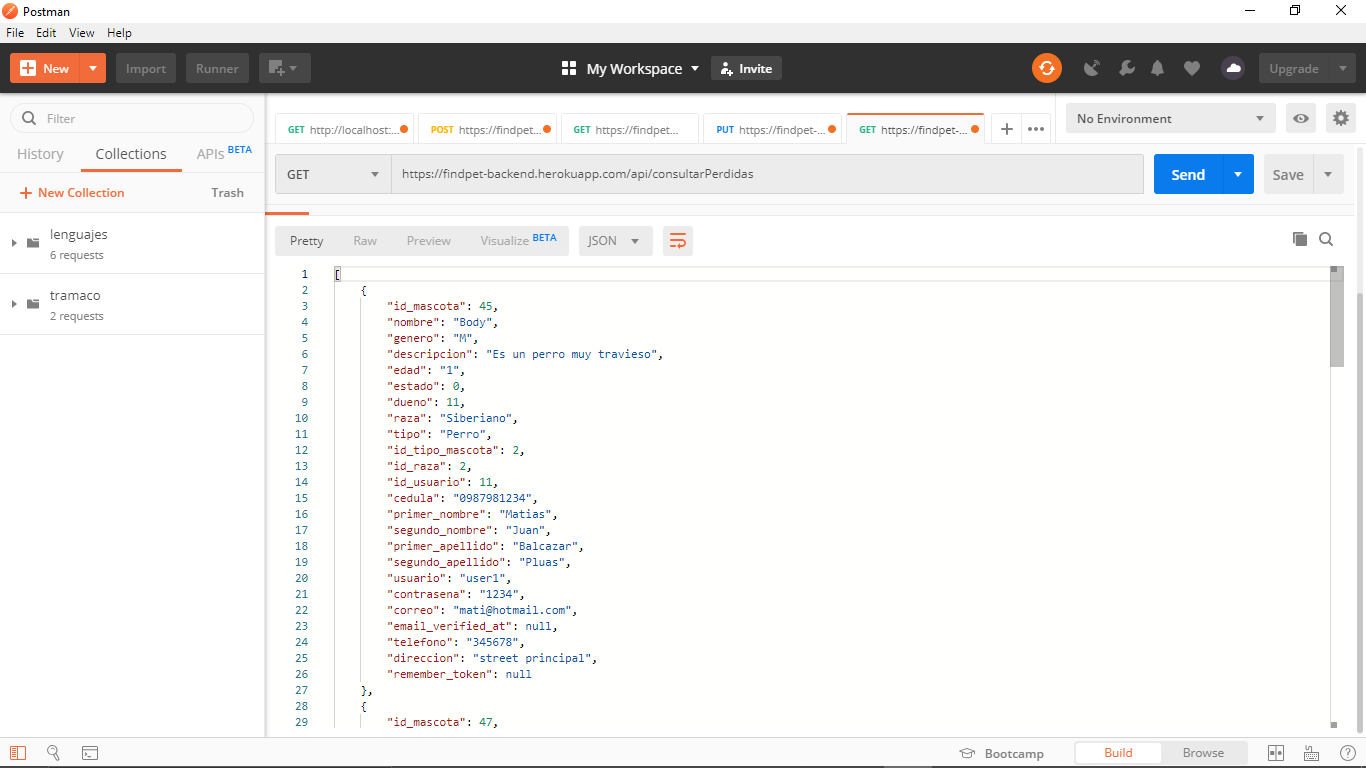
* **ConsultarAdopciones (GET)**

Este servicio no recibe ningún parámetro dentro del body del request al ser un método GET y retorna todas las mascotas que posean un “estado” con valor de 1.



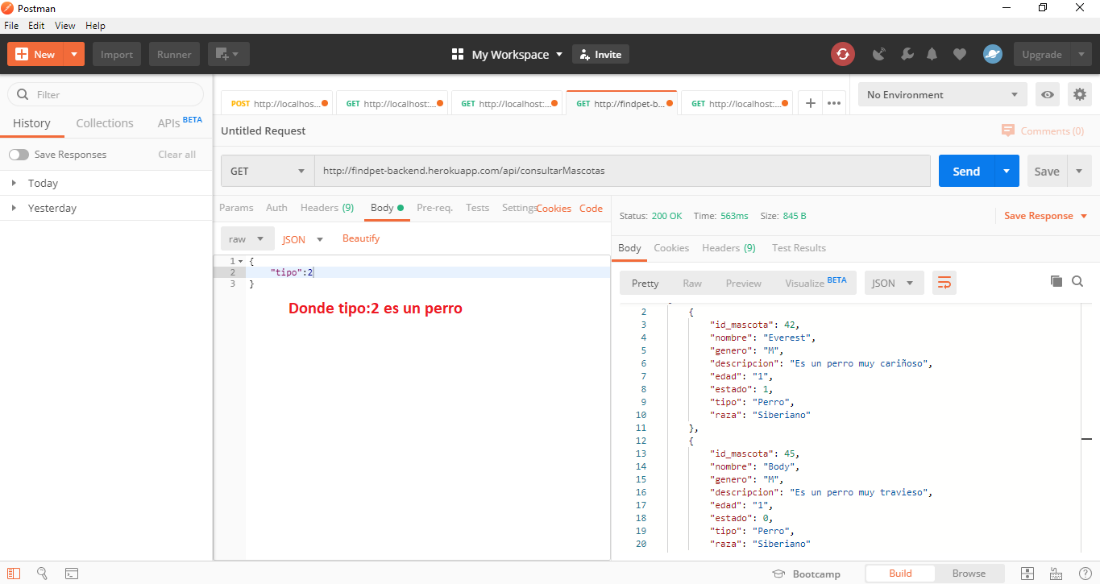
* **ConsultarPerdidas (GET)**

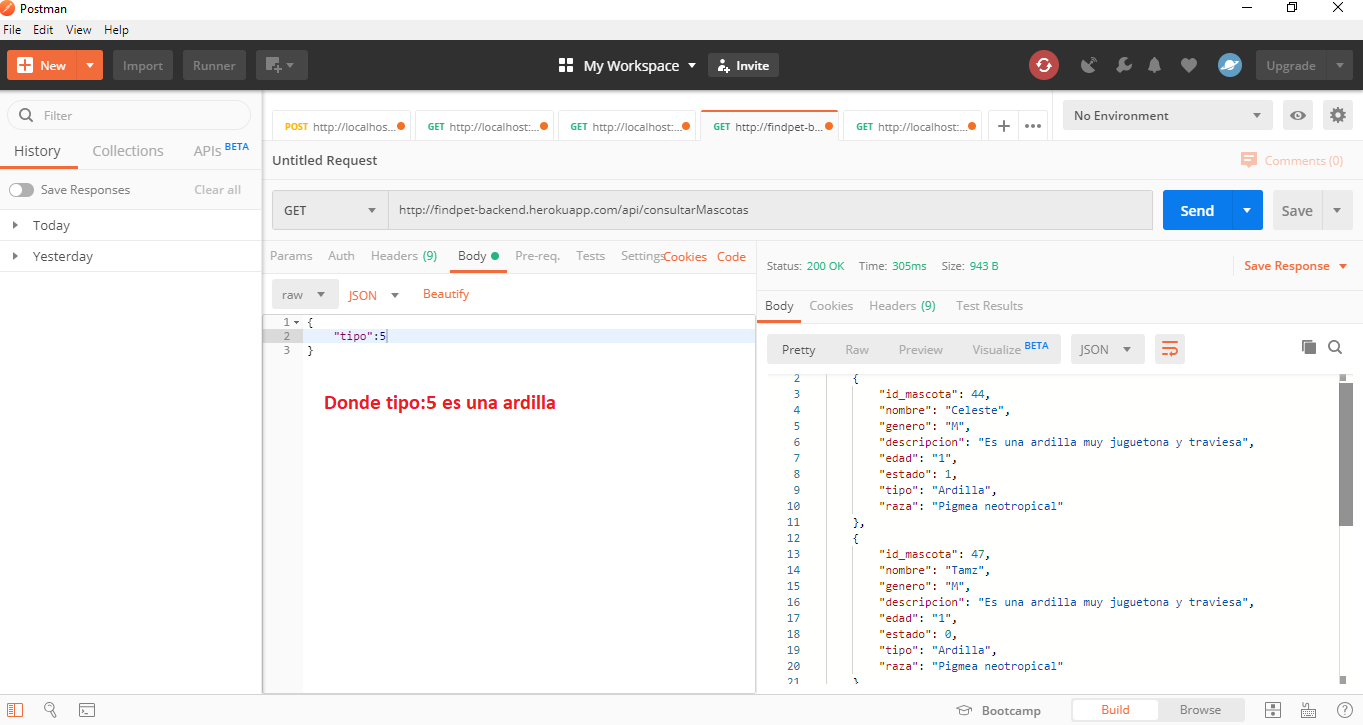
Este servicio no recibe ningún parámetro dentro del body del request al ser un método GET y retorna todas las mascotas que posean un “estado” con valor de 0.



* **ConsultarMascotas(GET)**

Este servicio permite consultar las mascotas por tipo, donde el body del request recibe el tipo de mascota a consultar, donde por el momento se definió que “2” corresponde a “perros”, “4” corresponde a “gallinas” y “5” corresponde a “ardillas”. Con lo cual nos retornará todas las mascotas de un dueño.





* **AllMascotas(GET)**

Este servicio fue creado con fines utilitarios, ya que permite evidenciar los cambios realizados a los registros ya existentes y así verificar el correcto funcionamiento de los demás servicios, por el momento no se tiene pensado usarlo dentro del App Móvil.

En los servicios anteriores ya se presentaron capturas de este servicio, pero de ser necesario puede ingresar al siguiente link y verificar su funcionamiento.

**Link: https://findpet-backend.herokuapp.com/api/allmascotas**