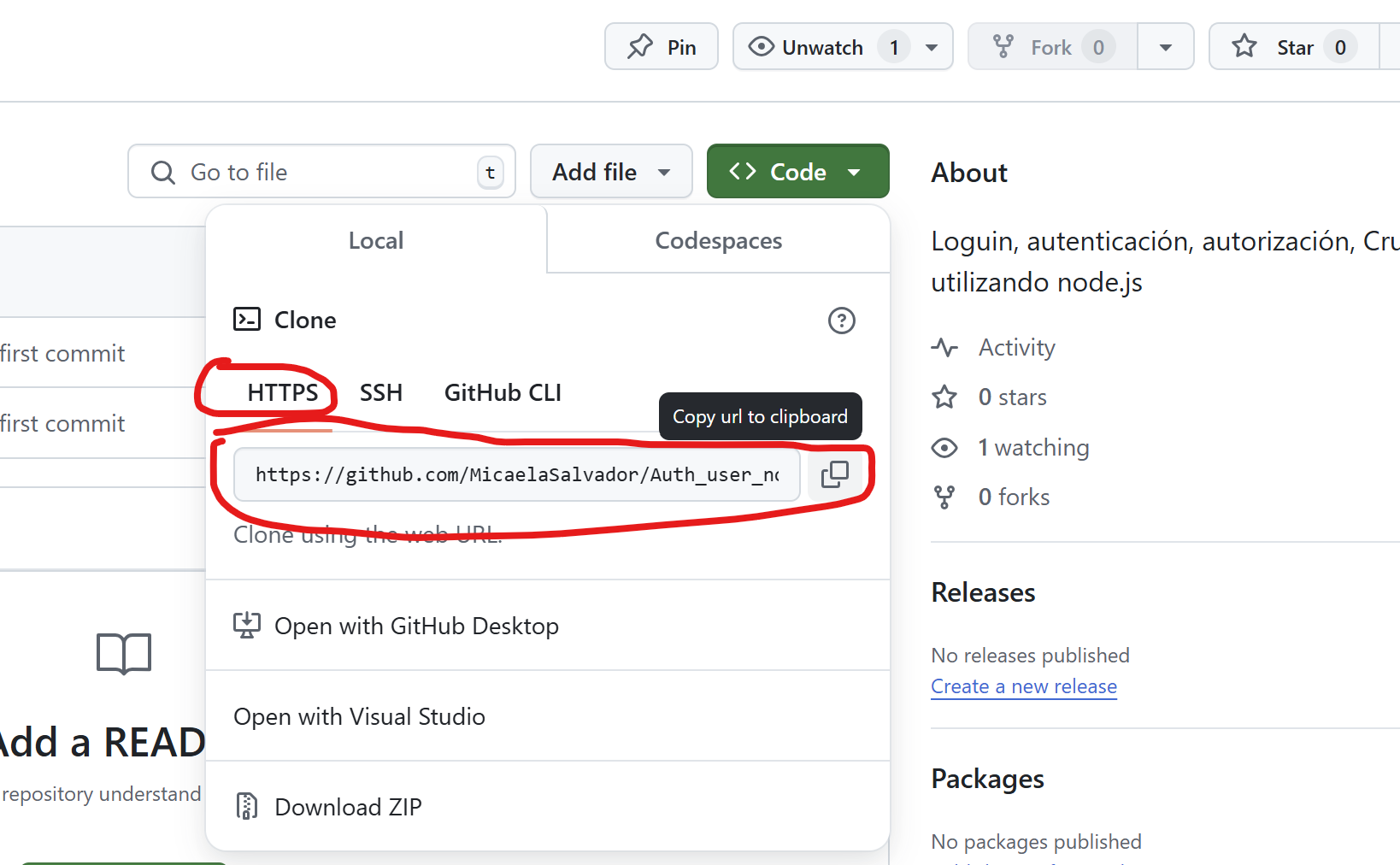
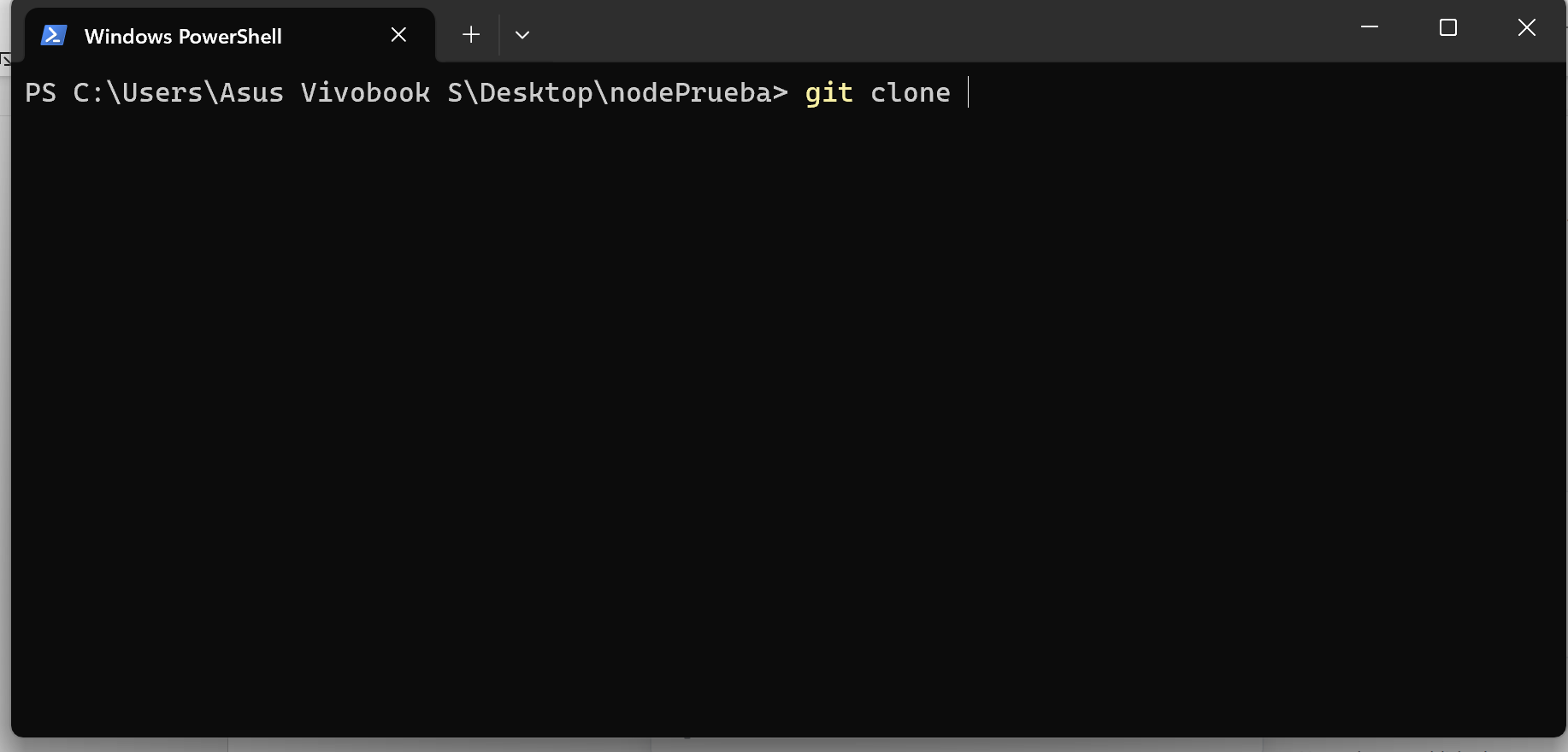
**BACKEND AUTENTICACION USUARIOS EN NODE.JS**

1.\_Para poder ejecutar el proyecto authuser\_node es necesario descargarlo o clonarlo, debes de ir al siguiente link <https://github.com/MicaelaSalvador/Auth_user_node> veras una carpeta el cual contiene el proyecto authuser, un script de la base de datos, y este manual que incluye los pasos para correr el proyecto, hasta la ejecución de los endpoints.

2.\_ Para clonarlo, ya teniendo abierto el link del proyecto de GitHub debes ir a code y copiar el link como se muestra en la imagen.



Después debes de abrir una pestaña de Windows PowerShell con la ubicación donde deseas guardar el proyecto, en este caso yo elegí el escritorio y cree una carpeta llamada nodePrueba , a continuación tienes que poner el siguiente comando git clone y el link del proyecto de GitHub que copiamos anteriormente como se muestra en la imagen .



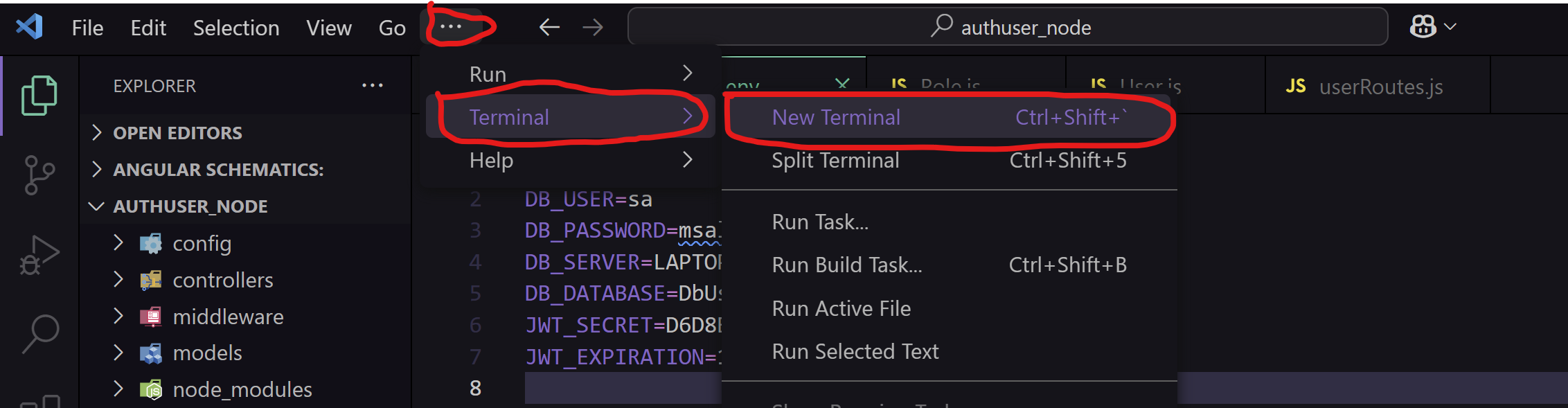
Siguiendo estos pasos te aparecerán todas las carpetas que se encuentran almacenas en github.

3.\_Siguiente paso debes abrir management studio de SQL Server e iniciar sesión con tus credenciales, posteriormente ir a la carpeta del proyecto y abrimos el script DbUser\_auth, copiamos su contenido, creamos una nueva consulta en Management Studio y pegamos el contenido del script y ejecutamos, con esto se creará la base de datos DbUser\_auth y las tablas correspondientes.

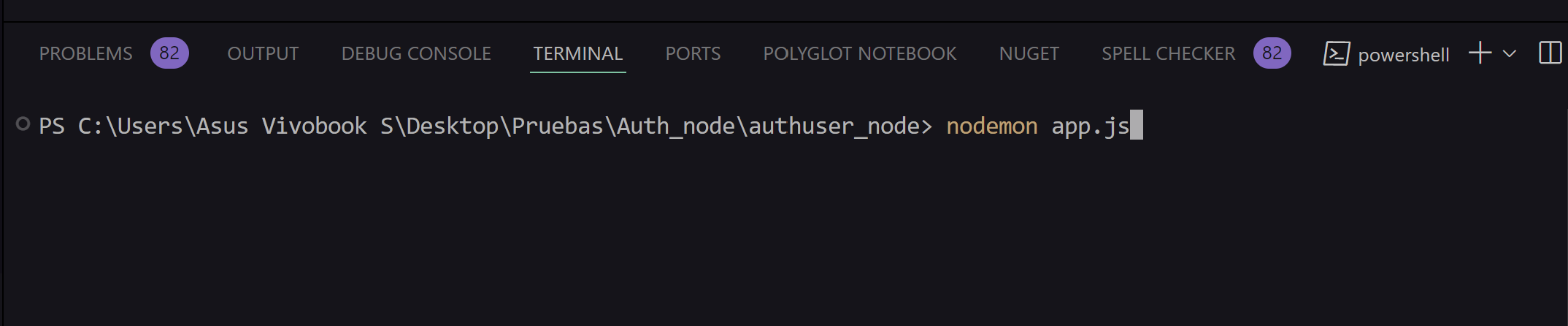
5.\_ Ya después de haber descargado y descomprimido o clonado el proyecto y hacer los pasos anteriores, debes de ir a la carpeta del proyecto localizar la carpeta authuser\_node , abrir Visual studio code(VSC) y arrastrar la carpeta authuser\_node a visual studio code.

6.\_ Posteriormente dentro del proyecto que abriste en visual studio code debe localizar. env debes abrirlo y configurar el archivo de acuerdo a la configuración de tu SQL Server, esto te permite conectar el web api con SQL Server.

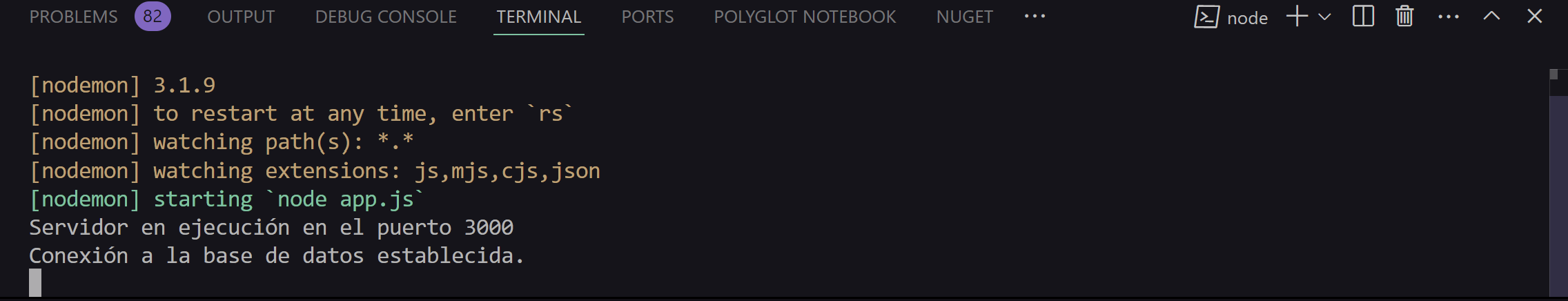
7-Para ejecutar el proyecto debes abrir una nueva terminal en Visual studio code como se muestra en la imagen siguiente.



Ya que tenemos la nueva terminal debemos poner la siguiente línea de código.



En seguida te mostrará que el proyecto está corriendo en el puerto localhost:3000 y que está conectado a la base de datos como se muestra en el ejemplo.



Listo proyecto ejecutándose perfectamente, sigue los siguientes pasos para ejecutar los endpoint que se detallan a continuación .

En todos los ejemplos que se muestran en seguida se utilizó Postman para realizar pruebas de Register, login y CRUD en la tabla Usuarios, también se incluye ejemplos de CRUD de la tabla Roles de la base de datos llamada DbUser\_auth utilizando el SGBD **Microsoft SQL Server**.

1.-**Ejemplo para registrarse**

POST: <http://localhost:3000/api/auth/register>

{

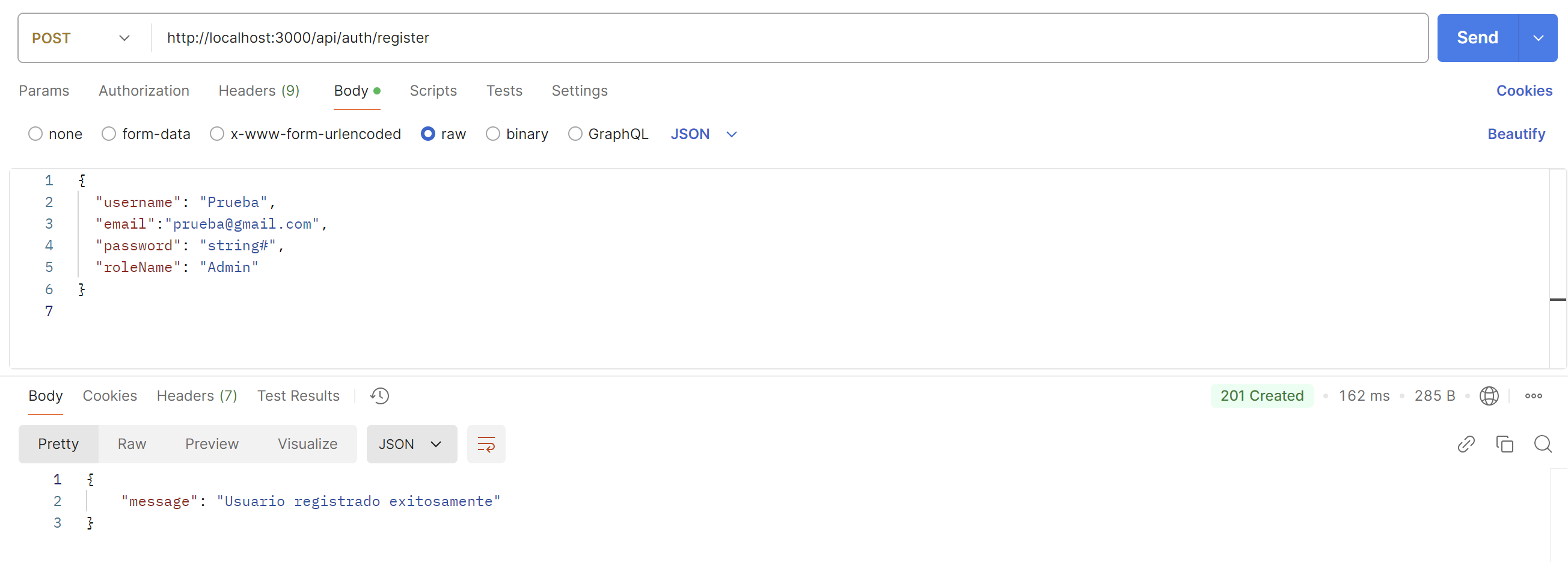
"username": "Prueba",

"email":"prueba@gmail.com",

"password": "string#",

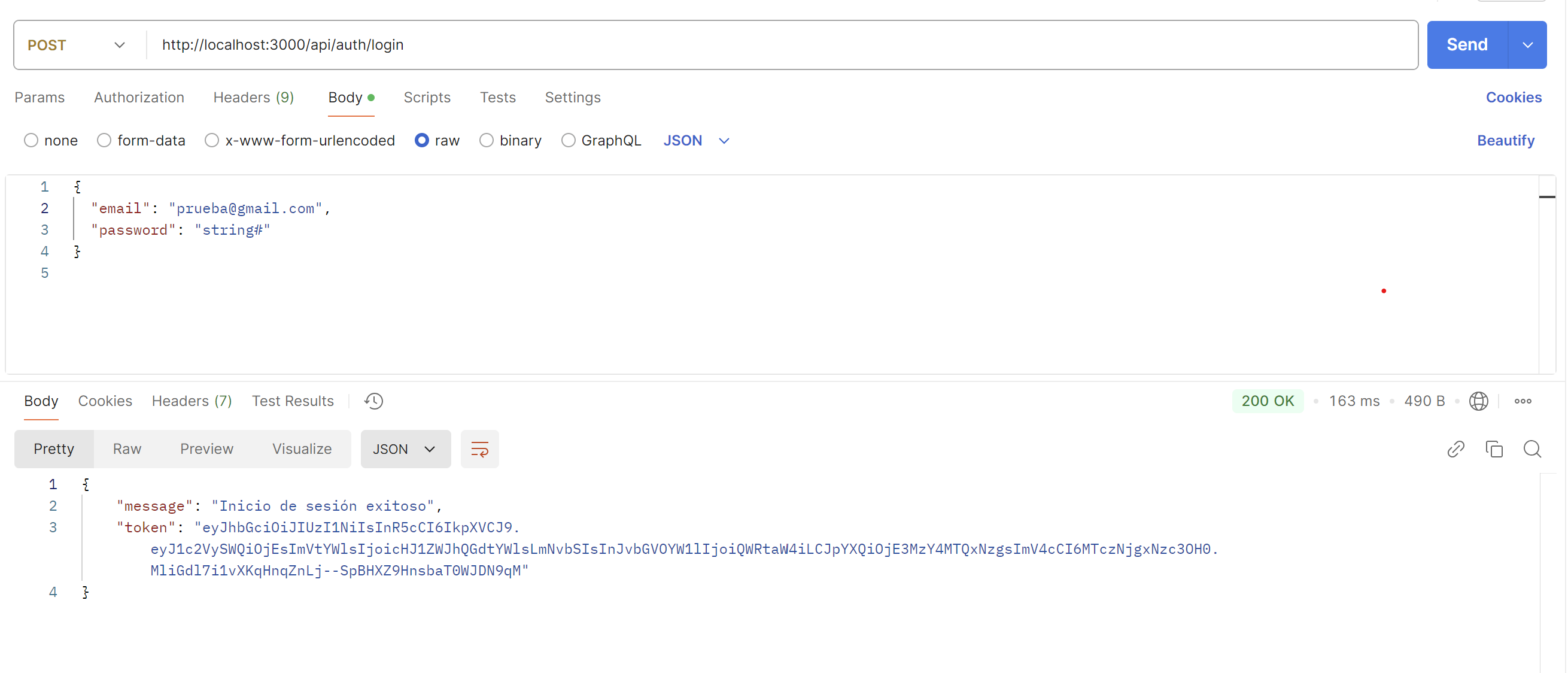
"roleName": "Admin"

}



**2.-Ejemplo de Login**

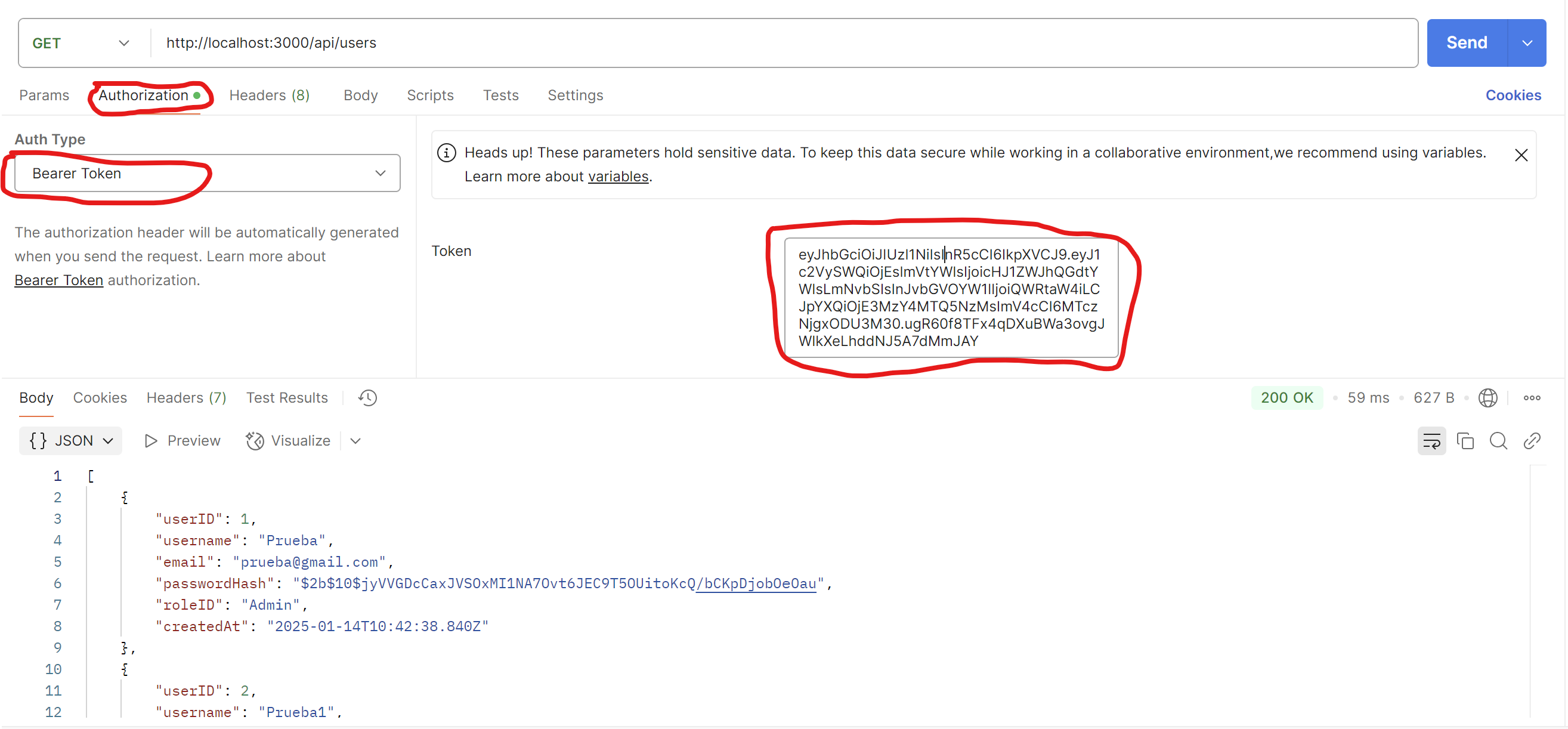
POST: <http://localhost:3000/api/auth/login>



**3.\_Obtener todos los registros de los Usuarios.**

Para obtener todos los registros de los Usuarios es necesario copiar el token que nos generó el Loguin como se ve en la imagen siguiente.

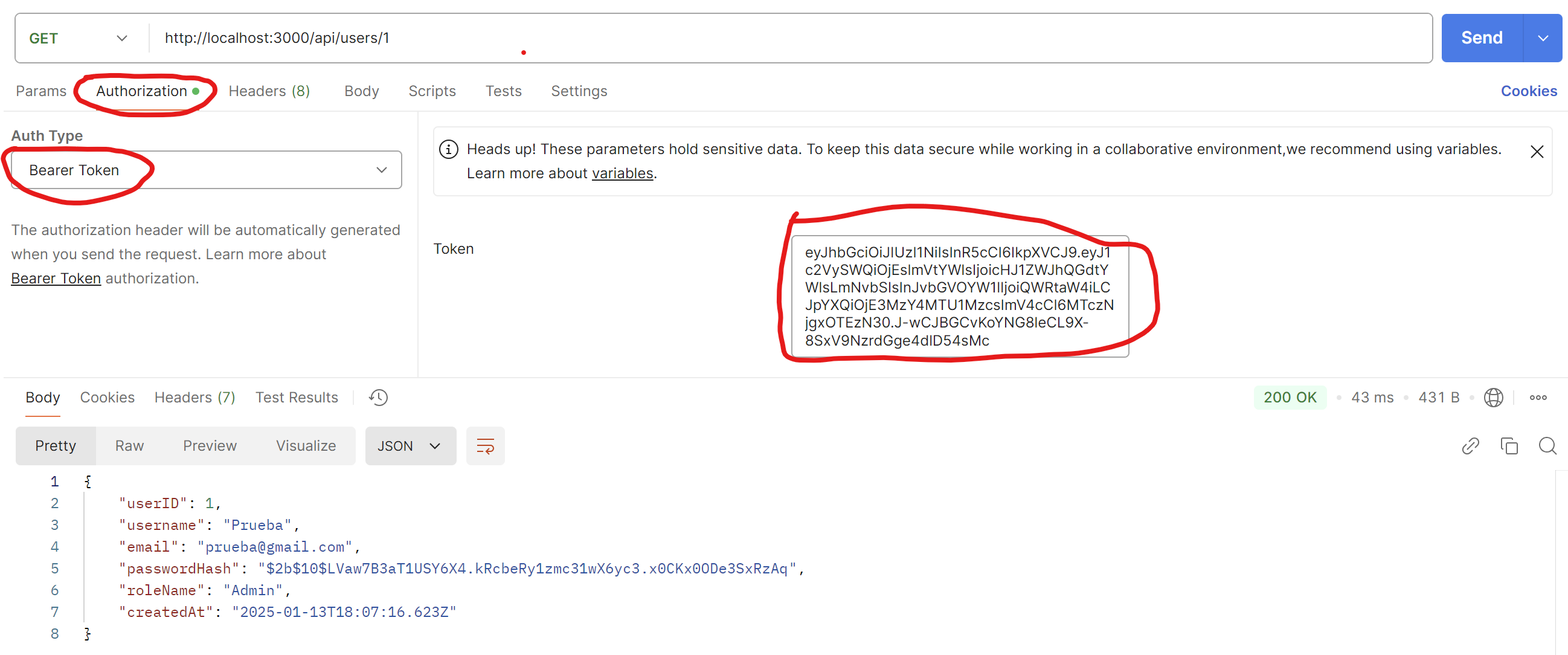
GET: <http://localhost:3000/api/users>



**4.\_obtener Usuarios por ID**

Para obtener un usuario por ID(getById) es necesario también copiar el token de la misma manera que Obtener todos los usuarios como se muestra en la imagen siguiente.

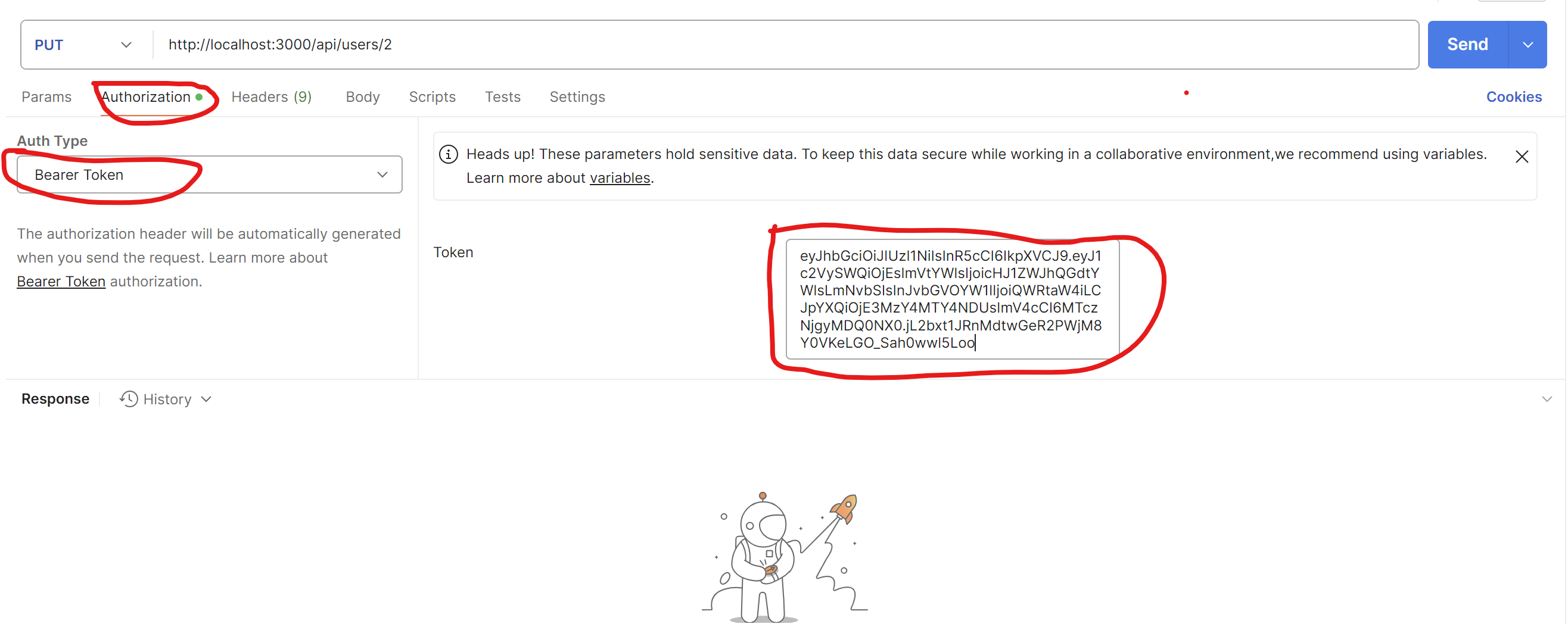
GET: <http://localhost:3000/api/users/1>

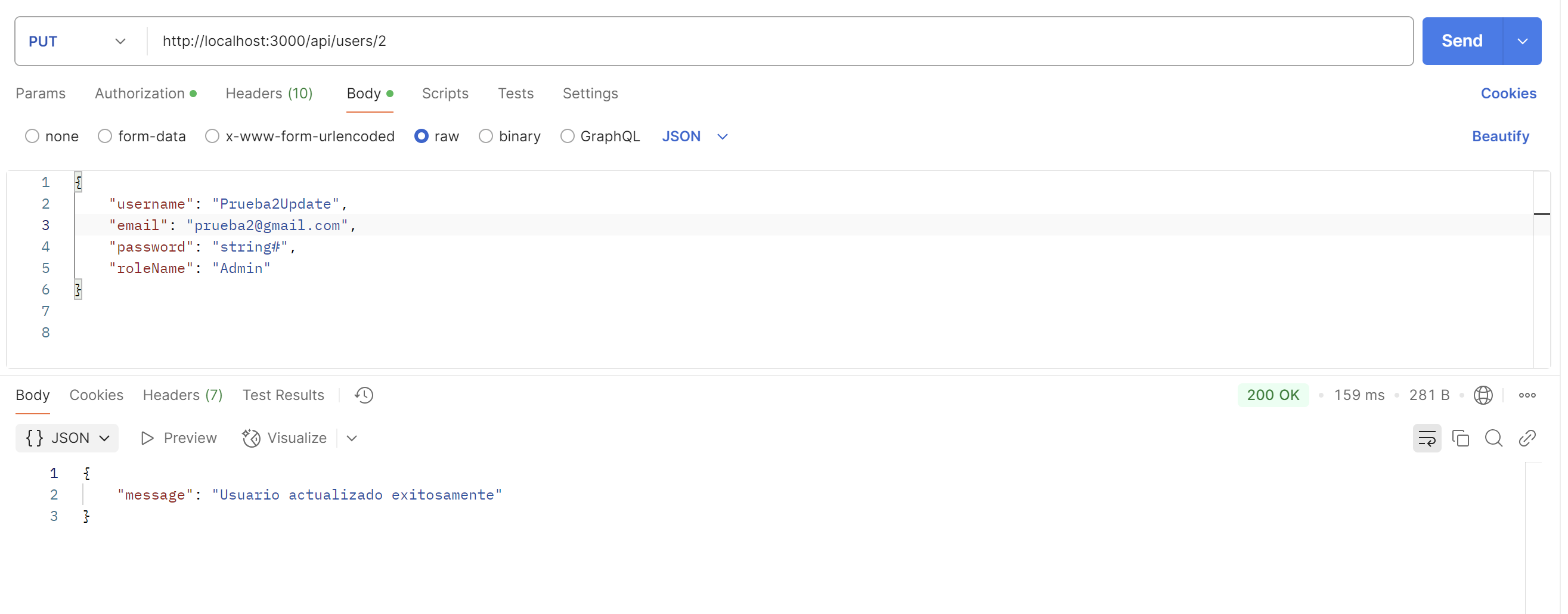


**4.\_ Actualizar Usuario**

Para actualizar también tenemos que poner el token indicando que somos Admin y por lo tanto podemos hacer modificaciones.

PUT: <http://localhost:3000/api/users/>2

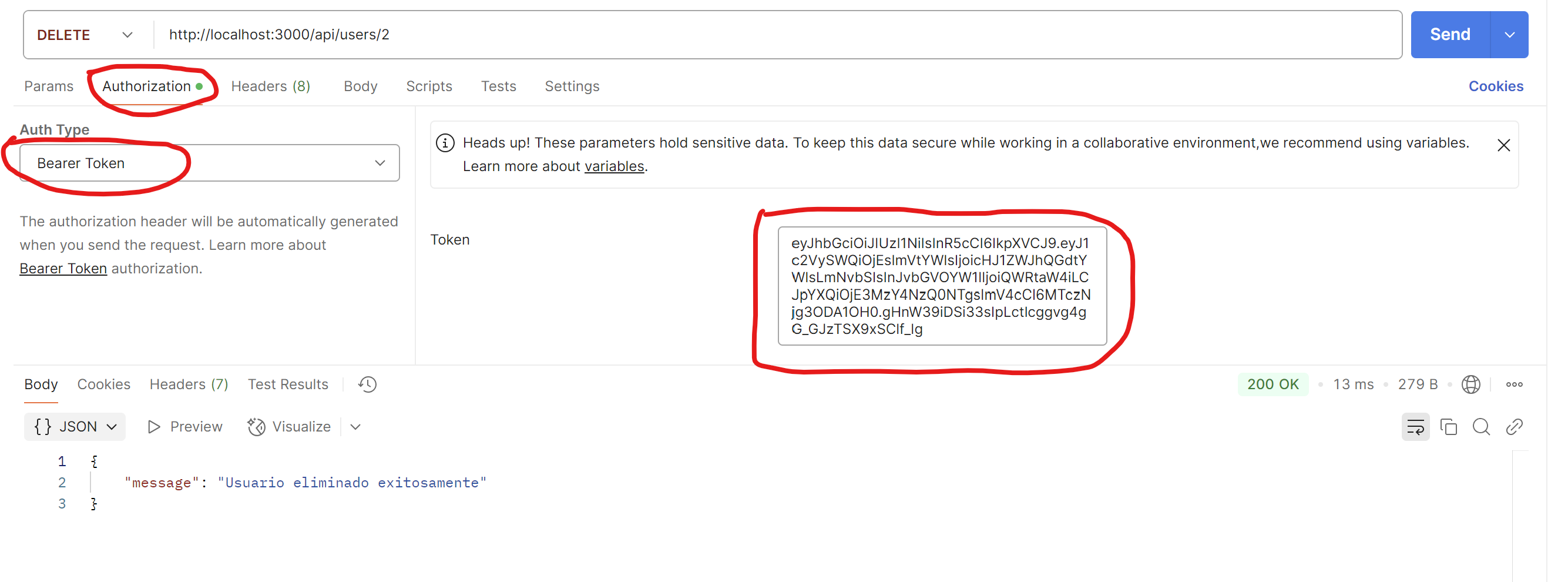




**5.\_Eliminar Usuarios**

Para eliminar un usuario también es necesario copiar el token del Loguin en Authorization ya que no cualquier usuario tiene permitido eliminar un registro en este caso solo un Admin.

DELETE: <http://localhost:3000/api/users/2>

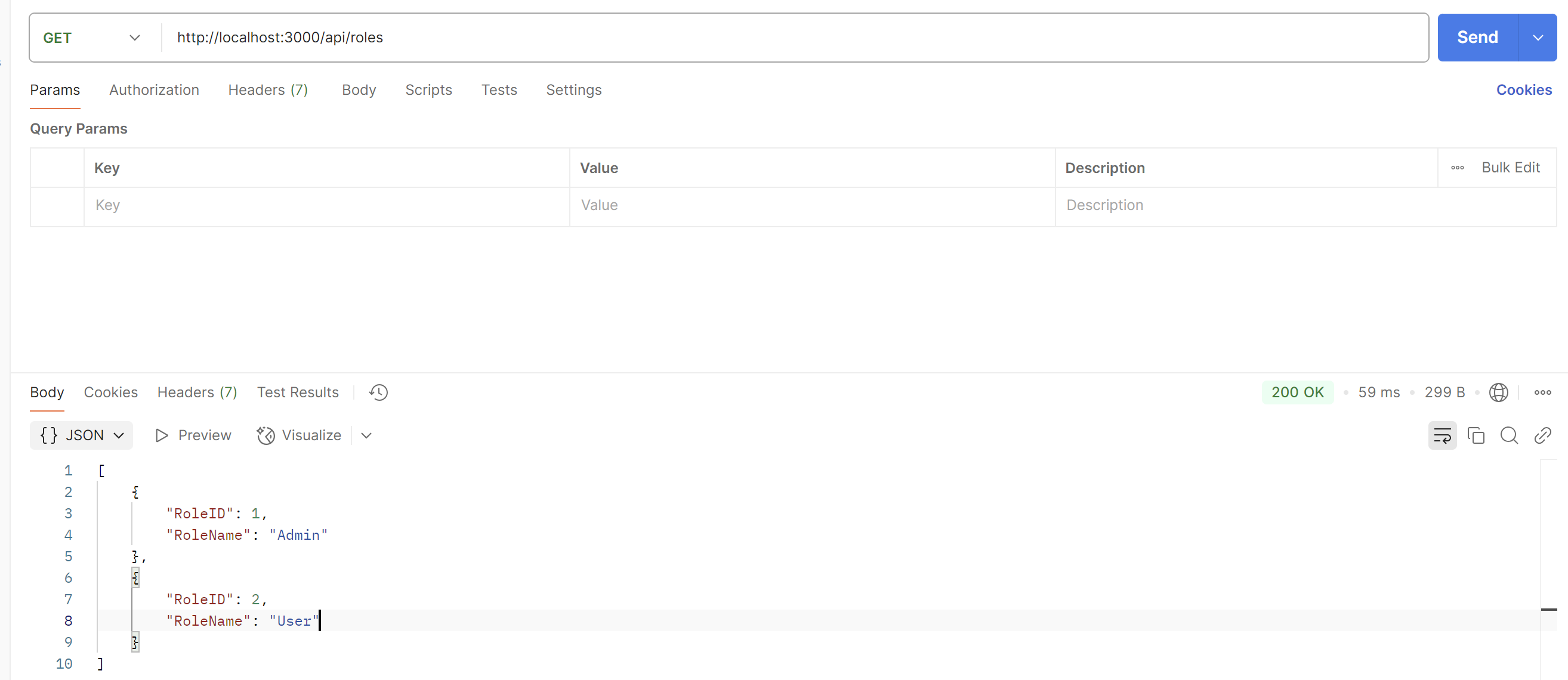


**EJEMPLO DE COMO EJECUTAR LOS ENDPOINT PARA ROLES(CRUD)**

**1.\_Obtener todos los roles.**

Para obtener todos los Roles no necesitamos ningún token, ya está permitido tanto Admin como User poder acceder a la información.

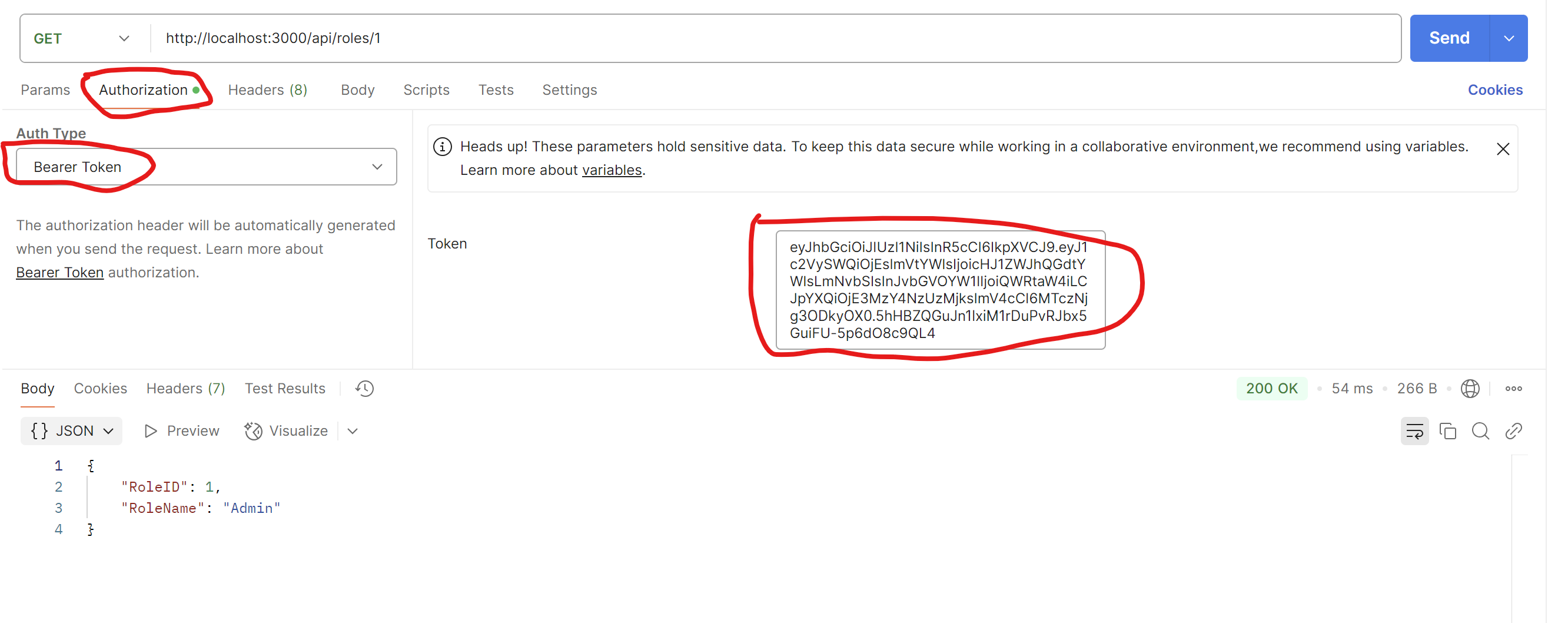
GET: <http://localhost:3000/api/roles>



**2.\_Obtener Roles por ID**

Para obtener los roles por Id(getById) es necesario agregar el token que nos devuelve el login ya que solo está permitido ver esa información a Usuarios con el Rol Admin.

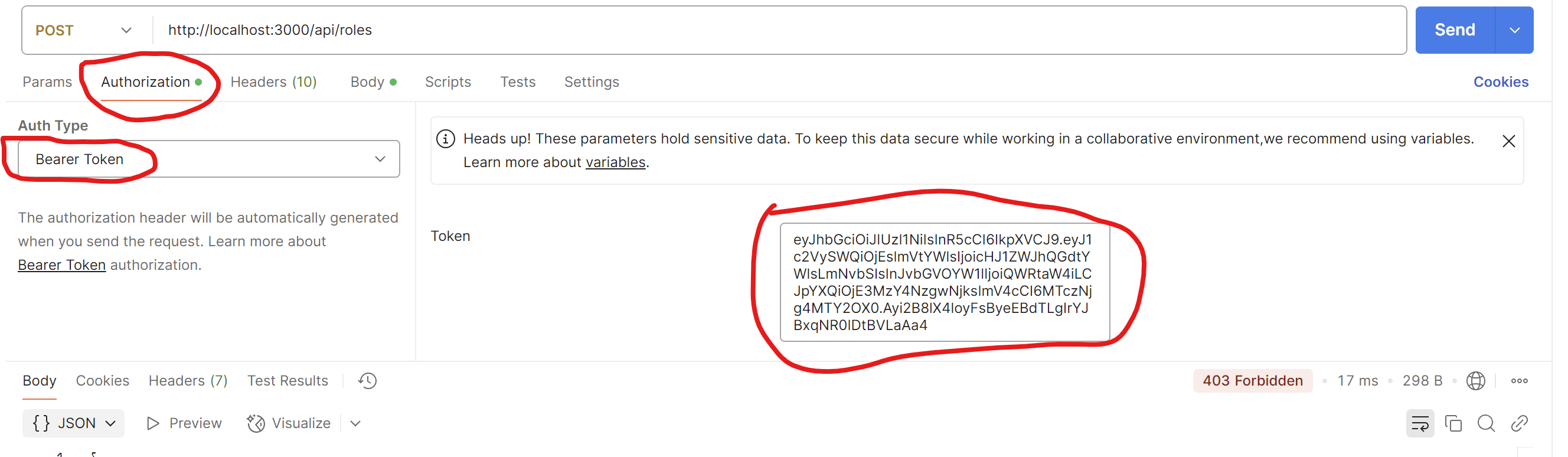
GET: <http://localhost:3000/api/roles/1>

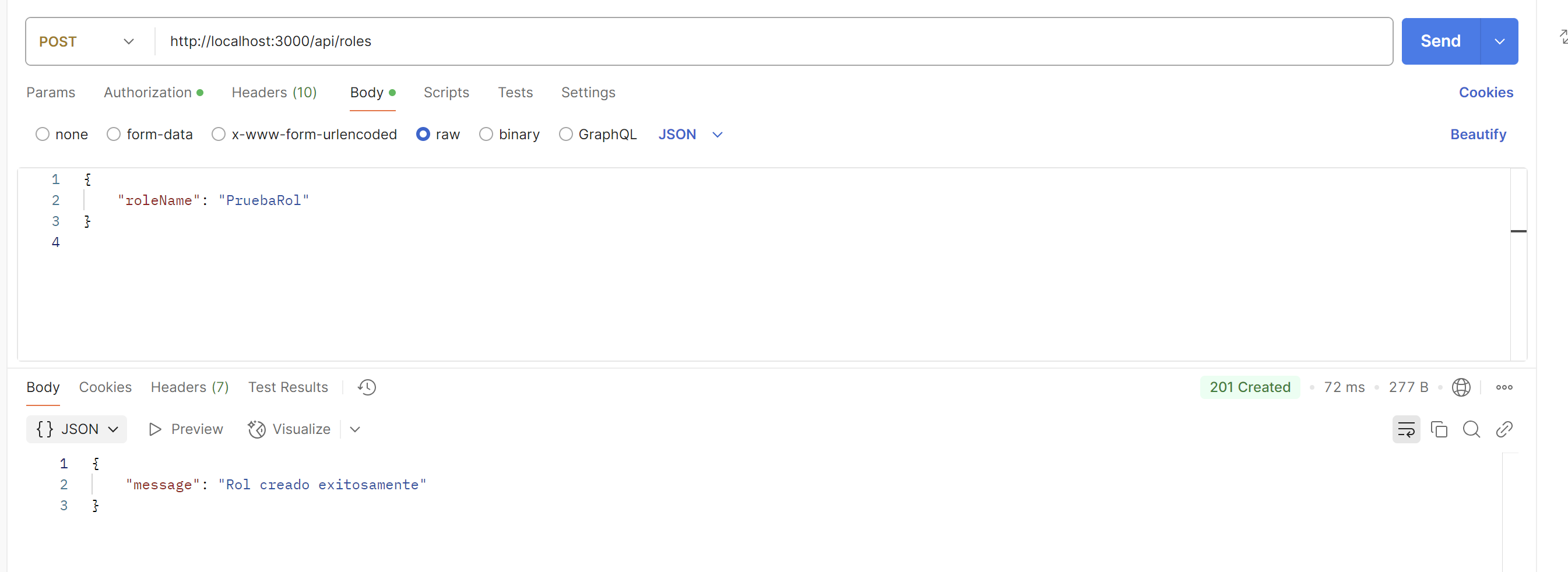


**3.\_Crear Rol**

Para crear un Rol también es necesario poner o agregar el token que nos devuelve el login.

POST: <http://localhost:3000/api/roles>

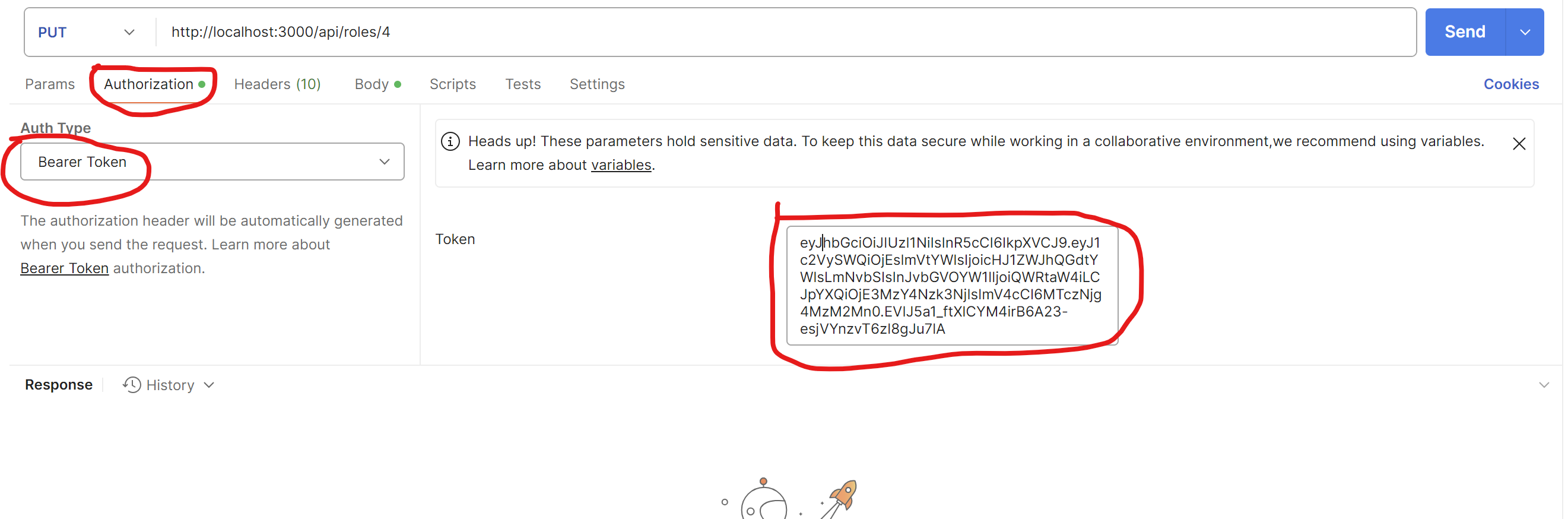


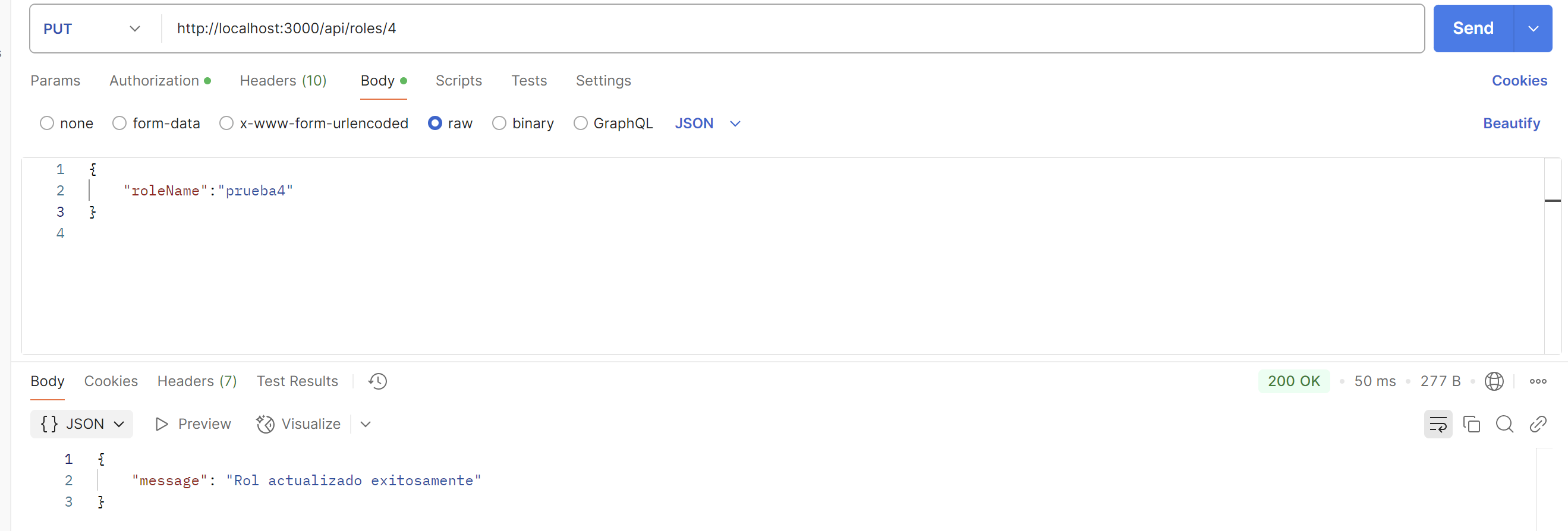


**4.\_ Actualizar Rol**

Para actualizar también es necesario poner el token que nos devuelve el login.

PUT: http://localhost:3000/api/roles/4





**5.\_Eliminar Rol**

Para eliminar un Rol solo lo puede hacer un Usuario con Rol Admin, por eso es necesario poner el token que nos devuelve el login de la siguiente manera.

DELETE: <http://localhost:3000/api/roles/4>

