FRANCISCO MANUEL CHAN COLLI

**1. SQL STATEMENT**

Consulta

select

re.name,b.title,ra.rating,ra.rating\_date

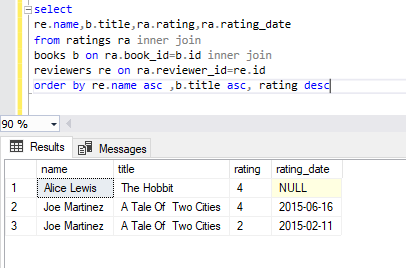
from ratings ra inner join

books b on ra.book\_id=b.id inner join

reviewers re on ra.reviewer\_id=re.id

order by re.name asc ,b.title asc, rating desc

Resultado de consulta:



**2. Programming Test**

Opte por importar la base de datos a SQL server para ya luego, poder hacer las peticiones get, post, put, delete. Mi idea era una vez exportada la base de datos genero procedimientos almacenados en SQLServer y los llamo desde mi api en nodejs y ahí estarían todas mis funciones.

**IMPORTANTE:**

**1.Ejecute el script(ScriptDatabase.sql)**

**2.Abrir una terminal y escribes npm install dentro de la ruta del proyecto para instalar las dependencias.**

**3.Ejecuta el proyecto, escriba en la terminal (npm run start).**

**4.Listo**

**GET /contacts**

**Ejemplo:**

[**http://localhost:3888/api/contacts**](http://localhost:3888/api/contacts)

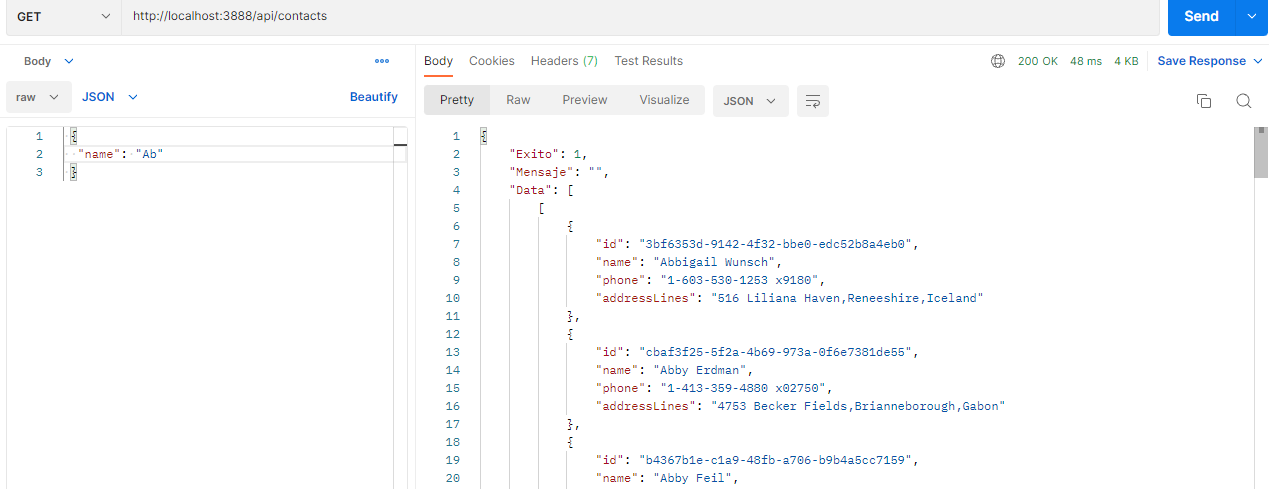
**Desde el Body**

{

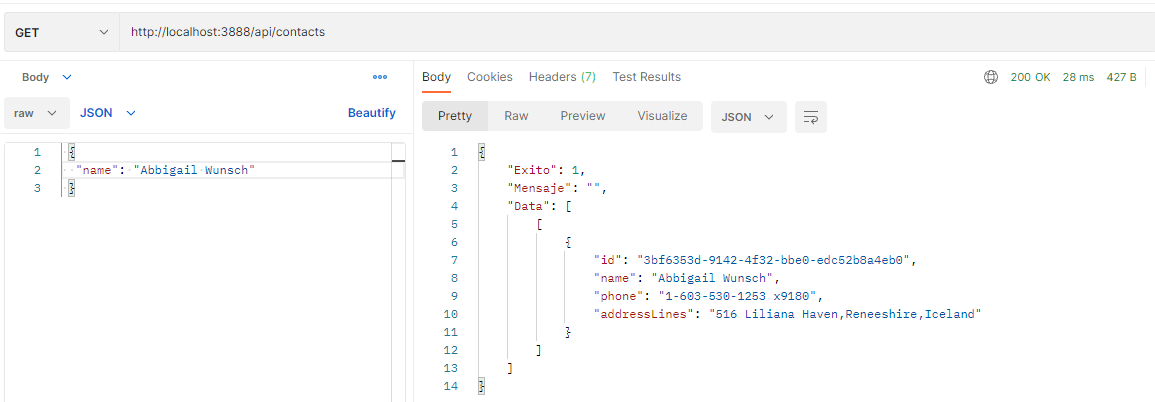
  "name": "Abbigail Wunsch"

 }

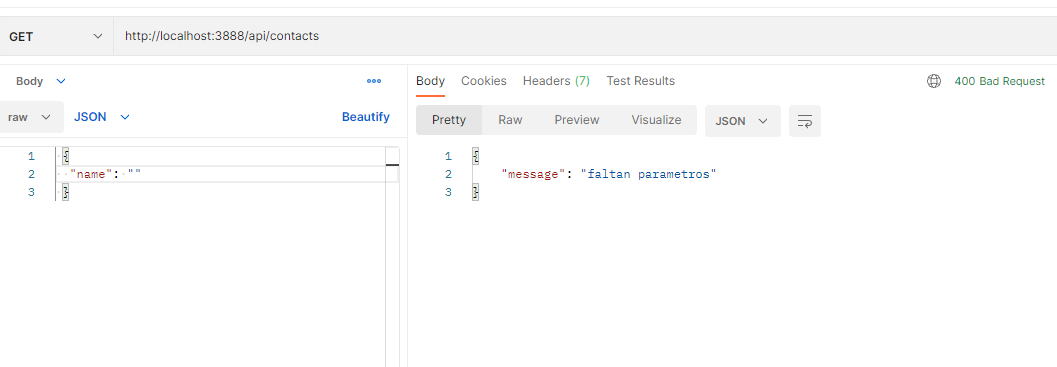
**Búsqueda por nombre filtrando por minúsculas o mayúsculas**



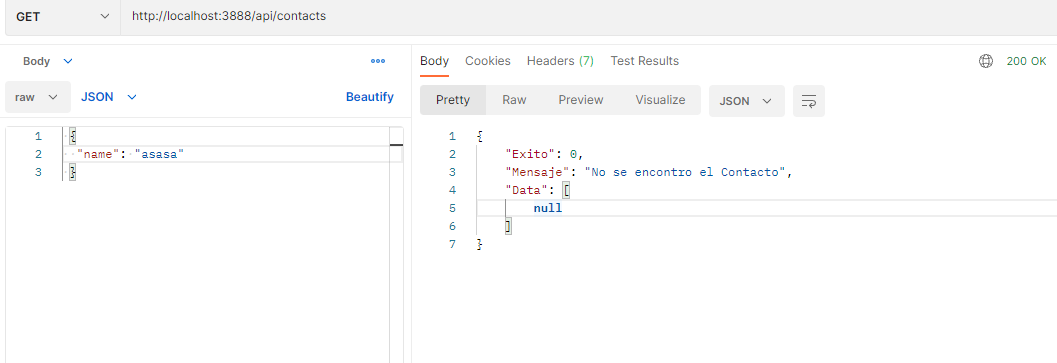
**Mostrar datos especifico**



**Si el parámetro no es enviado envía error 400**



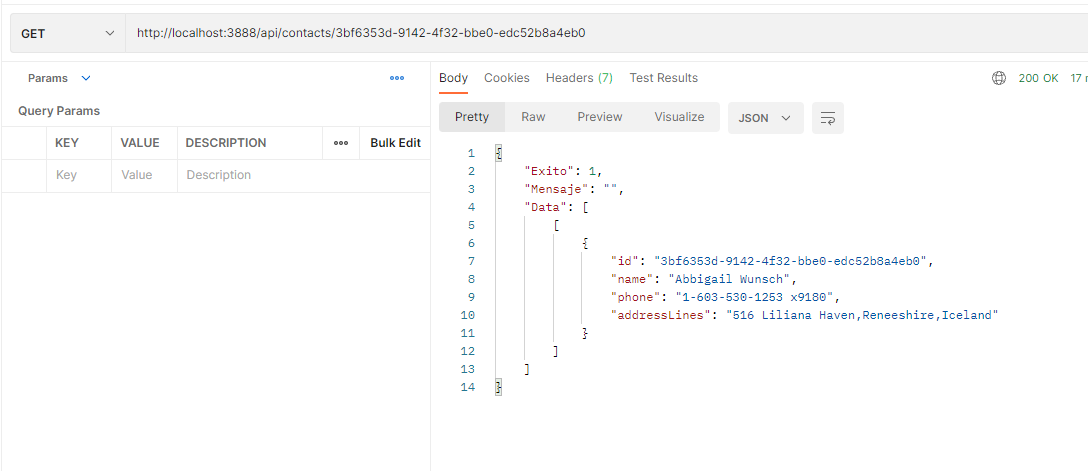
**Si el contacto no existe**

****

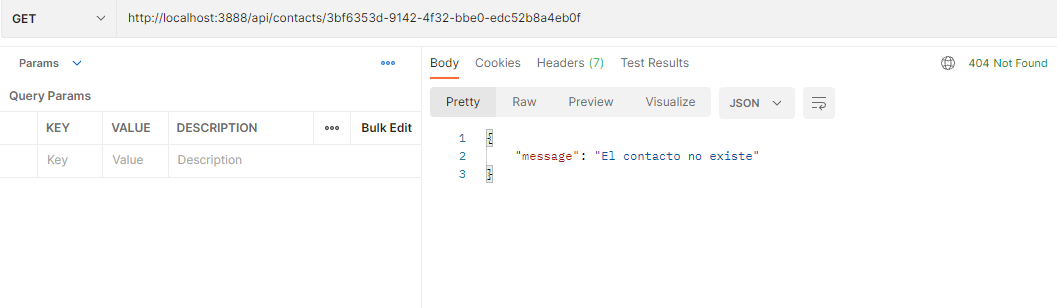
**GET /contacts/**

**Ejmplo:** http://localhost:3888/api/contacts/3bf6353d-9142-4f32-bbe0-edc52b8a4eb0f

**Si existe el id del contacto**



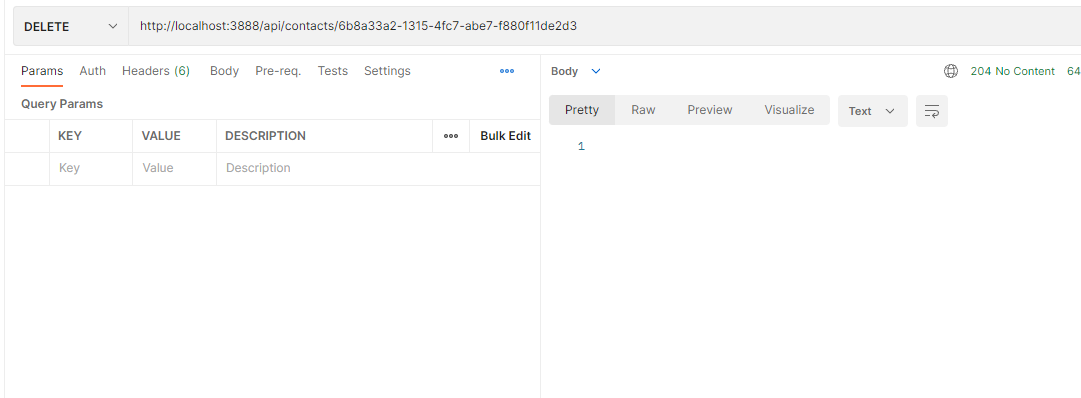
**Si No existe el id envía error 404**



**DELETE /contacts/**

**Ejemplo:** **http://localhost:3888/api/contacts/**6b8a33a2-1315-4fc7-abe7-f880f11de2d3

**Si el id del contacto existe**

****

**Si el id del contacto no existe**

