

Manual Usuario

API EventosDeportivos

1. Tecnologías

- **Para tener en cuenta:**

Debemos tener actualizados los repositorios e instalado pip:

- `sudo apt-get update`
- `sudo apt-get upgrade`
- `sudo apt-get install python3-pip`

- **Instalar Python**

- `sudo apt-get install python3`

- **Instalar sqlite3**

- `sudo apt-get install sqlite3`

- **Instalar Flask**

- `sudo apt-get install python3-flask`

- **Instalar SQLAlchemy**

- `sudo pip install flask`

- **Volcar base de datos**

- `sqlite3 eventosDeportivos.db < EventosDeportivos.sql`

El path de eventosdeportivos.db hay que indicarlo en el archivo `__init__.py`

- **Rellenar tablas:**

- Una vez tengamos todo configurado, ejecutaremos el fichero `addData.py` para rellenar las tablas (excepto la tabla 'participan', que la rellenaremos manualmente).

2. EndPoint

En este apartado vamos a ver cuáles son nuestros endpoint:

➤ EVENTOS

```

localhost:5000/eventos

GET localhost:5000/eventos

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ⌵

1 {
2   {
3     "actividad": "okeyssssssssssssss",
4     "ano": 2022,
5     "deporte": "JUDO",
6     "entidad": "nombreprueba",
7     "instalacion": "PABELLON C.D. PRINCIPE FELIPE",
8     "mes": "FEBRERO"
9   },
10  {
11    "actividad": "Eventos Deportivos Especiales",
12    "ano": 2022,
13    "deporte": "ATLETISMO",
14    "entidad": "CLUB ATLETISMO ARGANDA",
15    "instalacion": "PISTA DE ATLETISMO. ESTADIO MUNICIPAL DE DEPORTES.",
16    "mes": "FEBRERO"
17  },
18  {
19    "actividad": "Eventos Deportivos Especiales",
20    "ano": 2022,
21    "deporte": "PATINAJE",
22    "entidad": "CLUB PATINAJE ARGANDA",
23    "instalacion": "PATINODROMO CIUDAD DEPORTIVA",
24    "mes": "FEBRERO"
25  }
26 }
  
```

- **localhost:5000/eventos (GET)**

- Obtenemos un archivo JSON con el listado de todos los eventos. Utilizaremos el método de transferencia de datos GET.

localhost:5000/eventos/put

POST localhost:5000/eventos/put

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

⓪ none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> mes	FEBRERO	
<input checked="" type="checkbox"/> ano	2022	
<input checked="" type="checkbox"/> Instalacion	ESTADIO LOS CARMENES	
<input checked="" type="checkbox"/> IdEmpresa	2	
<input checked="" type="checkbox"/> actividad	Eventos Deportivos Reales	
<input checked="" type="checkbox"/> deporte	HOCKEY	

Body Cookies Headers (5) Test Results

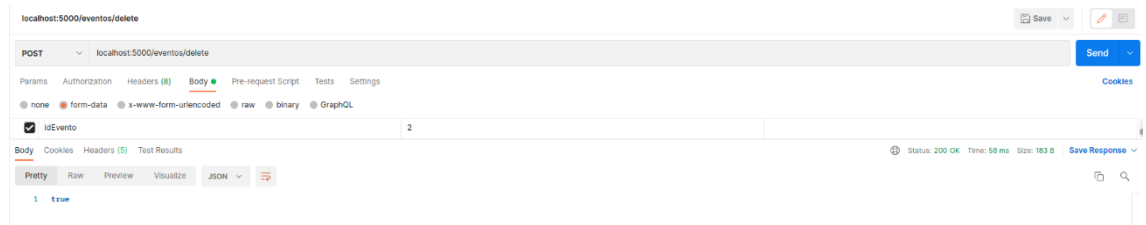
⓪ Status: 200 OK Time: 60 ms Size: 340 B Save Response

```

1 {
2   "actividad": "Eventos Deportivos Reales",
3   "ano": 2022,
4   "deporte": "HOCKEY",
5   "entidad": "CLUB ATLETISMO ARGANDA",
6   "instalacion": "ESTADIO LOS CARMENES",
7   "mes": "FEBRERO"
8 }
  
```

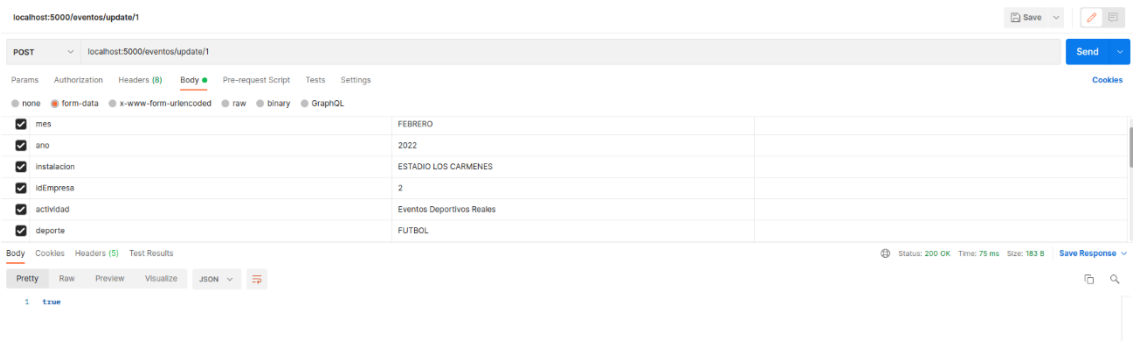
- **localhost:5000/eventos/put (POST)**

- Utilizaremos este endpoint para crear un nuevo evento en nuestra base de datos. Utilizaremos el método de transferencia de datos POST.



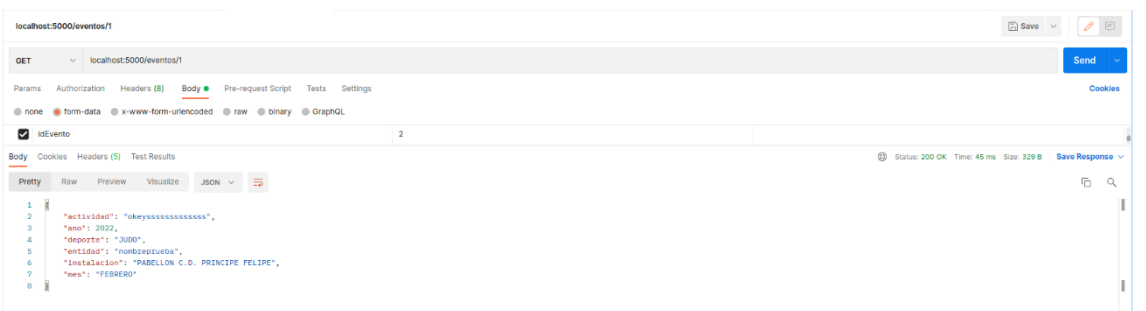
- **localhost:5000/eventos/delete/<eventoid> (POST)**

- A través de la id de un evento podremos eliminarlo. Utilizaremos el método de transferencia de datos POST.



- **localhost:5000/eventos/update/<eventoid> (POST)**

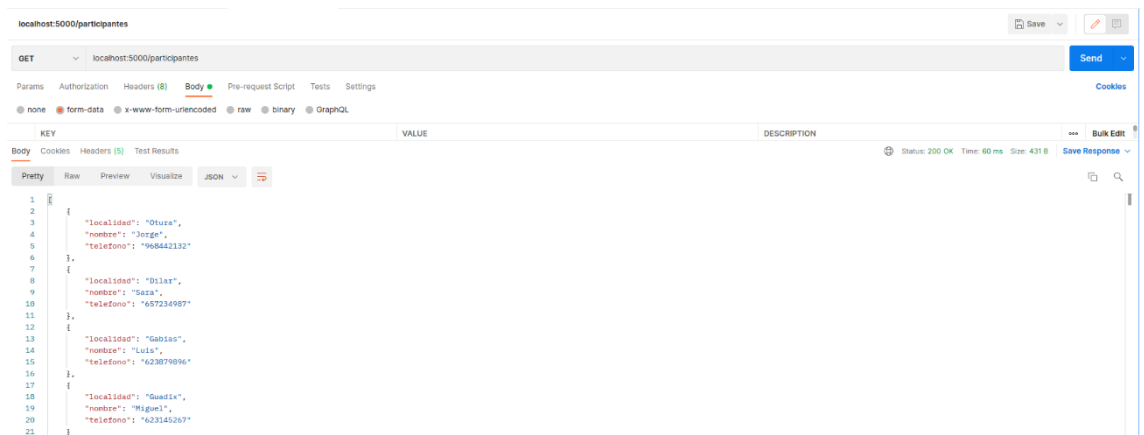
- Este endpoint nos permite actualizar los datos de un registro sobre un evento, siempre se deben de enviar todos los campos al servidor (en este caso mes, año, instalación, empresald, actividad, deporte). Utilizaremos el método de transferencia de datos POST.



- **localhost:5000/eventos/<eventoid> (GET)**

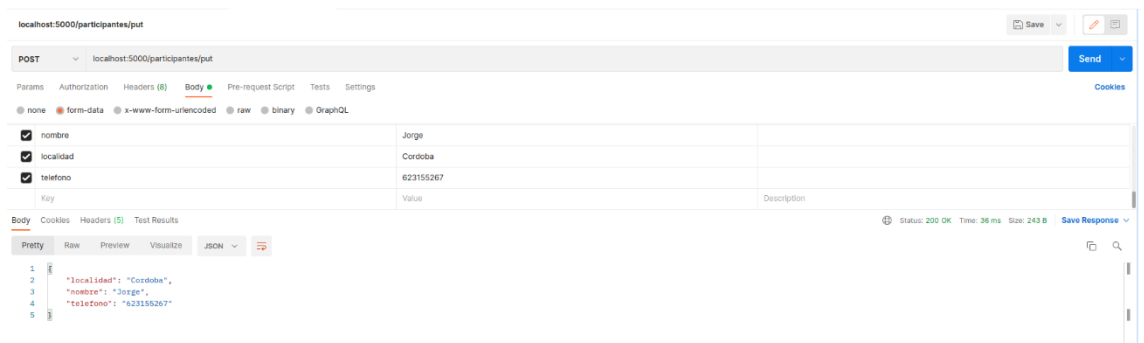
- Obtenemos un archivo JSON con la información del evento que hemos buscado a través de su id. Utilizaremos el método de transferencia de datos GET.

➤ PARTICIPANTES



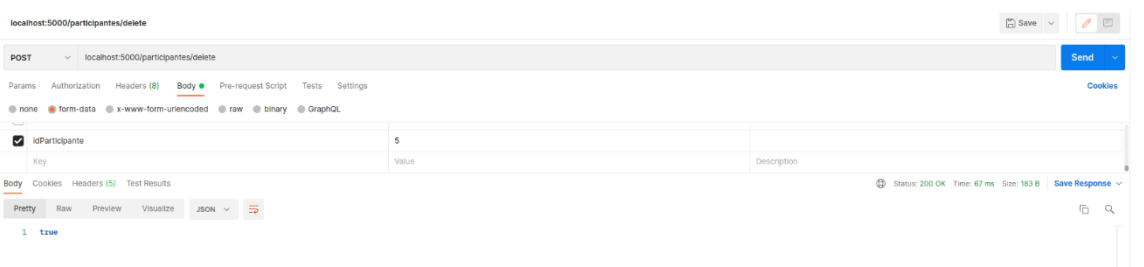
- **localhost:5000/participantes (GET)**

- Obtenemos un archivo JSON con el listado de todos los participantes. Utilizaremos el método de transferencia de datos GET.



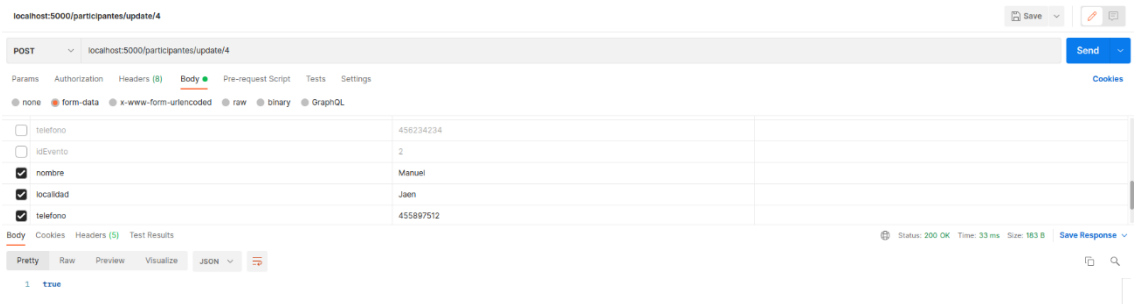
- **localhost:5000/participantes/put (POST)**

- Utilizaremos este endpoint para crear un nuevo participante en nuestra base de datos. Utilizaremos el método de transferencia de datos POST.



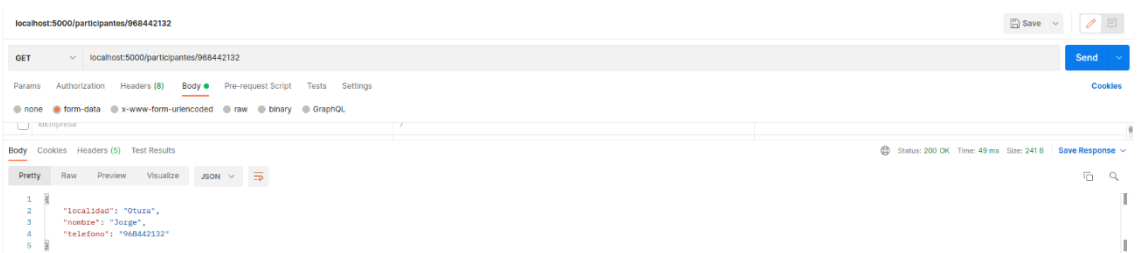
- **localhost:5000/participantes/delete/<participanteid> (POST)**

- A través de la id de un participante podremos eliminarlo. Utilizaremos el método de transferencia de datos POST.



- **localhost:5000/participantes/update/<participanteid> (POST)**

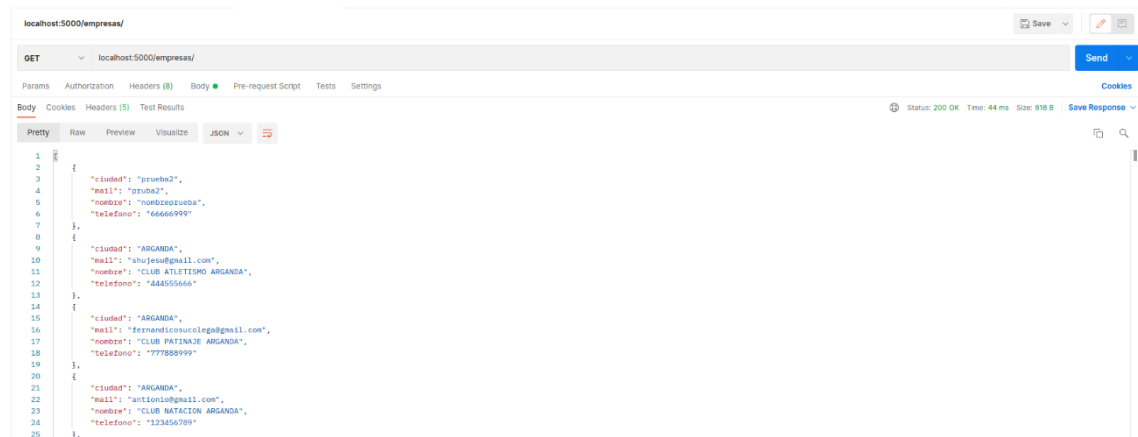
- Este endpoint nos permite actualizar los datos de un registro sobre un participante, siempre se deben de enviar todos los campos al servidor (en este caso nombre, localidad, teléfono). Utilizaremos el método de transferencia de datos POST.



- **localhost:5000/participantes/<telefono> (GET)**

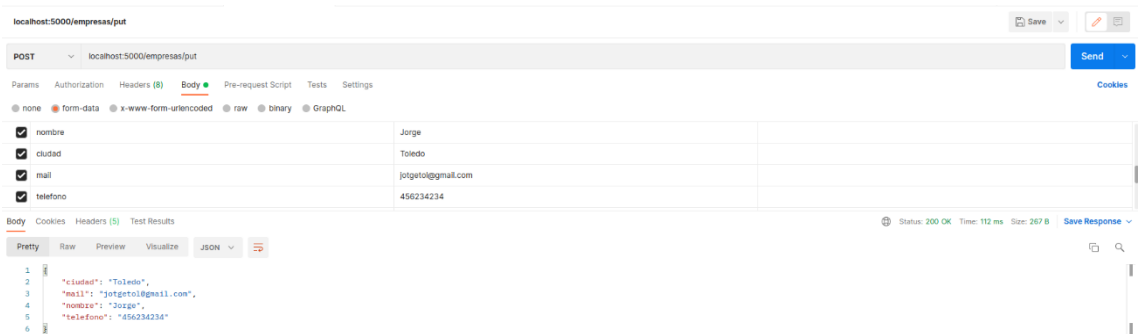
- Obtenemos un archivo JSON con la información del participante que hemos buscado a través de su teléfono. Utilizaremos el método de transferencia de datos GET.

➤ EMPRESAS



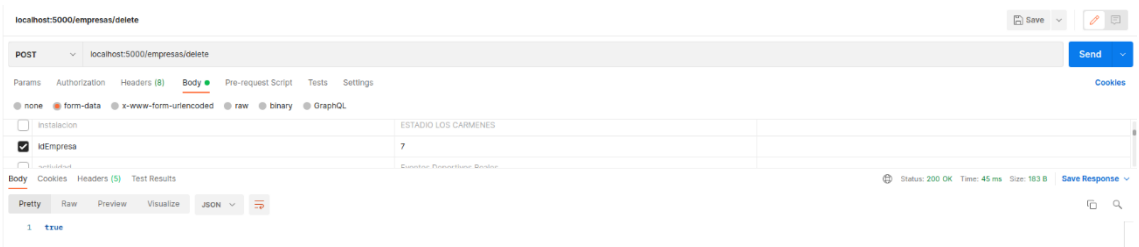
- **localhost:5000/empresas (GET)**

- Obtenemos un archivo JSON con el listado de todas las empresas.
- Utilizaremos el método de transferencia de datos GET.



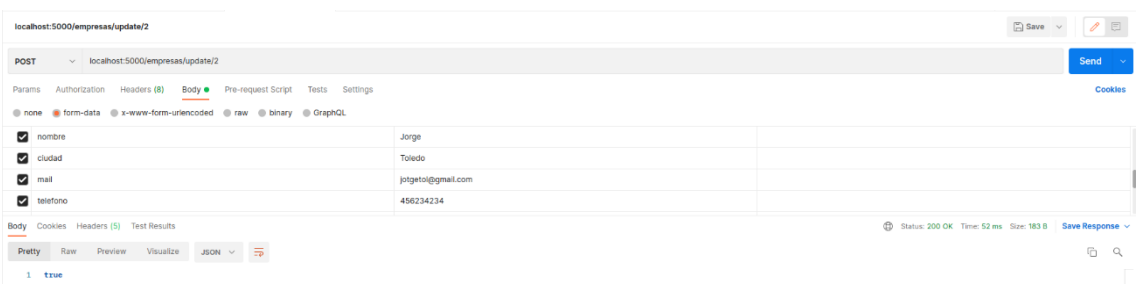
- **localhost:5000/empresas/put (POST)**

- Utilizaremos este endpoint para crear un nuevo participante en nuestra base de datos. Utilizaremos el método de transferencia de datos POST.



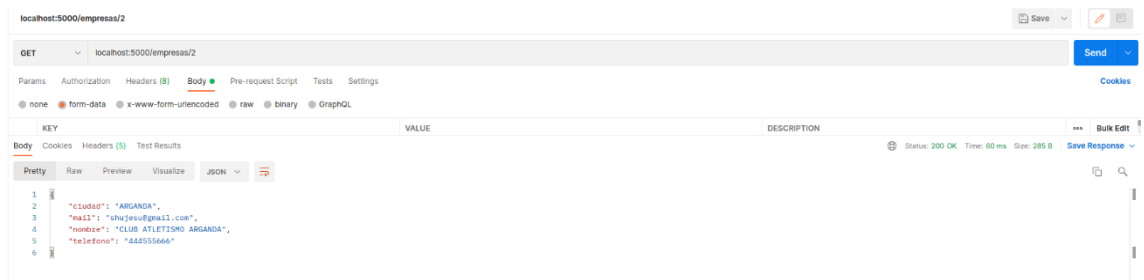
- **localhost:5000/empresas/delete/<empresasid> (POST)**

- A través de la id de una empresa podremos eliminarlo. Utilizaremos el método de transferencia de datos POST.



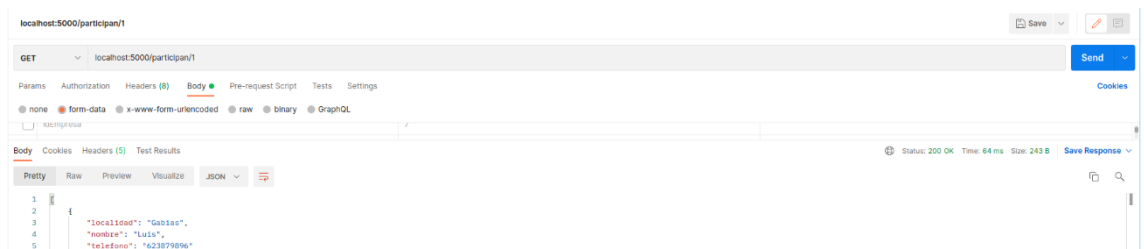
- **localhost:5000/empresas/update/<empresasid> (POST)**

- Este endpoint nos permite actualizar los datos de un registro sobre una empresa, siempre se deben de enviar todos los campos al servidor (en este caso nombre, ciudad, mail, teléfono). Utilizaremos el método de transferencia de datos POST.

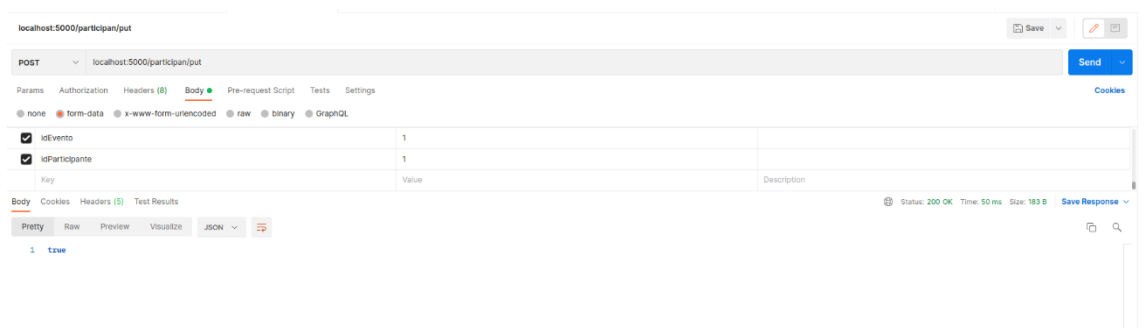


- **localhost:5000/empresas/<empresasid> (GET)**
 - Obtenemos un archivo JSON con la información de la empresa que hemos buscado a través de su id. Utilizaremos el método de transferencia de datos GET.

➤ PARTICIPAN (MUCHOS A MUCHOS)



- **localhost:5000/participan/<eventoid> (GET)**
 - Obtenemos un archivo JSON con el listado de todas los participantes, y su información, de un evento concreto. Pasamos en el endpoint el id del evento del cual queremos saber quiénes son los participantes. Utilizaremos el método de transferencia de datos GET.



- **localhost:5000/participan/put (POST)**
 - Utilizaremos este endpoint para crear una nueva relación entre evento y participante, es decir, lo utilizaremos para registrar un participante en un evento. Para ello deberemos de conocer el id del evento y el del participante. Utilizaremos el método de transferencia de datos POST.