Consumo API REST para listar eventos

URL	Method
http://localhost:8080/eventos/list http://localhost:8080/eventos http://localhost:8080/eventos/list?fecha=dd-MM-yy http://localhost:8080/eventos?fecha=dd-MM-yyyy	GET

1. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
fecha	Date	- Tiene que presentarlo en el formato indicado si no saldrá un error por formato incorrecto ya sea porque lo introduce en formato americano o porque ingresa cualquier otra secuencia no válida.

2. Argumentos de salida:

```
Caso solo listado:
{
    "eventos": [
            "idEventos": 1,
            "nombreEvento": "Concierto de Rock",
            "categoria": "Música",
            "capacidadMax": 200,
            "numReservasActual": 52,
            "fecha": "05-11-2024"
       },
{
            "idEventos": 2,
            "nombreEvento": "Taller de Pintura",
            "categoria": "Arte",
            "capacidadMax": 30,
            "numReservasActual": 10,
            "fecha": "06-11-2024"
        }
   ]
Caso listado con filtro por fecha (puede haber más de un evento en la
misma fecha:
```

```
"eventos": [
        {
            "idEventos": 1,
            "nombreEvento": "Concierto de Rock",
            "categoria": "Música",
            "capacidadMax": 200,
            "numReservasActual": 52,
            "fecha": "05-11-2024"
       },
{
            "idEventos": 2,
            "nombreEvento": "Taller de Pintura",
            "categoria": "Arte",
            "capacidadMax": 30,
            "numReservasActual": 10,
            "fecha": "05-11-2024"
        }
  ]
}
```

• Datos:

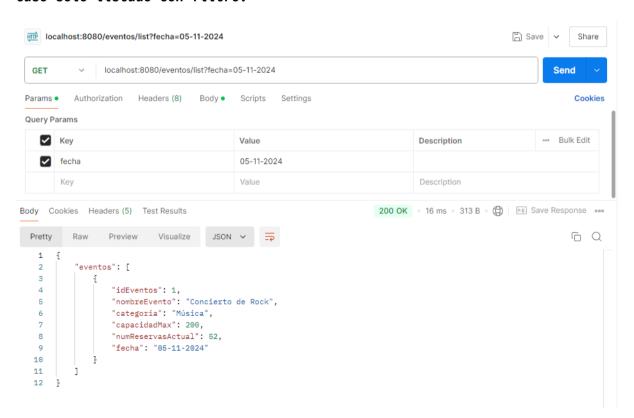
Campo	Tipo de Dato	Observaciones
idEventos	entero	-
nombreEvento	cadena	-
categoria	cadena	-
capacidadMax	entero	- Capacidad máxima para inscribirse
numReservasActual	entero	- Número total de inscritos
fecha	date	- Fecha de creación del evento

Ejemplo:

Caso solo listado:

```
localhost:8080/eventos
                                                                                                            □ Save ∨
                                                                                                                           Share
                                                                                                                                      </>>
            ∨ localhost:8080/eventos
  GET
                                                                                                                      Send
Params Authorization Headers (8) Body • Scripts Settings
                                                                                                                          Cookies
Body Cookies Headers (5) Test Results
                                                                            200 OK = 27 ms = 2.88 KB = ( Save Response •••
 Pretty
           Raw Preview Visualize JSON V
                                                                                                                          © Q
            "eventos": [
   4
                    "idEventos": 1,
                    "nombreEvento": "Concierto de Rock",
"categoria": "Música",
   5
                    "capacidadMax": 200,
   8
                    "numReservasActual": 52,
                    "fecha": "05-11-2024"
   10
   11
                    "idEventos": 2,
"nombreEvento": "Taller de Pintura",
   12
   13
                    "categoria": "Arte",
   14
                    "capacidadMax": 30,
   15
                     "numReservasActual": 10,
   16
                    "fecha": "06-11-2024"
   17
   18
   19
   20
                    "idEventos": 3,
                    "nombreEvento": "Conferencia de Tecnología",
   21
                    "categoria": "Tecnología",
   22
```

Caso solo listado con filtro:



Consumo API REST para crear evento

URL	Method
localhost:8080/eventos/crear	POST
localhost:8080/eventos/crear?fetchId=true	

1. Cuerpo del mensaje a enviar

Body:

```
"nombreEvento": "Conferencia sobre Antenas",
    "categoria": "Charla académica",
    "capacidadMax": 200,
    "fecha": "15-12-2025"
}
```

Observación:

Debe incluirse todos estos parámetros de lo contrario no se podrá realizar la creación del evento, del mismo modo se debe tener cuidado con el formato de la fecha y también si esta es posterior a la fecha de creación. Tenga cuidado con el límite de caracteres en categoría y nombre del evento (max 100).

2. Argumentos de salida:

Opción 1: Solicitud correcta

- Http Status: HTTP 201 Creada
- Body:

```
{
    "result": "sucessfull",
    "estado": "creado"
}
```

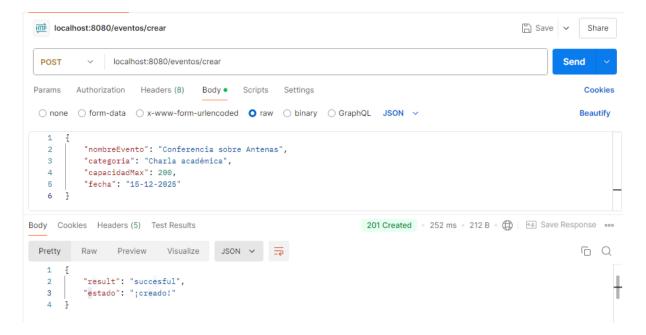
Opción 2: Error en los parámetros de la solicitud o en el establecimiento de una fecha que es pasada

- Http Status: HTTP 400 Bad Request
- Body:

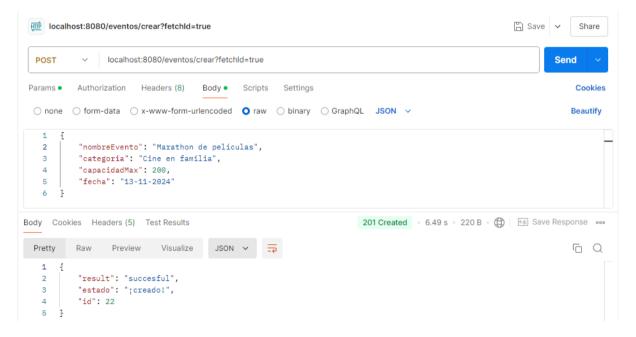
```
{
    "result": "failure",
    "msg": "información del error"
}
```

3. Ejemplos

Solicitud correcta:

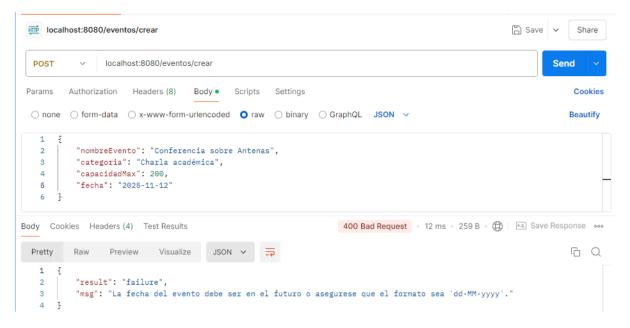


Utilizando el parámetro fetchId para saber el id del evento creado:

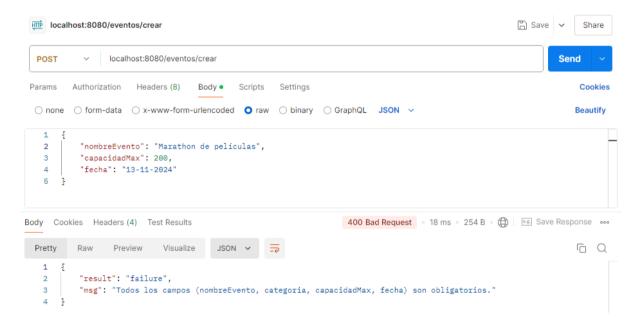


Solicitud incorrecta:

Fecha que no sigue el formato o es pasada:



No se especifica todos los campos requeridos:



Consumo API REST para registrarse a un evento

URL	Method
localhost:8080/eventos/registrar	POST

4. Cuerpo del mensaje a enviar

Body:

```
"evento": {
    "idEventos": 11
},
    "numeroCuposReserva": 3,
    "nombreApellido": "Juan Perez",
    "correo": "juan.perez@example.com"
}
```

Observación:

Debe incluirse todos estos parámetros de lo contrario no se podrá realizar el registro al evento, por otro lado es relevante resaltar que si el evento para el que se quiere registrar no presenta vacantes no se podrá continuar con el registro.

5. Argumentos de salida:

Opción 1: Solicitud correcta

- Http Status: HTTP 201 Creada
- Body:

```
{
    "result": "sucessfull",
    "msg": "Reserva realizada con éxito"
    "Id del registro": "123"
}
```

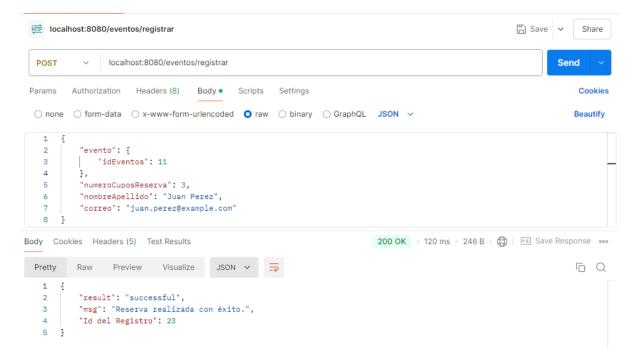
Opción 2: Error en los parámetros de la solicitud o problema con las vacantes

- Http Status: HTTP 400 Bad Request
- Body:

```
{
    "result": "failure",
    "msg": "información del error"
}
```

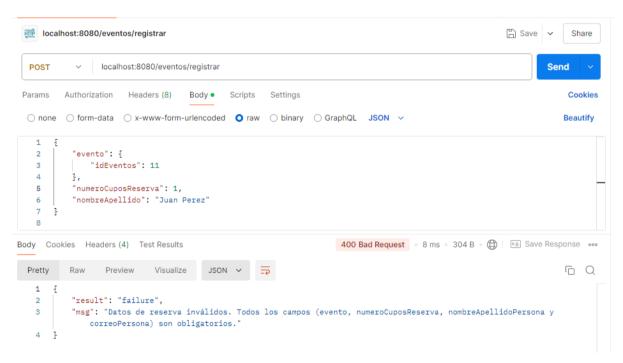
6. Ejemplos

Solicitud correcta:

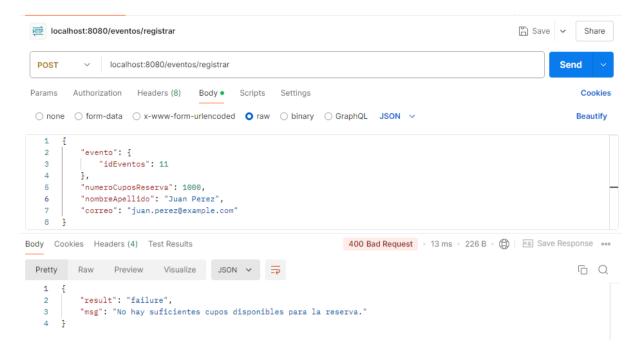


Solicitud incorrecta:

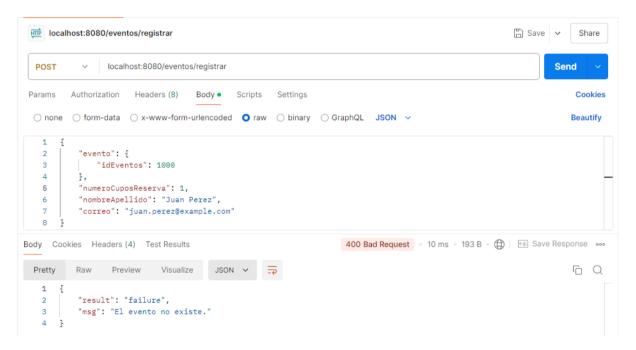
No se especifica todos los campos requeridos:



No hay más cupos para el evento:



El evento no existe:



Consumo API REST para borrar un registro

URL	Method
http://localhost:8080/eventos/borrar?id=xxx	DELETE

7. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	Integer	- Tiene que ser un ID válido, este es proporcionado en el momento de registrarse al evento.

8. Argumentos de salida:

Opción 1: Solicitud correcta

- Http Status: HTTP 201 Creada
- Body:

```
{
    "result": "sucessfull",
    "msg": "Reserva borrada con éxito. Se han liberado X cupos."
}
```

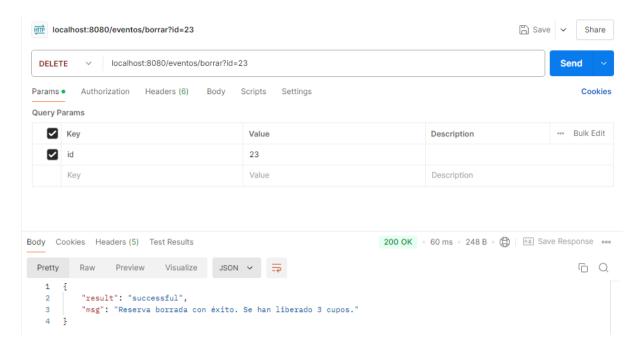
Opción 2: Error en los parámetros de la solicitud o problema con las vacantes

- Http Status: HTTP 400 Bad Request
- Body:

```
{
    "result": "failure",
    "msg": "información del error"
}
```

9. Ejemplos

Solicitud correcta:



Solicitud incorrecta:

ID de la reserva inválido:

