

## 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
<i>idEventos</i>	entero	-
<i>nombreEvento</i>	cadena	-
<i>categoria</i>	cadena	-
<i>capacidadMax</i>	entero	- Capacidad máxima para inscribirse
<i>numReservasActual</i>	entero	- Número total de inscritos
<i>fecha</i>	date	- Fecha de creación del evento

Ejemplo:

Caso solo listado:

localhost:8080/eventos

SaveShare

GETlocalhost:8080/eventosSend

ParamsAuthorizationHeaders (8)Body●ScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results200 OK27 ms2.88 KBSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "eventos": [
3     {
4       "idEventos": 1,
5       "nombreEvento": "Concierto de Rock",
6       "categoria": "Música",
7       "capacidadMax": 200,
8       "numReservasActual": 52,
9       "fecha": "05-11-2024"
10    },
11    {
12      "idEventos": 2,
13      "nombreEvento": "Taller de Pintura",
14      "categoria": "Arte",
15      "capacidadMax": 30,
16      "numReservasActual": 10,
17      "fecha": "06-11-2024"
18    },
19    {
20      "idEventos": 3,
21      "nombreEvento": "Conferencia de Tecnología",
22      "categoria": "Tecnología",
```

Caso solo listado con filtro:

localhost:8080/eventos/list?fecha=05-11-2024

SaveShare

GETlocalhost:8080/eventos/list?fecha=05-11-2024Send

Params●AuthorizationHeaders (8)Body●ScriptsSettingsCookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	fecha	05-11-2024		
	Key	Value	Description	

BodyCookiesHeaders (5)Test Results200 OK16 ms313 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "eventos": [
3     {
4       "idEventos": 1,
5       "nombreEvento": "Concierto de Rock",
6       "categoria": "Música",
7       "capacidadMax": 200,
8       "numReservasActual": 52,
9       "fecha": "05-11-2024"
10    }
11  ]
12 }
```

## Consumo API REST para crear evento

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

### 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:**

localhost:8080/eventos/crear

POST localhost:8080/eventos/crear

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "nombreEvento": "Conferencia sobre Antenas",
3   "categoria": "Charla académica",
4   "capacidadMax": 200,
5   "fecha": "15-12-2025"
6 }
```

Body Cookies Headers (5) Test Results

201 Created • 252 ms • 212 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "succesful",
3   "estado": "¡creado!"
4 }
```

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

localhost:8080/eventos/crear?fetchId=true

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

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "nombreEvento": "Marathon de películas",
3   "categoria": "Cine en familia",
4   "capacidadMax": 200,
5   "fecha": "13-11-2024"
6 }
```

Body Cookies Headers (5) Test Results

201 Created • 6.49 s • 220 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "succesful",
3   "estado": "¡creado!",
4   "id": 22
5 }
```

**Solicitud incorrecta:**

**Fecha que no sigue el formato o es pasada:**

HTTP localhost:8080/eventos/crear Save Share

POST localhost:8080/eventos/crear Send

Params Authorization Headers (8) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreEvento": "Conferencia sobre Antenas",
3   "categoria": "Charla académica",
4   "capacidadMax": 200,
5   "fecha": "2025-11-12"
6 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 12 ms 259 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "failure",
3   "msg": "La fecha del evento debe ser en el futuro o asegurese que el formato sea 'dd-MM-yyyy'."
4 }
```

No se especifica todos los campos requeridos:

HTTP localhost:8080/eventos/crear Save Share

POST localhost:8080/eventos/crear Send

Params Authorization Headers (8) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "nombreEvento": "Marathon de películas",
3   "capacidadMax": 200,
4   "fecha": "13-11-2024"
5 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 18 ms 254 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "failure",
3   "msg": "Todos los campos (nombreEvento, categoria, capacidadMax, fecha) son obligatorios."
4 }
```

## Consumo API REST para registrarse a un evento

URL	Method
localhost:8080/eventos/regar	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:**

localhost:8080/eventos/regar

POST localhost:8080/eventos/regar

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

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

```
1 {
2   "evento": {
3     "idEventos": 11
4   },
5   "numeroCuposReserva": 3,
6   "nombreApellido": "Juan Perez",
7   "correo": "juan.perez@example.com"
8 }
```

Body Cookies Headers (5) Test Results 200 OK • 120 ms • 246 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "successful",
3   "msg": "Reserva realizada con éxito.",
4   "Id del Registro": 23
5 }
```

### Solicitud incorrecta:

No se especifica todos los campos requeridos:

localhost:8080/eventos/regar

POST localhost:8080/eventos/regar

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

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

```
1 {
2   "evento": {
3     "idEventos": 11
4   },
5   "numeroCuposReserva": 1,
6   "nombreApellido": "Juan Perez"
7 }
8
```

Body Cookies Headers (4) Test Results 400 Bad Request • 8 ms • 304 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "failure",
3   "msg": "Datos de reserva inválidos. Todos los campos (evento, numeroCuposReserva, nombreApellidoPersona y correoPersona) son obligatorios."
4 }
```

No hay más cupos para el evento:



localhost:8080/eventos/regar

POST localhost:8080/eventos/regar

Params Authorization Headers (8) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "evento": {
3     "idEventos": 11
4   },
5   "numeroCuposReserva": 1000,
6   "nombreApellido": "Juan Perez",
7   "correo": "juan.perez@example.com"
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 13 ms 226 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "failure",
3   "msg": "No hay suficientes cupos disponibles para la reserva."
4 }
```

**El evento no existe:**

localhost:8080/eventos/regar

POST localhost:8080/eventos/regar

Params Authorization Headers (8) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "evento": {
3     "idEventos": 1000
4   },
5   "numeroCuposReserva": 1,
6   "nombreApellido": "Juan Perez",
7   "correo": "juan.perez@example.com"
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 10 ms 193 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "failure",
3   "msg": "El evento no existe."
4 }
```

## 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:**

localhost:8080/eventos/borrar?id=23

DELETE localhost:8080/eventos/borrar?id=23

Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	id	23			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK • 60 ms • 248 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "successful",
3   "msg": "Reserva borrada con éxito. Se han liberado 3 cupos."
4 }
```

## Solicitud incorrecta:

ID de la reserva inválido:

localhost:8080/eventos/borrar?id=100

DELETE localhost:8080/eventos/borrar?id=100

Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	id	100			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK • 13 ms • 217 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": "failure",
3   "msg": "El ID enviado no existe."
4 }
```