

Objetivo de la prueba:

Debe entregar el proyecto en un repositorio GitHub con su README explicando los diferentes pasos de instalación, configuración y uso **(20 puntos)**.

Respuesta:

Usted podrá encontrar el repositorio y el archivo readme.md en la siguiente dirección:

<https://github.com/felipedelosh/PruebaTecnicaSIER>

1. **(30 puntos)** Implementar una API que permita registrar eventos y el log de eventos donde:

En la API se puedan agregar, editar, leer, y borrar (soft destroy) la información de los eventos.

Los eventos contienen la siguiente información:

- ID
- Nombre del evento
- Tipo de Evento
- Descripción del Evento
- Fecha
- Estado (Pendiente por revisar / Revisado)

Respuesta:

Para los Logs usamos la librería por defecto de python: logging

El api fue creada en el lenguaje de programación python, con base de datos SQLite3 y todos los endpoints se manejan sobre el siguiente url: localhost:4000/event para agregar un nuevo evento cambia el verbo a POST y envía un json con los parámetros: nombre, tipo, descripción, fecha y estado, para editar cambia el verbo a PATCH y envíe en un json el identificador del evento junto a los parámetros que usted desea modificar, para leer cambie el verbo a GET y con el parámetro id envíe el identificador que quiere consultar o la palabra mágica para más filtros, para borrar cambie el verbo de DELETE y envíe en un Json el identificador que usted desee borrar.

2. **(20 puntos)** Genere de manera estática algunos eventos de ejemplo.

Respuesta:

Si la base de datos está vacía se ejecutará la línea de código: 19 del archivo "main.py" el cual insertará 6 eventos por defecto en la base de datos.

Los eventos los puede observar en: EventController.py método: generateStaticEvents()

3. **(30 puntos)** Suponga que los eventos después de revisados pueden ser clasificados en "Requiere gestión" o "sin gestión" dependiendo del tipo de Evento. Proponga las modificaciones necesarias (lógicas y endpoints) para agregar la funcionalidad de organización de eventos revisados por Requiere Gestión / Sin Gestión.

Respuesta:

0 -> Agregaremos un nuevo campo a la tabla event que se llamará "gestion" que será de tipo entero y contendrá los valores 1 o 0 e indicará si el evento ya fue gestionado o no.

Nota: no afecta los métodos anteriores insert debido a que están protegidos con especificación de values(a,b,c), (?, ?, ?)

1 -> Crearemos una nueva tabla en la base de datos llamada type_event_need_gestion. Dicha tabla contendrá los valores ID integer pk autoincrement, type_event string y esa tabla relaciona los eventos que necesitan gestión.

Ejemplo: todos los eventos "tipo A" necesitan gestión, por ello en la tabla type_event_need_gestion almacenaremos "tipo A" y así todos los eventos que tengan el tipo A van a necesitar gestión.

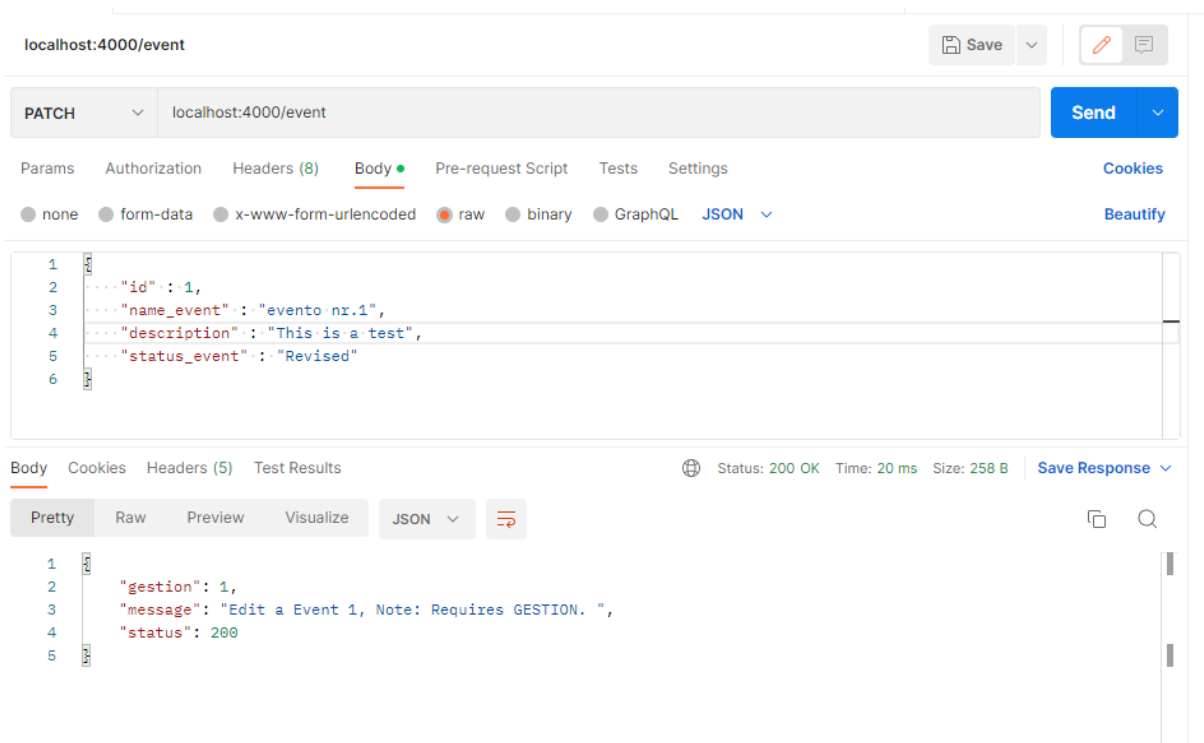
2 -> Vamos a crear el endpoint /gestion con los respectivos verbos GET, PATCH, POST, DELETE y luego gestionar los mismos desde el controlador de eventos.

3 -> Para la calificación de gestión:

En el controlador vamos a crear un método que diga si el evento requiere gestión y retornará falso o verdadero. y solo se activará cuando el usuario modifique el estado del evento. y en el Json vamos a agregar el campo "gestion" que indicara con 1 si necesita gestión o con 0 si no necesita gestion

Ejemplo:

Entre los eventos por defecto hay 2 tipos que requieren gestión "clase A" y "clase D" (sensible a mayúsculas) y entre los eventos por defecto el 1 y 2 son clase A y al momento de modificarlos saltará el siguiente mensaje:



Ejemplo 2:

El evento 3 no requiere gestión al modificarlo aparece lo siguiente:

localhost:4000/event

Save

PATCH

localhost:4000/event

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

2

3

4

5

6

```
{
  "id": 3,
  "name_event": "evento nr.1",
  "description": "This is a test",
  "status_event": "Revised"
}
```

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 19 ms

Size: 232 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

```
{
  "gestion": 0,
  "message": "Edit a Event 3",
  "status": 200
}
```