

AYÚDAME A CREAR UNA API EN PYTHON PARA EXTRAER REGISTROS DE LA TABLA "CURSO" PARA UNA BASE DE DATOS MYSQL

ChatGPT

Claro, te puedo ayudar a crear una API básica en Python utilizando **Flask** para interactuar con una base de datos MySQL y extraer registros de la tabla "Curso". Vamos a hacerlo paso a paso.

Paso 1: Instalación de dependencias

Primero, asegúrate de tener instalados los siguientes paquetes:

- Flask: Framework para construir la API.
- mysql-connector-python: Para conectarse a la base de datos MySQL.

Puedes instalarlos usando pip:

bash

Copiar código

```
pip install Flask mysql-connector-python
```

Paso 2: Configuración de la base de datos

Vamos a asumir que tienes una base de datos MySQL con una tabla llamada Curso. Asegúrate de que la tabla tenga la siguiente estructura básica:

sql

Copiar código

```
CREATE TABLE Curso (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(255),  
    descripcion TEXT,  
    fecha_inicio DATE,  
    fecha_fin DATE  
);
```

Paso 3: Crear el archivo app.py

Crea un archivo llamado app.py y agrega el siguiente código:

python

Copiar código

```
from flask import Flask, jsonify  
import mysql.connector  
from mysql.connector import Error  
  
app = Flask(__name__)
```

```
# Configuración de la conexión a la base de datos
```

```

def connect_to_db():
    try:
        connection = mysql.connector.connect(
            host='localhost',
            database='nombre_de_tu_base_de_datos',
            user='tu_usuario',
            password='tu_contraseña'
        )
        if connection.is_connected():
            return connection
    except Error as e:
        print("Error al conectar a la base de datos", e)
        return None

# Ruta para obtener todos los registros de la tabla Curso
@app.route('/cursos', methods=['GET'])
def get_cursos():
    connection = connect_to_db()
    if connection:
        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT * FROM Curso")
        cursos = cursor.fetchall()
        cursor.close()
        connection.close()
        return jsonify(cursos)
    else:
        return jsonify({"error": "Error en la conexión a la base de datos"}), 500

if __name__ == '__main__':
    app.run(debug=True)

```

Paso 4: Ejecución de la API

Guarda el archivo y ejecuta el siguiente comando en la terminal:

```
bash
```

Copiar código

```
python app.py
```

Esto iniciará un servidor web en `http://127.0.0.1:5000/`. Puedes acceder a los registros de la tabla `Curso` accediendo a `http://127.0.0.1:5000/cursos` en tu navegador o mediante herramientas como Postman.

Explicación del código:

- **connect_to_db:** Establece la conexión con la base de datos MySQL.
- **/cursos:** Esta ruta responde a las solicitudes GET y devuelve todos los registros de la tabla `Curso` en formato JSON.
- **jsonify:** Convierte los datos obtenidos de la base de datos en una respuesta JSON.

Paso 5: Personalizar y Expandir

Puedes expandir esta API agregando más rutas para insertar, actualizar o eliminar registros. Por ejemplo, podrías agregar una ruta para obtener un curso por su id, o una para crear un nuevo curso.

¿Te gustaría que te ayude a agregar más funcionalidades o tienes alguna otra pregunta sobre este código?