



Uso de SQLITE con Python y Flask



```
import sqlite3

conn = sqlite3.connect('mydb.db3')
c = conn.cursor()

c.execute('SELECT * FROM Users')

for row in c:
    print(row)

c.close()
conn.close()
```



Ejemplos



Tipos de datos

Python type	SQLite type
<code>None</code>	<code>NULL</code>
<code>int</code>	<code>INTEGER</code>
<code>float</code>	<code>REAL</code>
<code>str</code>	<code>TEXT</code>
<code>bytes</code>	<code>BLOB</code>

SQLite type	Python type
<code>NULL</code>	<code>None</code>
<code>INTEGER</code>	<code>int</code>
<code>REAL</code>	<code>float</code>
<code>TEXT</code>	depends on <code>text_factory</code> , <code>str</code> by default
<code>BLOB</code>	<code>bytes</code>

Multi thread

- No está permitido compartir conexiones entre threads.
- Se debe generar una conexión por thread.
- `sqlite3.connect()` tiene el argumento ***check_same_thread***. Si se le pasa `False`, se permite compartir la conexión entre threads, pero hay que sincronizar los accesos manualmente.



Múltiples procesos

- Cuando un proceso escribe la DB, la misma se bloquea hasta que termine la transacción.
- `sqlite3.connect()` tiene el argumento ***timeout*** que permite configurar un tiempo de espera mientras la DB está locked.
- Luego de este tiempo, se lanzará una excepción.
- El tiempo por default es de 5 segundos.



Uso con Flask

```
import sqlite3
from flask import g

DATABASE = '/path/to/database.db'

def get_db():
    db = getattr(g, '_database', None)
    if db is None:
        db = g._database = sqlite3.connect(DATABASE)
    return db

@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, '_database', None)
    if db is not None:
        db.close()
```



Uso con Flask

```
@app.route('/')  
def index():  
    cur = get_db().cursor()  
    ...
```




Ejemplo



Cómo leer valores en URL con Flask

Si el request es:

- URL=/log?page=2
- Método=GET

```
@app.route('/log')  
def log():  
    page = request.args.get('page', '0')  
    ...
```



Cómo leer valores por PUT/POST con Flask

Si el request es:

- URL=/devices
- Método=PUT
- Body= "state":0 // clave-valor

```
@app.route('/devices')
def devices():
    state = request.form['state'] #leemos campo state
    ...
```



Bibliografía

- <https://docs.python.org/3.5/library/sqlite3.html>
- <https://flask.palletsprojects.com/en/1.1.x/patterns/sqlite3/>