

# Come creare un sistema di logging delle visite di un sito web con Python

Creare un sistema di logging delle visite per un sito web è un aspetto cruciale per chiunque desideri monitorare l'attività degli utenti, raccogliere dati per analisi o migliorare l'esperienza utente. Python offre una vasta gamma di librerie e strumenti che facilitano questo processo. Di seguito, esploreremo come costruire un sistema di logging delle visite web usando Python, partendo da una semplice implementazione basata su file fino a soluzioni più avanzate che utilizzano database e strumenti di visualizzazione.

Il logging delle visite permette di:

- Monitorare il traffico del sito in tempo reale.
- Comprendere quali pagine e risorse sono più popolari.
- Rilevare comportamenti insoliti, come attacchi o attività sospette.
- Migliorare l'ottimizzazione per i motori di ricerca (SEO).
- Creare report per capire l'andamento del sito e prendere decisioni strategiche.

Per ogni visita, è possibile registrare le seguenti informazioni:

- **Timestamp:** l'ora in cui la visita è avvenuta.
- **Indirizzo IP:** l'indirizzo IP del visitatore.
- **User-Agent:** informazioni sul browser e il sistema operativo del visitatore.
- **Pagina visitata:** l'URL della pagina visitata.
- **Referrer:** la pagina da cui proviene il visitatore, utile per capire da dove arrivano i visitatori.

Per cominciare, possiamo creare un semplice sistema di logging basato su file, utilizzando la libreria logging di Python, che è potente e semplice da configurare.

```
import logging
from datetime import datetime

# Configurazione del logger
logging.basicConfig(
    filename='web_visits.log',
    level=logging.INFO,
    format='%(asctime)s - %(message)s',
)

def log_visit(ip_address, user_agent, page,
referrer):
    # Logga la visita
    visit_data = f"IP: {ip_address}, Page: {page},
User-Agent: {user_agent}, Referrer: {referrer}"
    logging.info(visit_data)

# Esempio di utilizzo
log_visit("192.168.0.1", "Mozilla/5.0", "/home",
"https://google.com")
```

In questo esempio, ogni volta che un utente visita una pagina, possiamo chiamare la funzione `log_visit()` con i parametri necessari. Il logger scriverà le informazioni in un file di log, aggiungendo automaticamente il timestamp.

Per siti con un traffico elevato, salvare i dati in un database è spesso una soluzione migliore, poiché rende più semplice analizzare i dati in seguito. In

questo caso, possiamo usare SQLite per testare, oppure PostgreSQL o MySQL per implementazioni in produzione.

Ecco un esempio di come creare un database per il logging delle visite usando SQLite:

```
import sqlite3

# Crea il database e la tabella per le visite
conn = sqlite3.connect('web_visits.db')
cursor = conn.cursor()
cursor.execute('''CREATE TABLE IF NOT EXISTS visits (
                id INTEGER PRIMARY KEY
                AUTOINCREMENT,
                timestamp DATETIME DEFAULT
                CURRENT_TIMESTAMP,
                ip_address TEXT,
                user_agent TEXT,
                page TEXT,
                referrer TEXT
                )''')
conn.commit()
```

Ora possiamo creare una funzione `log_visit_to_db` per registrare i dati delle visite nel database:

```
def log_visit_to_db(ip_address, user_agent, page,
referrer):
    conn = sqlite3.connect('web_visits.db')
    cursor = conn.cursor()
    cursor.execute('''INSERT INTO visits (ip_address,
```

```
user_agent, page, referrer)
                VALUES (?, ?, ?, ?)'''
(ip_address, user_agent, page, referrer))
    conn.commit()
    conn.close()
```

Con questa funzione, ogni visita viene registrata come una nuova riga nella tabella `visits`.

Se il tuo sito web è sviluppato in Python, ad esempio utilizzando il framework Flask, puoi facilmente integrare il sistema di logging. Flask fornisce strumenti per accedere agli header della richiesta, che includono dati come l'IP e l'User-Agent.

Ecco come estendere il nostro sistema di logging per funzionare con un'app Flask:

```
from flask import Flask, request

app = Flask(__name__)

@app.route('/')
def home():
    log_visit_to_db(
        ip_address=request.remote_addr,
        user_agent=request.headers.get('User-Agent'),
        page=request.path,
        referrer=request.referrer
    )
    return "Benvenuto alla homepage!"
```

```
if __name__ == "__main__":  
    app.run()
```

In questo esempio, ogni volta che un utente accede alla homepage (/), i suoi dati vengono registrati nel database. Possiamo estendere questo comportamento a tutte le pagine dell'app, chiamando la funzione `log_visit_to_db` in ogni route.

## Conclusione

Implementare un sistema di logging delle visite con Python consente di raccogliere preziosi dati sui visitatori e di ottenere informazioni sul comportamento degli utenti. Partendo da un semplice sistema basato su file, è possibile evolvere verso soluzioni basate su database e visualizzazioni avanzate per migliorare la gestione e l'analisi dei dati.