

GABRIELE ROMANATO

Menu

Effettuare richieste HTTP in Python con il modulo requests

Il modulo `requests` è uno dei modi più semplici e popolari per eseguire richieste HTTP in Python. È facile da usare e consente di interagire con API web o scaricare dati da Internet in pochi passaggi.

Installazione

Per installare `requests`, basta utilizzare `pip`:

```
pip install requests
```

Effettuare una richiesta GET

Una richiesta GET viene utilizzata per recuperare dati da un server. Ecco un esempio base:

```
import requests

response = requests.get('https://api.example.com/data')

if response.status_code == 200:
    print(response.text)
else:
    print(f'Errore: {response.status_code}')
```

Inviare dati con una richiesta POST

Se devi inviare dati a un server, puoi usare una richiesta POST:

```
import requests

url = 'https://api.example.com/submit'
data = {'nome': 'Mario', 'età': 30}

response = requests.post(url, data=data)

if response.ok:
    print('Dati inviati con successo')
else:
    print('Errore nell'invio dei dati')
```

Gestire le eccezioni

È buona pratica gestire eventuali errori durante le richieste, ad esempio problemi di connessione:

```
import requests

try:
    response = requests.get('https://api.example.com/data', timeout=5)
    response.raise_for_status()
    print(response.json())
except requests.exceptions.HTTPError as errh:
    print(f'Errore HTTP: {errh}')
except requests.exceptions.ConnectionError as errc:
    print(f'Errore di Connessione: {errc}')
except requests.exceptions.Timeout as errt:
    print(f'Tempo di richiesta scaduto: {errt}')
except requests.exceptions.RequestException as err:
    print(f'Errore generico: {err}')
```

Conclusione

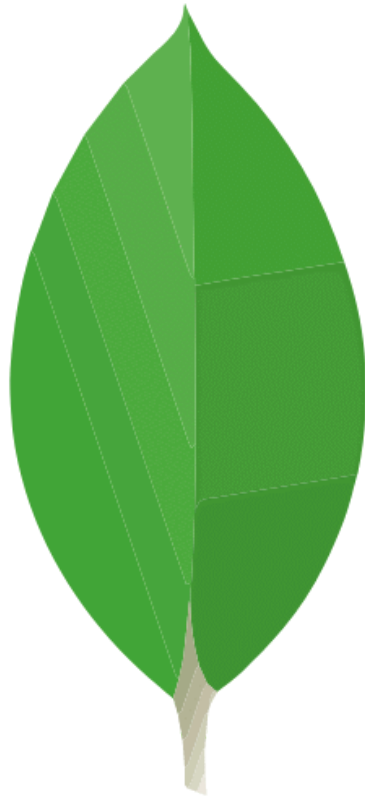
Il modulo `requests` rende semplice e immediato lavorare con HTTP in Python. Con pochi comandi puoi eseguire richieste, inviare dati e gestire eventuali errori, rendendo le tue applicazioni più robuste e affidabili.

Applicazioni Correlate



Python Placeholder Image

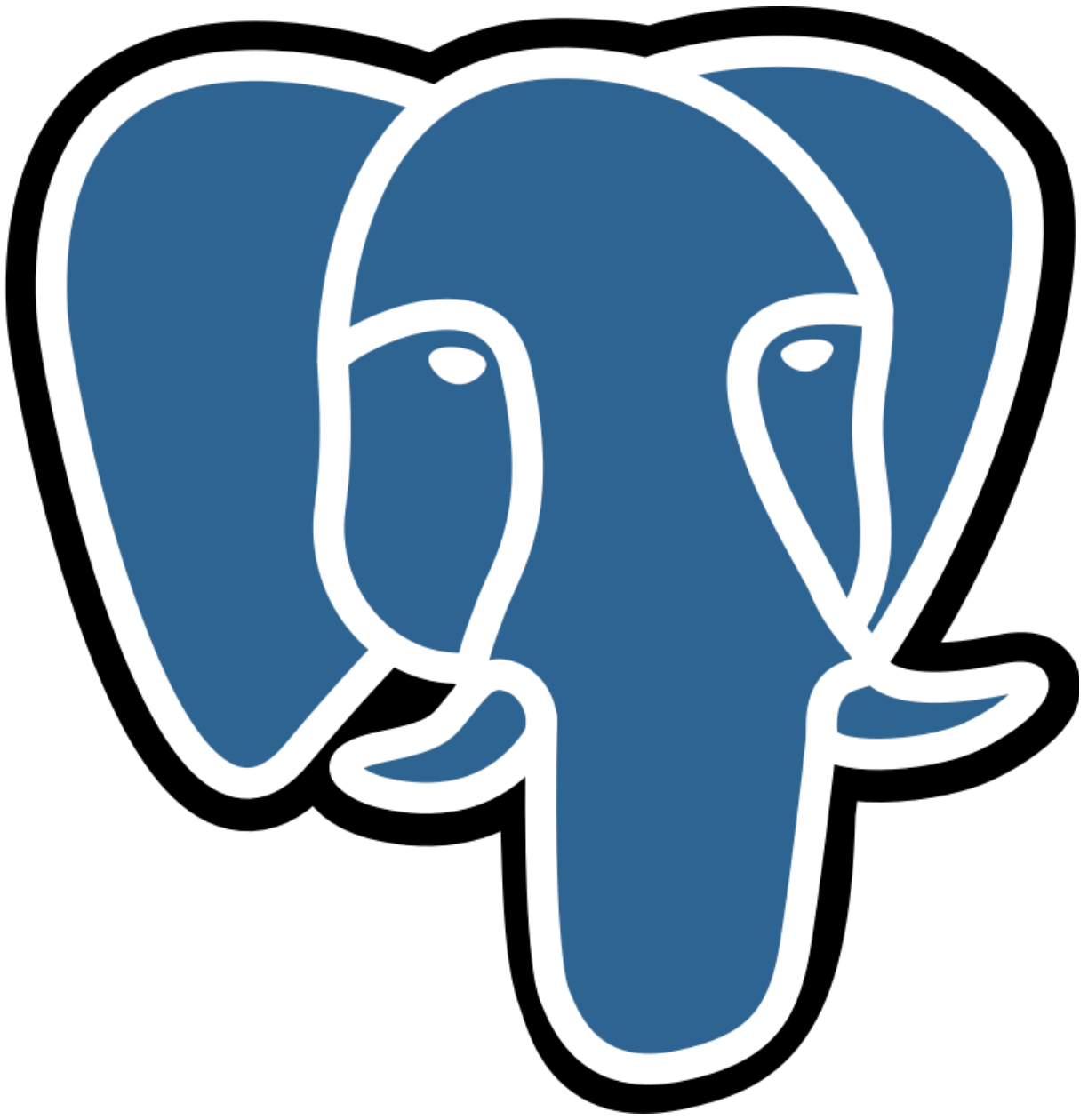
Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.
Docker Docker Compose Python Flask JavaScript



-

Python MongoDB App

Applicazione basata su MongoDB e sviluppata in Python e Flask.
DockerDocker ComposePythonFlaskMongoDB



•

Python PostgreSQL App

Applicazione basata su PostgreSQL con Python e Flask.

Docker Docker Compose Python Flask