

Python: implementazione delle Web Push Notifications con Flask

Le web push notifications sono uno strumento efficace per mantenere l'utente coinvolto anche quando non è attivamente sulla tua applicazione web. In questo articolo, esploreremo come implementare le web push notifications utilizzando il framework Python Flask.

Prima di iniziare, è necessario avere una conoscenza di base di Python, Flask e JavaScript. Dovrai anche avere Node.js installato sul tuo sistema per poter utilizzare alcuni strumenti necessari.

Per implementare le push notifications, seguiremo questi passaggi:

1. Configurare il server Flask.
2. Generare le chiavi VAPID per l'autenticazione.
3. Gestire l'iscrizione alle notifiche nel client.
4. Inviare le notifiche dal server Flask.

Per prima cosa, crea un progetto Flask. Inizia con l'installazione di Flask:

```
pip install Flask
```

Poi, crea un file `app.py` e configura una semplice app Flask:

```
from flask import Flask, render_template

app = Flask(__name__)
```

```
@app.route('/')
def index():
    return render_template('index.html')

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

Crea anche un template HTML di base chiamato `index.html` nella cartella `templates`:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>Web Push Notifications</title>
</head>
<body>
    <h1>Web Push Notifications con Flask</h1>
    <button id="subscribe">Iscriviti alle
Notifiche</button>

    <script src="app.js"></script>
</body>
</html>
```

Le chiavi VAPID (Voluntary Application Server Identification) sono utilizzate per autenticare il server che invia le notifiche. Puoi generarle utilizzando un pacchetto npm chiamato `web-push`.

Installa web-push:

```
npm install web-push -g
```

Genera le chiavi VAPID:

```
web-push generate-vapid-keys
```

Questo comando ti fornirà una chiave pubblica e una chiave privata. Salva queste chiavi perché le useremo nel server Flask.

Nel file `app.js`, aggiungi il seguente codice per gestire l'iscrizione del client alle notifiche push:

```
const publicVapidKey =  
'LA_TUA_CHIAVE_VAPID_PUBBLICA';  
  
if ('serviceWorker' in navigator) {  
  send().catch(err => console.error(err));  
}  
  
async function send() {  
  const register = await  
  navigator.serviceWorker.register('/worker.js', {  
    scope: '/'  
  });  
  
  const subscription = await
```

```

register.pushManager.subscribe({
    userVisibleOnly: true,
    applicationServerKey:
urlBase64ToUint8Array(publicVapidKey)
});

    await fetch('/subscribe', {
        method: 'POST',
        body: JSON.stringify(subscription),
        headers: {
            'content-type': 'application/json'
        }
    });
}

function urlBase64ToUint8Array(base64String) {
    const padding = '='.repeat((4 -
base64String.length % 4) % 4);
    const base64 = (base64String + padding)
        .replace(/-/g, '+')
        .replace(/_/g, '/');

    const rawData = window.atob(base64);
    const outputArray = new
Uint8Array(rawData.length);

    for (let i = 0; i < rawData.length; ++i) {
        outputArray[i] = rawData.charCodeAt(i);
    }
    return outputArray;
}

```

Crea anche un file `worker.js` nella root del tuo progetto:

```
self.addEventListener('push', event => {
  const data = event.data.json();
  self.registration.showNotification(data.title, {
    body: data.message,
    icon: 'icon.png'
  });
});
```

Installa il pacchetto pywebpush:

```
pip install pywebpush
```

Aggiungi le seguenti righe al tuo `app.py` per gestire la sottoscrizione e inviare notifiche:

```
import json
from flask import request
from pywebpush import webpush, WebPushException

VAPID_PUBLIC_KEY = "LA_TUA_CHIAVE_VAPID_PUBBLICA"
VAPID_PRIVATE_KEY = "LA_TUA_CHIAVE_VAPID_PRIVATA"

@app.route('/subscribe', methods=['POST'])
def subscribe():
    subscription_info = request.get_json()
    return "Subscribed", 201

def send_web_push(subscription_information,
```

```

message_body):
    try:
        webpush(

subscription_info=subscription_information,
                data=json.dumps(message_body),
                vapid_private_key=VAPID_PRIVATE_KEY,
                vapid_claims={
                    "sub": "mailto:tuo@email.com"
                }
            )
    except WebPushException as ex:
        print(f"Error: {repr(ex)}")

@app.route('/notify', methods=['POST'])
def notify():
    subscription_info = request.get_json()
    message = {
        "title": "Titolo della Notifica",
        "message": "Questo è il corpo della notifica"
    }
    send_web_push(subscription_info, message)
    return "Notification sent", 200

```

Ora, puoi testare le notifiche eseguendo l'app Flask e inviando una richiesta POST a /notify con i dati della sottoscrizione.

Conclusione

Con questi passaggi, hai implementato un semplice sistema di web push notifications utilizzando Flask. Questo esempio può essere ampliato e personalizzato per soddisfare le esigenze specifiche della tua applicazione.

Le notifiche push sono un potente strumento per mantenere gli utenti coinvolti, e con Flask, la loro implementazione è abbastanza diretta.