

Hackathon Notification System with Twilio SMS API

Project Overview

This project demonstrates an automated system to send personalized SMS notifications to participants during a hackathon using Twilio's SMS API. It integrates dynamic recipient retrieval, error handling, and best practices for secure API usage.

1. Prerequisites

- **Twilio Account:** Sign up at [Twilio](#).
- **Twilio Phone Number:** Capable of sending SMS.
- **Python 3.x** installed.
- Install Twilio library:

```
pip install twilio
```

- Install dotenv for secure credential storage:

```
pip install python-dotenv
```

Create a `.env` file:

```
TWILIO_ACCOUNT_SID=your_account_sid_here  
TWILIO_AUTH_TOKEN=your_auth_token_here
```

2. Integration Code

`send_sms.py`

```
from twilio.rest import Client  
import os  
  
# Load credentials  
from dotenv import load_dotenv  
load_dotenv()  
account_sid = os.getenv('TWILIO_ACCOUNT_SID')  
auth_token = os.getenv('TWILIO_AUTH_TOKEN')  
client = Client(account_sid, auth_token)  
  
# Fetch recipient number dynamically
```

```
def get_recipient_number():
    try:
        messages = client.messages.list(limit=1)
        if messages:
            return messages[0].to
        else:
            print("No recent messages found to fetch a recipient number.")
            return None
    except Exception as e:
        print(f"Error fetching recipient: {e}")
        return None

# Send notification
def send_hackathon_reminder(event_name, session_time):
    recipient_phone = get_recipient_number()
    if recipient_phone:
        message_body = f"Reminder: '{event_name}' starts at {session_time}. Don't miss it!"
        try:
            message = client.messages.create(body=message_body,
            from_='+1234567890', to=recipient_phone)
            print(f"Message sent! SID: {message.sid}")
        except Exception as e:
            print(f"Failed to send message: {e}")
        else:
            print("Skipping reminder: No recipient number found.")

if __name__ == "__main__":
    send_hackathon_reminder("AI Innovations Workshop", "2:00 PM")
```

3. Workflow Diagram

1. Initialize and load credentials.
2. Fetch recipient from recent messages.
3. Create and send SMS notification.
4. Check API response for success or failure.
5. Log status or handle error.

- [Workflow Diagram](#)

4. Best Practices

- **Secure API Keys:** Use environment variables with `python-dotenv`.
- **Log Message SIDs:** Track for auditing and troubleshooting.
- **Implement Rate Limiting:** Avoid exceeding API limits.

5. Example curl Request

```
curl -X POST https://api.twilio.com/2010-04-01/Accounts/{AccountSID}/Messages.json \
--user "{AccountSID}:{AuthToken}" \
--data-urlencode "To="+1987654321" \
--data-urlencode "From="+1234567890" \
--data-urlencode "Body=Reminder: Hackathon starts at 10 AM."
```

Explanation of the curl Request

- **Endpoint:** `https://api.twilio.com/2010-04-01/Accounts/{AccountSID}/Messages.json`
 - This is Twilio's API endpoint for sending messages.
- **Authentication:**
 - `--user "{AccountSID}:{AuthToken}"` authenticates the request using your Twilio Account SID and Auth Token.
- **Parameters:**
 - **To:** The recipient's phone number (replace with actual number).
 - **From:** Your Twilio phone number.
 - **Body:** The content of the SMS message.

**6. Status Codes **

Response Status Codes

- When working with Twilio's SMS API, you'll encounter these common status codes:

```
# Success Codes
200: "Request successful"
201: "Message created successfully"

# Client Error Codes
400: "Bad request - Check message parameters"
401: "Authentication failed - Verify credentials"
404: "Resource not found"
429: "Too many requests - Rate limit exceeded"

# Server Error Codes
500: "Internal server error"
503: "Service temporarily unavailable"
```

Message Status Values

```
MESSAGE_STATUSES = {
    'queued': "Message is queued to be sent",
    'sending': "Message is being sent",
    'sent': "Message was sent successfully",
    'delivered': "Message was delivered to recipient",
```

```
'failed': "Message failed to send",  
'undelivered': "Message couldn't be delivered",  
}
```
