How to Change Telegram Account for Message Monitoring

This guide explains how to switch between different Telegram accounts for message monitoring in your scraper application.

🔄 Method 1: Quick Account Switch (Recommended)

Use this method to completely switch from one account to another.

Step 1: Stop Current App

```bash
# Press Ctrl+C to stop the current monitoring session
```

Step 2: Delete Current Session

```bash
# From project root (C:\Desktop\web_scrapper)
# Delete the current Telegram session file
del sessions\message_scraper.session

# On Linux/Mac:
# rm sessions/message_scraper.session
```

Note: This will log out the current account and require re-authentication.

Step 3: Update Config with New Account

Edit `config/.env` with the new account details:

```bash
# Change to new account credentials
TELEGRAM_API_ID=your_api_id        # Same API ID (can reuse)
TELEGRAM_API_HASH=your_api_hash    # Same API hash (can reuse)
TELEGRAM_PHONE=+1234567890         # NEW PHONE NUMBER HERE
```

Important: Only the phone number needs to change. The API ID and API hash can be reused for different accounts.

Step 4: Clear Target Chats (Temporary)

Edit `config/config.yaml`:

```yaml
telegram:
  target_chats: []  # Clear old chats temporarily
```

Step 5: Restart App

```bash
python src/main.py start
```

The app will prompt for a verification code for the **new phone number**.

🔄 Method 2: Use Different Session Name

Use this method to keep multiple account sessions without deleting the previous one.

Update config/config.yaml:

```yaml
telegram:
  api_id: "your_api_id"
  api_hash: "your_api_hash"
  phone_number: "+NEW_PHONE_NUMBER"     # New account phone
  session_name: "new_account_session"   # Different session name
  target_chats: []                      # Will need to update these
```

Benefits:

- Keeps both accounts' sessions
- Easy to switch between accounts
- No need to re-authenticate each time

🔢 Method 3: Multi-Account Setup

Use this method to monitor multiple accounts simultaneously (advanced setup).

Create Separate Config Files:

config/account1.yaml:

```yaml
telegram:
  api_id: "your_api_id"
  api_hash: "your_api_hash"
  session_name: "account1_session"
  phone_number: "+1234567890"
  target_chats: [1786888960, "Group1"]

notifications:
  method: "pushbullet"
  pushbullet:
    access_token: "your_pushbullet_token"
```

config/account2.yaml:

```yaml
telegram:
  api_id: "your_api_id"
  api_hash: "your_api_hash"
  session_name: "account2_session"
  phone_number: "+9876543210"
  target_chats: [9876543210, "Group2"]

notifications:
  method: "pushbullet"
  pushbullet:
    access_token: "your_pushbullet_token"
```

Run with Specific Config:

```bash
# Copy desired config to main config
copy config\account1.yaml config\config.yaml
python src/main.py start
```

🔍 Method 4: Find Available Chats for New Account

After switching accounts, you need to discover the available chats for the new account.

Step 1: Login with New Account

bash

```bash
python src/main.py start
# Enter verification code when prompted
# App will connect as new user
```

Step 2: Stop and Check Available Chats

bash

```bash
# Stop the app (Ctrl+C)
python src/main.py chats
```

Expected Output:

```
📱 Available Telegram Chats:

Chat ID | Chat Name | Type
--------------------------------------------------
5555555555      | New Friend      | Private
9999999999      | Another Contact | Private
-1001111111111  | New Group       | Group
-1002222222222  | Work Team       | Group
```

Step 3: Update Target Chats

Edit `config/config.yaml` with the new chat IDs:

yaml

```yaml
telegram:
  target_chats:
    - 5555555555      # New friend's chat
    - -1001111111111  # New group
    - "Group Name"    # Or use display names
```

Step 4: Start Monitoring

bash

```bash
python src/main.py start
```

🎯 Complete Step-by-Step Example

Let's walk through switching from **Vedant's account** to **another account**:

Current State:

- Monitoring: Vedant's account (+919054484280)
- Chats: Vedant (1786888960), Dharmishtha (811761632)

Goal:

- Switch to: Different account (+1234567890)
- Monitor: New account's chats

Steps:

1. **Stop Current Monitoring:**

   ```bash
   # Press Ctrl+C in the running app
   ```

2. **Delete Current Session:**

   ```bash
   del sessions\message_scraper.session
   ```

3. **Update Phone Number:** Edit `config/.env`:

   ```bash
   TELEGRAM_PHONE=+1234567890  # New account phone
   ```

4. **Clear Old Chats:** Edit `config/config.yaml`:

   ```yaml
   telegram:
     target_chats: []  # Clear old chats
   ```

5. **Login with New Account:**

   ```bash
   python src/main.py start
   # Enter verification code for +1234567890
   ```

6. **Find New Account's Chats:**

```bash
# Stop app (Ctrl+C)
python src/main.py chats
```

7. **Update Target Chats:** Based on the output, update `config/config.yaml`:

```yaml
telegram:
  target_chats:
    - 5555555555      # New contact
    - -1001111111111  # New group
```

8. **Start Final Monitoring:**

```bash
python src/main.py start
```

Result: Now monitoring the new account's chats!

🔧 Quick Switch Automation

Create batch files for easy account switching:

Windows: switch_account1.bat

```batch
@echo off
echo Switching to Account 1...
del sessions\message_scraper.session
echo TELEGRAM_PHONE=+1234567890 > temp_env.txt
type config\.env | findstr /v "TELEGRAM_PHONE" > temp_config.txt
type temp_env.txt >> temp_config.txt
move temp_config.txt config\.env
del temp_env.txt
echo Starting monitoring for Account 1...
python src/main.py start
```

Windows: switch_account2.bat

batch

```batch
@echo off
echo Switching to Account 2...
del sessions\message_scraper.session
echo TELEGRAM_PHONE=+9876543210 > temp_env.txt
type config\.env | findstr /v "TELEGRAM_PHONE" > temp_config.txt
type temp_env.txt >> temp_config.txt
move temp_config.txt config\.env
del temp_env.txt
echo Starting monitoring for Account 2...
python src/main.py start
```

Linux/Mac: switch_account1.sh

bash

```bash
#!/bin/bash
echo "Switching to Account 1..."
rm sessions/message_scraper.session
sed -i '/TELEGRAM_PHONE/d' config/.env
echo "TELEGRAM_PHONE=+1234567890" >> config/.env
echo "Starting monitoring for Account 1..."
python src/main.py start
```

📋 Important Notes

API Credentials Reuse:

- ✅ **Same API ID and API Hash** can be used for multiple accounts
- ✅ **Only phone number** needs to change per account
- ✅ **Session files** are account-specific

Session Management:

- Each account creates its own `.session` file
- Session files store login state to avoid re-authentication
- Deleting session file = need to re-login with verification code

Chat IDs are Account-Specific:

- Chat IDs are different for each account

- Same group may have different ID when accessed from different accounts
- Always run `python src/main.py chats` after switching accounts

Notification Settings:

- Notification settings (Pushbullet token) remain the same
- All accounts will send notifications to the same device
- Consider adding account identifiers in notification text

🚨 Troubleshooting

Problem: "Phone number already registered"

Solution: You're trying to register the same phone number twice. Delete the session file first.

Problem: "Can't find any chats"

Solution: Run `python src/main.py chats` to see available chats for the current account.

Problem: "Invalid session"

Solution: Delete the session file and re-authenticate:

```bash
del sessions\message_scraper.session
python src/main.py start
```

Problem: "No target chats configured"

Solution: Update `config/config.yaml` with valid chat IDs for the current account.

✅ Verification Checklist

Before switching accounts, ensure:

- [ ] Current monitoring is stopped (Ctrl+C)
- [ ] Session file is deleted (if doing complete switch)
- [ ] Phone number is updated in config/.env
- [ ] Target chats are cleared or updated
- [ ] New account has access to desired chats
- [ ] Pushbullet/notification settings are configured

After switching:

- [ ] New account logged in successfully
- [ ] Available chats discovered with `chats` command
- [ ] Target chats updated in config
- [ ] Test notification sent successfully
- [ ] Monitoring started for new account

🎯 Quick Reference Commands

```bash
# Stop monitoring
Ctrl+C

# Delete session (complete switch)
del sessions\message_scraper.session

# Check available chats for current account
python src/main.py chats

# Start monitoring
python src/main.py start

# Send test notification
python src/main.py test-notify
```

Remember: Each Telegram account will have different available chats and chat IDs. Always verify the chats are accessible from the new account before adding them to your target list.