

Flickr to Immich Migration

Complete Guide with Docker/Podman

Cross-Platform: Linux, Mac, Windows

Platform	Browser	Support
Linux	X11 Forwarding (automatic)	Fully supported
Mac	Open URL manually	Supported
Windows (WSL2)	Open URL manually	Supported

1. Overview

This guide describes how to migrate a Flickr photo library to Immich using the Flickr API. The process runs in a Docker/Podman container and works on Linux, Mac, and Windows.

Workflow:

- **flickr_download** - Download photos via API
- **ExifTool** - Embed metadata into files
- **Immich CLI** - Upload to Immich

Note: The script automatically detects the operating system and adapts accordingly.

2. Prerequisites

Component	Linux	Mac	Windows
Container Runtime	Docker or Podman	Docker Desktop	Docker Desktop / WSL2
X11	Yes (xauth)	Not required	Not required
Shell	Bash (native)	Bash (native)	WSL2 or Git Bash

3. Create Flickr API Key

1. Go to: <https://www.flickr.com/services/apps/create/>
2. Click 'Request an API Key' → 'Non-Commercial'
3. Fill out the form (Name: e.g., 'Flickr Backup')
4. Note down **API Key** and **API Secret**

4. Script Setup

Download and make executable:

```
# Download script (or copy from attachment)
chmod +x flickr-docker.sh

# Show system info
./flickr-docker.sh info
```

Build Docker image:

```
./flickr-docker.sh build
```

Create API configuration:

```
mkdir -p flickr-config
```

```
cat > flickr-config/.flickr_download << EOF
api_key: YOUR_API_KEY
api_secret: YOUR_API_SECRET
EOF
```

5. Authentication

Linux:

On Linux, a browser window opens automatically for OAuth authentication.

```
# Default (Chrome)
./flickr-docker.sh auth

# With Firefox
BROWSER=firefox ./flickr-docker.sh auth
```

Mac / Windows:

On Mac and Windows, the OAuth URL is displayed in the terminal. This URL must be opened manually in a browser.

```
./flickr-docker.sh auth

# Output:
# NOTE for Mac/Windows:
#   - A URL will be displayed in the terminal
#   - Open this URL manually in your browser
#   - After login, the callback is processed automatically
```

Important: After successful login, the token is saved to `flickr-config/.flickr_token`

6. Download Photos

All albums of a user:

```
# Download all albums
./flickr-docker.sh download your_flickr_username

# Or with full URL
./flickr-docker.sh download https://www.flickr.com/photos/username/
```

List albums:

```
./flickr-docker.sh list your_flickr_username
```

Single album:

```
./flickr-docker.sh album 72157622764287329
```

Photos are saved to **`./flickr-backup/`**, along with JSON metadata.

7. Embed Metadata

After downloading, the Flickr metadata (date, title, tags) must be embedded into the image files so that Immich recognizes them correctly.

Metadata embedding script:

```
#!/bin/bash
# embed_metadata.sh

BACKUP_DIR="./flickr-backup"

find "$BACKUP_DIR" -name "*.json" | while read json_file; do
    base="${json_file%.json}"

    # Find associated media file
    for ext in jpg jpeg png gif mp4 mov avi; do
        media_file="$base.${ext}"
        [ -f "$media_file" ] && break
    done

    [ ! -f "$media_file" ] && continue

    # Extract and embed date
    date_taken=$(jq -r ".datetaken" "$json_file")
    if [ "$date_taken" != "null" ]; then
        exif_date=$(echo "$date_taken" | sed "s/-/:/g")
        exiftool -overwrite_original \
            -DateTimeOriginal="$exif_date" \
            -CreateDate="$exif_date" \
            "$media_file"
    fi
done
```

Note: Videos require different EXIF tags (MediaCreateDate, TrackCreateDate).

8. Upload to Immich

Install Immich CLI:

```
npm install -g @immich/cli
```

Login:

```
immich login https://immich.example.com API_KEY
```

Upload:

```
# Upload all photos
immich upload ./flickr-backup --recursive

# With album creation
immich upload ./flickr-backup --recursive --album
```

9. Command Reference

Command	Description
build	Build Docker image
auth	Start OAuth authentication
download <user>	Download all albums of a user
album <id>	Download single album
list <user>	List albums of a user
shell	Interactive shell in container
test-browser [url]	Test X11 connection (Linux only)
info	Show system information
clean	Remove image and temp files

Environment Variables:

Variable	Description	Default
BROWSER	Browser for OAuth (Linux)	chrome
BROWSER	Browser for OAuth (Mac/Win)	<empty> = System

10. Troubleshooting

Browser doesn't open (Linux):

```
# Check DISPLAY
echo $DISPLAY

# Check xauth
xauth list

# Test
./flickr-docker.sh test-browser
```

Token invalid:

```
# Delete token and re-authenticate
rm flickr-config/.flickr_token
./flickr-docker.sh auth
```

Podman issues:

```
# The script detects Podman automatically and uses:
```

```
# --userns=keep-id (for X11 access)
# --security-opt label=disable (for SELinux)
```

Mac/Windows OAuth callback:

If the OAuth callback doesn't work, check if ports 8080-8100 are free. The script automatically publishes these ports for the callback server.

11. Directory Structure

Directory	Contents
./flickr-config/	API keys (.flickr_download) and token (.flickr_token)
./flickr-backup/	Downloaded photos and JSON metadata
./flickr-cache/	API cache for resume functionality