

# Immich to WinFE Documentation

A tool for instant filesystem-level access to photos from the Immich web-app (Immich's main desktop interface). *Ctrl+Alt+I* is the default hotkey.

## Menu:

Use *Ctrl+F* to navigate:

Requirements:

Guide:

    Final screen

Possible issues/debug

Integration details

Example working code:

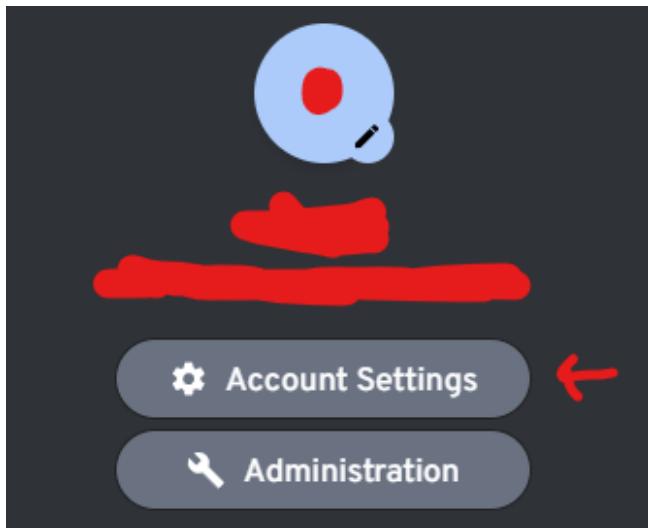
## Requirements

- The Windows operating system (I couldn't code something without testing it - not going to offer untested slop).
- Python 3x interpreter, (contains os, webbrowser, textwrap (specifically dedent), sys, re, json, urllib.request, subprocess modules)
- AutoHotkey (*installation is provided from within the script*)
- PowerShell, confirmed to work on PowerShell 5.1

## Guide

There's a good reason why I told you to read the docs first. In order for the seamless functionality of this shortcut to work, you will **need** an API key. (Not to worry, if the command line tool works, and your URL is highlighted but nothing happens, you can add this now)

Before you install the software, navigate to your Immich server and click first your user icon, and then account settings:



Click New API Key (3 rows from the top):

A screenshot of the "API Keys" section of the application. It contains four collapsed sections: "App Settings", "Account", "Account usage statistics", and "API Keys". The "API Keys" section is expanded, showing the sub-section "Manage your API keys". In the bottom right corner of this section is a blue rounded rectangular button labeled "New API Key".

Click asset.read. This is critical for technical reasons described below:

 New API Key

Name  
API Key

Permission

Select all

Search

activity

activity.create       activity.read       activity.update

activity.delete       activity.statistics

apiKey

apiKey.create       apiKey.read       apiKey.update

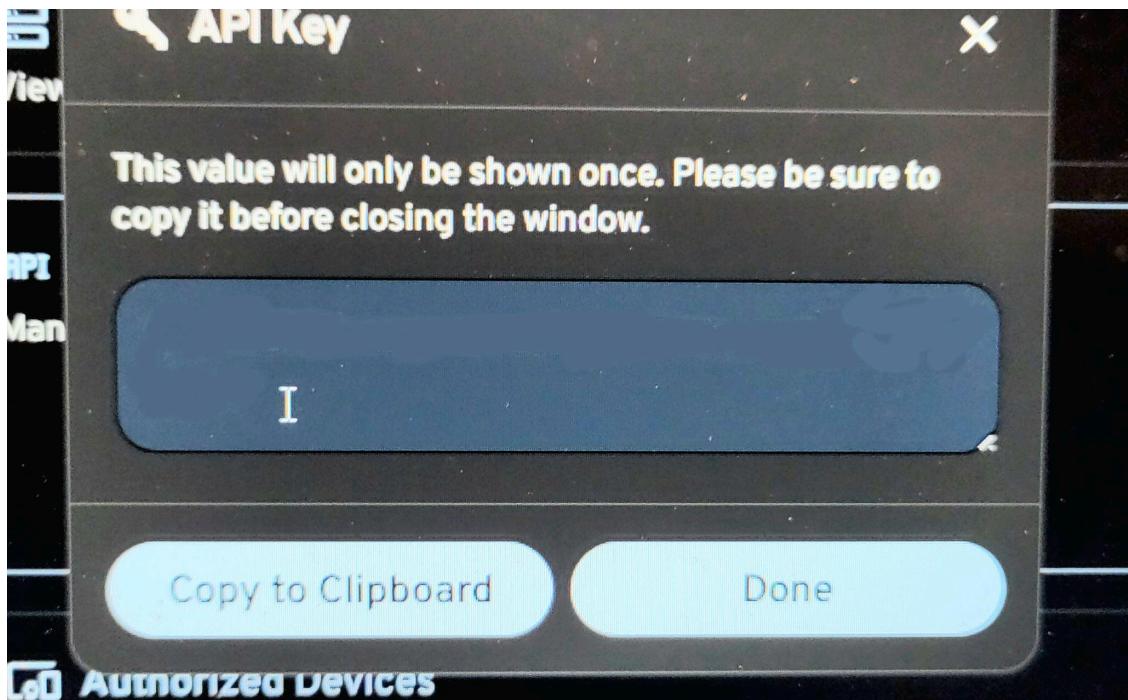
apiKey.delete

asset

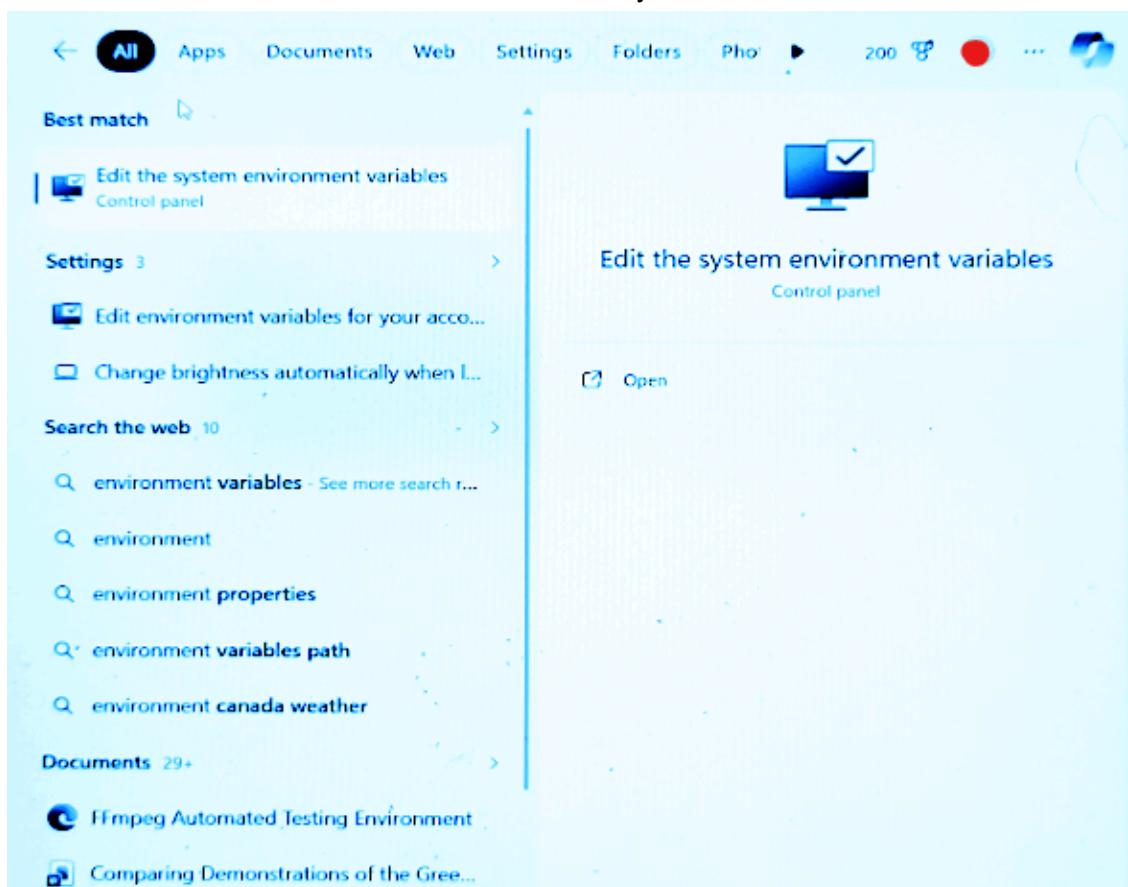
asset.read       asset.update       asset.delete

*The shortcut is wired to start with a URL for long-term compatibility purposes, and the URL only contains a random string of text (UUID). The software can't mind-read, so it needs the read authorization assigned to its key so it can ask for Immich's filepath. Without a key, this is a CLI only tool.*

When you receive your API key, **copy** using your cursor, the Copy to Clipboard feature, **AND** a reliable phone camera. This is the only time your key will be available. Some features don't work over http, so better safe than sorry.

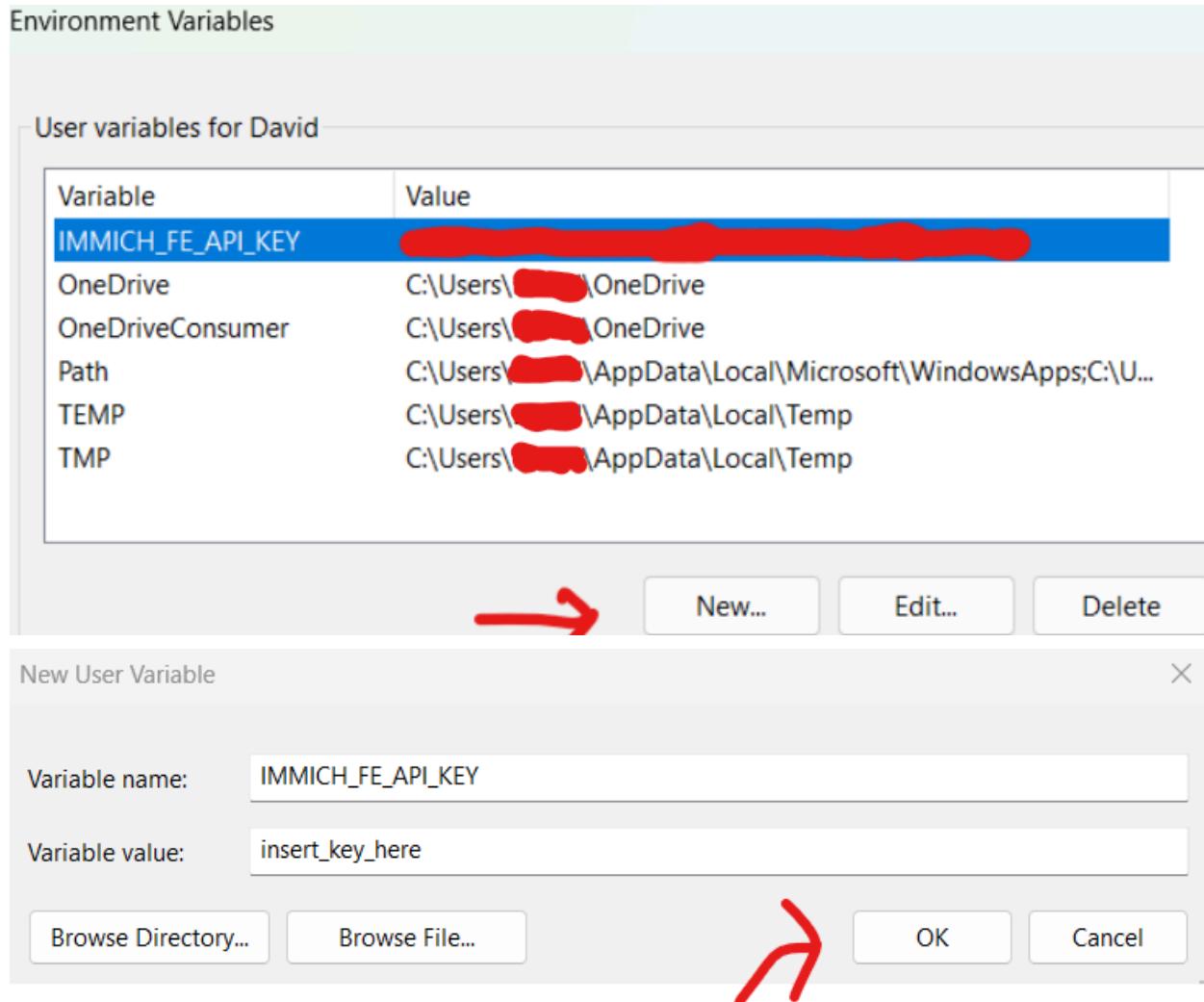


Go to Environment Variables via the Windows key or some other method.



*Environment variables are not stored at the file level (\*.py) and therefore slightly more secure against scrapers.*

Create a new User Environment Variable `IMMICH_FE_API_KEY` (exactly that, I already have it), and paste the key. The name is important because it is hard-coded into the script:



Then click OK at the bottom of the window again.

Now you're finished with setting up the environment variable. Download the install files and move them to your user folder, or a different working directory (folder) if you're feeling adventurous.

The default user folder (`C:\Users\NAME`) is the only one I've tested it on. You won't be able to change the directory of the final code without breaking and or manually modifying both final scripts.

**Open the Command Prompt.** Do not simply click on the script. If you felt adventurous, you will have to `-cd` to your new directory. Otherwise, just type `py Install.py` (try `python` if starting with `py` doesn't work) and the script will guide you the rest of the way.

You will be asked what your server IP is, which allows the script to ask the correct server for the path it needs. It will then ask you to input your Immich libraries. You'll input the mounted

locations (i.e. files/extlibrary or data/upload), followed by the true locations on your mapped SMB drive (i.e Z:\Extlibraryfolder or Z:\Immich\upload). This will produce immich\_to\_smb.py in the working directory, a CLI tool capable of using Immich URLs and file paths to guide you to the original file. It will then prompt you to type py install\_hotkey.py. Use python if that is the command which works on your computer.

Upon entering the hotkey installation, you'll be asked if you have Auto Hotkey, if not, it will download the installer and you will be asked to install Auto Hotkey, which is open source hotkey software. **The command line is still open, just type py install\_hotkey.py again**

It will then ask you whether you've used py or python. Enter whichever one has worked successfully. It will ask for the location of the first script, usually your working directory followed by the name of the script. It will then write out your AutoHotkey script in your Documents folder, the default location for AHK scripts.

### Final screen (if you're overwhelmed):

If you **already have** Auto Hotkey installed, you will likely have v2 installed. Since v1 is required for this script, you will need to go to the script in your Documents folder. You will need to open it with AutoHotkey. **Either way**, you'll also need to open it in order for it to run before restarting the computer.

To enable the hotkey on **login**, your command prompt will tell you to simply enter the powershell command into the command prompt (I didn't bother going through the whole process again). **Do not** use an actual powershell window. The text of the powershell command is the same, however.

Read the prompts in your command prompt window carefully and you'll probably be fine.

```
> C:\Users\████████\immich_to_smb.py

You're almost done. Go to \Documents\AutoHotkey and open the script with AutoHotkey.
Written to:
C:\Users\████████\Documents\AutoHotkey\ImmichSelect.ahk

If you have v2.0, you will be asked upon opening to download v.1.1

To enable the hotkey on startup, copy-paste the following into a new PowerShell tab or window:
powershell -NoProfile -Command "$WshShell = New-Object -ComObject WScript.Shell; $Shortcut = $WshShell.CreateShortcut('C
:Users\████████\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\ImmichSelect.lnk'); $Shortcut.TargetPath =
C:\Program Files\AutoHotkey\v1.1.37.02\AutoHotkeyU64.exe'; $Shortcut.Arguments = 'C:\Users\████████\Documents\AutoHotkey\Im
michSelect.ahk'; $Shortcut.WorkingDirectory = 'C:\Users\████████\Documents\AutoHotkey'; $Shortcut.Save()"

C:\Users\████████>
```

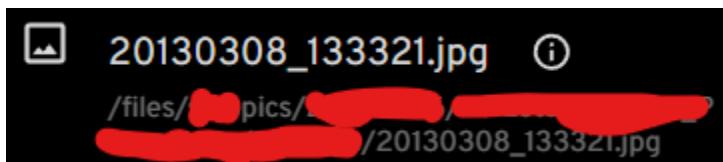
# Possible issues

**Full screen mode** - you'll have to toggle out of it to use this tool. A simple beep will indicate you are in full screen mode.

**Typos** - two approaches: delete the files and install fresh, or browse the code of the runtime files looking for typos.

**Lack of API key** - the URL cannot be decoded into a file path without access to the server API.

**Debug for lack of API key:** Open command prompt in the user (or working) directory. Copy the contents of the Info radio button, shown below, then open Command Prompt and type py or python immich\_to\_smb.py followed by the string you just copied (for instance python immich\_to\_smb.py /files/zfspics/2013/sliderphone/20130308\_133321.jpg). If it opens successfully, try with a URL. If the URL fails, the issue is probably a missing API key. If so, you will see "IMMICH\_FE\_API\_KEY environment variable not set".



**AutoHotkey or script folder installed** in non-default places - duh. Most normal users won't have to worry about this, however. If you still have a 32 bit machine or something and it doesn't work - feel free to modify the pathways and give me a PR / message me on Reddit.

Attempting to run install scripts **not located** in the user (or working) **directory**.

Lack of **software requirements**

Has not been tested with working directories other than **user folder**. In principle it should work.

## Integration details

Append --print-only to requests in the CLI or in other tools to receive the file path without opening extraneous Explorer windows.

The CLI tool can only accept **container filepaths**, **asset IDs**, and **URLs** (the default hotkey input). Names are not a reliable unique identifier in a large photo library spanning multiple folders.

## Example Working Code (runtime)

### Shortcut (ImmichSelect.lnk)

Target = "C:\Program Files\AutoHotkey\v1.1.37.02\AutoHotkeyU64.exe"

C:\Users\\_\_\_\_\Documents\AutoHotkey\ImmichSelect.ahk

Start in = "C:\Users\\_\_\_\_\Documents\AutoHotkey"

### AHK script (ImmichSelect.ahk)

; Immich → Explorer selector

```

; Hotkey: Ctrl + Alt + I

^!i::
    ; Save clipboard
    ClipSaved := ClipboardAll
    Clipboard := ""

    ; Focus address bar
    Send ^I
    Sleep 60

    Send ^c
    Sleep 60

    url := Clipboard

    ; Restore clipboard
    Clipboard := ClipSaved
    ClipSaved := ""

    if (url = "")
    {
        SoundBeep, 750
        return
    }

    python := "python"
    script := "C:\Users\____\immich_to_smb.py"

    Run, % python " """ script """ """ url """", , Hide
    return

```

### **Python runtime (immich\_to\_smb.py)**

```

import os
import sys
import re
import json
import urllib.request
import subprocess

```

```
IMMICH_URL = 'http://192.168.1.x:2283'
```

```
LIBRARY_MAP = {
```

```

os.path.normpath('/files/phonepics'):
os.path.normpath('E:\\AllTimeBackups\\2023-present\\redactedsmartphone\\DCIM'),
    os.path.normpath('/files/zfspics'): os.path.normpath('E:\\AllTimeBackups'),
    os.path.normpath('/data/upload'):
os.path.normpath('E:\\AllTimeBackups\\2023-present\\Immich\\upload'),
}

UUID_RE = re.compile(
    r"[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"
)

def translate_immich_path(immich_file_path: str) -> str:
    immich_file_path = os.path.normpath(immich_file_path)

    for immich_base, smb_base in LIBRARY_MAP.items():
        if immich_file_path.startswith(immich_base):
            rel = os.path.relpath(immich_file_path, immich_base)
            return os.path.join(smb_base, rel)

    raise ValueError("Path does not match any configured Immich library")

def extract_asset_id(input_str: str) -> str | None:
    m = UUID_RE.search(input_str)
    return m.group(0) if m else None

def fetch_asset_original_path(asset_id: str) -> str:
    api_key = os.environ.get("IMMICH_FE_API_KEY")
    if not api_key:
        raise RuntimeError("IMMICH_FE_API_KEY environment variable not set")

    url = f"{IMMICH_URL}/api/assets/{asset_id}"
    req = urllib.request.Request(
        url,
        headers={"x-api-key": api_key}
    )

    with urllib.request.urlopen(req) as resp:
        data = json.load(resp)

    if "originalPath" not in data:
        raise RuntimeError("Asset response missing originalPath")

    return data["originalPath"]

```

```
def select_in_explorer(path: str):
    if os.name != "nt":
        raise RuntimeError("Explorer selection is only supported on Windows")

    subprocess.run(
        ["explorer.exe", "/select,", os.path.normpath(path)],
        check=False
    )

def main():
    args = sys.argv[1:]

    print_only = False
    if args and args[0] == "--print-only":
        print_only = True
        args = args[1:]

    if len(args) != 1:
        print("Usage:")
        print("  py immich_to_smb.py [--print-only] <immich path | asset id | immich URL>")
        sys.exit(1)

    arg = args[0]

    try:
        if arg.startswith("/"):
            smb_path = translate_immich_path(arg)
        else:
            asset_id = extract_asset_id(arg)
            if not asset_id:
                raise RuntimeError("Input is not a valid Immich path, asset ID, or URL")

            immich_path = fetch_asset_original_path(asset_id)
            smb_path = translate_immich_path(immich_path)

        print(smb_path)

        if not print_only:
            select_in_explorer(smb_path)

    except Exception as e:
        print("Error:", e)
        sys.exit(2)
```

```
if __name__ == "__main__":
    main()
```