

Mathes Online Utilities - Shutdown & Restart Checklist

SHUTDOWN PROCEDURE

Step 1: Stop All Services (PowerShell as Admin)

```
powershell

# Stop the tasks
Stop-ScheduledTask -TaskName "Cloudflared Tunnel"
Stop-ScheduledTask -TaskName "Flask Excel Backend"
Stop-ScheduledTask -TaskName "Flask Full Edit Backend"

# Disable auto-restart on reboot
Disable-ScheduledTask -TaskName "Cloudflared Tunnel"
Disable-ScheduledTask -TaskName "Flask Excel Backend"
Disable-ScheduledTask -TaskName "Flask Full Edit Backend"
```

Step 2: Update Landing Page to Show Offline Status

Edit `D:\excel-web-interface\index.html` in the CONFIG section:

Find the Movies & Shows card and change:

```
javascript

{
  icon: "🎬",
  title: "Movies & Shows",
  description: "Full CRUD editor... (Currently offline)", // Add this
  buttons: [
    {
      text: "Full Editor",
      url: "...",
      style: "disabled" // Change from "primary" to "disabled"
    },
    {
      text: "Quick Add",
      url: "...",
      style: "disabled" // Change from "secondary" to "disabled"
    }
  ]
}
```

Then push the update:

```
powershell

cd D:\excel-web-interface
git add index.html
git commit -m "Mark Movies & Shows as offline"
git push origin main
```

Wait 1-2 minutes for GitHub Pages to deploy.

Step 3: Verify Everything is Stopped

```
powershell
```

```
# Check ports are no longer listening
```

```
netstat -ano | findstr :5000
```

```
netstat -ano | findstr :5001
```

```
# Should return nothing
```

```
# Check cloudflared isn't running
```

```
Get-Process -Name cloudflared -ErrorAction SilentlyContinue
```

```
# Should return nothing
```

```
# Check Flask apps aren't running
```

```
Get-Process -Name python -ErrorAction SilentlyContinue | Where-Object {$_.Path -like "**Python313*"} 
```

```
# Should return nothing related to your backends
```

Step 4: Test Shutdown Complete

Open browser and test:

- ❌ <https://api.ldmathes.cc/api/health> → Should timeout/fail
- ❌ <https://api-edit.ldmathes.cc/api/health> → Should timeout/fail
- ✅ <https://dennyrgood.github.io/> → Still loads (static, safe)

Status: All backend services are now offline. Your machine is no longer exposing anything to the internet.

Optional: Disable Cloudflare Tunnel Routes (Web Dashboard)

If you want to prevent any access attempts through Cloudflare:

- Go to **Cloudflare Zero Trust Dashboard**: <https://one.dash.cloudflare.com/>
- Navigate to **Networks** → **Tunnels**
- Click on **excel-backend** tunnel
- Go to **Public Hostname** tab
- You'll see two routes:
 - api.ldmathes.cc → <http://localhost:5000>
 - api-edit.ldmathes.cc → <http://localhost:5001>
- Click the **3 dots** (⋮) next to each route → **Delete**
- Confirm deletion

Note: Deleting routes is optional. If the local services are stopped, the routes won't work anyway. Delete them if you want to be extra cautious or if you're shutting down for an extended period.

✅ RESTART PROCEDURE

Important: Tunnel Management Strategy

Before restarting, read this section to understand how to prevent www conflicts.

Local vs. Remote Tunnel Management

Cloudflare Tunnels can be managed in two ways:

Locally-Managed (What You Had Before):

- Config stored in: (C:\Users\DrDen\.cloudflared\config.yml)
- Edit config file directly, restart tunnel to apply changes
- ❌ Can't edit routes in web dashboard (causes sync issues)
- ❌ Config only exists on your machine
- ⚠️ The mandatory catch-all rule ((http_status:404)) can intercept www traffic

Remotely-Managed (RECOMMENDED for restart):

- ✅ Config stored in Cloudflare cloud
- ✅ Edit routes via web dashboard (no file editing)
- ✅ Changes apply instantly (no tunnel restart needed)
- ✅ Can manage from anywhere
- ✅ Easier to prevent www conflicts by explicitly controlling routes

Why Remote Management is Better for Your Use Case:

1. Easier to pause/restart without editing files
2. Visual route management prevents accidental wildcard routes
3. No config.yml sync issues
4. Can add/remove routes without touching Windows machine
5. Easier to debug and see exactly what's configured

Step 1: Re-enable Task Scheduler Tasks (PowerShell as Admin)

```
powershell

# Re-enable the tasks
Enable-ScheduledTask -TaskName "Cloudflared Tunnel"
Enable-ScheduledTask -TaskName "Flask Excel Backend"
Enable-ScheduledTask -TaskName "Flask Full Edit Backend"
```

Step 2: Create Tunnel with Remote Management (RECOMMENDED)

Why Remote Management: Prevents the catch-all rule from intercepting www traffic, easier to manage, no config file editing needed.

Option A: Fresh Start with Remote-Managed Tunnel (Recommended)

Step 2A-1: Clean Up Old Tunnel (If Any)

```
powershell

# List existing tunnels
cloudflared tunnel list

# Delete old tunnel if it exists
cloudflared tunnel delete excel-backend

# OR
cloudflared tunnel delete 2d43262a-8fb5-4994-a261-5e6bd9729574

# Delete old local config
Remove-Item C:\Users\DrDen\.cloudflared\config.yml -ErrorAction SilentlyContinue
```

Step 2A-2: Create New Tunnel via Cloudflare Dashboard

1. Go to **Cloudflare Zero Trust Dashboard**: <https://one.dash.cloudflare.com/>
2. Navigate to **Networks** → **Tunnels**
3. Click **Create a tunnel**
4. Select **Cloudflared** connector
5. Name it: (or any name you prefer)
6. Click **Save tunnel**

Step 2A-3: Install Connector on Windows

7. In the dashboard, select **Windows** as the environment
8. Copy the command shown (will look like):

```
powershell

cloudflared.exe service install eyJhIjoN...very-long-token...
```

9. **IMPORTANT:** Run this command on your Windows machine in PowerShell as Admin
10. Wait for installation to complete
11. Back in the dashboard, click **Next**

Step 2A-4: Add Public Hostnames (CRITICAL - Only Add api and api-edit)

For api.ldmathes.cc:

- Click **Add a public hostname**
- Subdomain:
- Domain:
- Path: (leave blank)
- Type:
- URL:
- Click **Save hostname**

For api-edit.ldmathes.cc:

- Click **Add a public hostname**

- Subdomain:

- Domain:




- Path: (leave blank)

- Type:

- URL:

- Click **Save hostname**

CRITICAL - Do NOT add:

-  www.ldmathes.cc (leave this pointing to GitHub Pages)
-  ldmathes.cc (root domain)
-  *.ldmathes.cc (wildcard)

12. Click **Save tunnel**

Step 2A-5: Verify www Still Works BEFORE Starting Flask

Test immediately:



```
powershell

# Test www (should still work - GitHub Pages)
Invoke-WebRequest -Uri https://www.ldmathes.cc

# Test api (should 502 - tunnel works, Flask not running yet)
Invoke-WebRequest -Uri https://api.ldmathes.cc/api/health

# Test api-edit (should 502 - tunnel works, Flask not running yet)
Invoke-WebRequest -Uri https://api-edit.ldmathes.cc/api/health
```

In browser, verify:

-  <https://www.ldmathes.cc> → Should show GitHub Pages (**If this breaks, STOP HERE**)
-  <https://api.ldmathes.cc/api/health> → Should show 502 Bad Gateway (expected)

If www breaks at this stage:

1. The tunnel route configuration is wrong
2. Go back to the dashboard and verify you did NOT add www, root, or wildcard routes
3. Check Cloudflare DNS → Make sure www CNAME still points to

Option B: Re-enable Existing Local Tunnel (Not Recommended)

If you want to use the old local tunnel approach (not recommended due to catch-all issues):

```
powershell

# Re-enable the task
Enable-ScheduledTask -TaskName "Cloudflared Tunnel"

# Manually start
Start-ScheduledTask -TaskName "Cloudflared Tunnel"

# OR run manually to test
cloudflared tunnel run excel-backend
```

Then add DNS routes:

```
powershell

cloudflared tunnel route dns excel-backend api.ldmathes.cc
cloudflared tunnel route dns excel-backend api-edit.ldmathes.cc
```

WARNING: This approach may cause www conflicts due to the mandatory catch-all rule. Use Option A (remote management) instead.

Step 2B: Re-enable Flask Task Scheduler Tasks

```
powershell

Enable-ScheduledTask -TaskName "Flask Excel Backend"
Enable-ScheduledTask -TaskName "Flask Full Edit Backend"
```

Step 2C: Choose How to Start Services

Option A: Reboot (Recommended)

```
powershell

Restart-Computer
```

All services will auto-start on boot.

Option B: Manual Start (No Reboot)

```
powershell

# Start Flask tasks manually
Start-ScheduledTask -TaskName "Flask Excel Backend"
Start-ScheduledTask -TaskName "Flask Full Edit Backend"

# Wait 10 seconds for services to initialize
Start-Sleep -Seconds 10
```

Step 3: Update Landing Page to Show Online Status

Edit `D:\excel-web-interface\index.html` in the CONFIG section:

Find the Movies & Shows card and change:

```
javascript

{
  icon: "🎬",
  title: "Movies & Shows",
  description: "Full CRUD editor...", // Remove "(Currently offline)"
  buttons: [
    {
      text: "Full Editor",
      url: "...",
      style: "primary" // Change from "disabled" to "primary"
    },
    {
      text: "Quick Add",
      url: "...",
      style: "secondary" // Change from "disabled" to "secondary"
    }
  ]
}
```

Then push the update:

```
powershell

cd D:\excel-web-interface
git add index.html
git commit -m "Mark Movies & Shows as online"
git push origin main
```

Wait 1-2 minutes for GitHub Pages to deploy.

Step 4: Verify Services are Running

```
powershell

# Check Task Scheduler status
Get-ScheduledTask | Where-Object {$_.TaskName -like "*Flask*" -or $_.TaskName -like "*Cloudflared*"}
# All should show "Ready" or "Running"

# Check ports are listening
netstat -ano | findstr :5000
netstat -ano | findstr :5001
# Should show LISTENING for both ports

# Check cloudflared is running
Get-Process -Name cloudflared -ErrorAction SilentlyContinue
# Should show cloudflared.exe running

# Check Flask apps are running
Get-Process -Name python -ErrorAction SilentlyContinue | Where-Object {$_.Path -like "*Python313*"}
# Should show python.exe processes
```

Step 5: Test Endpoints

Backend Health Checks:

```
powershell
```

```
# Test port 5000 backend
```

```
Invoke-WebRequest -Uri https://api.ldmathes.cc/api/health
```




```
# Should return: {"status":"ok"}
```

```
# Test port 5001 backend
```

```
Invoke-WebRequest -Uri https://api-edit.ldmathes.cc/api/health
```

```
# Should return: {"status":"ok"}
```

Browser Tests:

-  <https://dennyrgood.github.io/> → Landing page loads, health check shows "online", Movies & Shows card active
-  <https://dennyrgood.github.io/excel-web-interface/> → Quick-add form works
-  <https://dennyrgood.github.io/MoviesShowsFullEdit/> → Full editor loads rows

Step 6: End-to-End Test

1. Go to <https://dennyrgood.github.io/excel-web-interface/>
2. Submit a test entry (Code: 999.1, Title: "Test Restart")
3. Verify entry appears in Excel file: `D:\OneDrive\MS\MoviesShows.xlsx`
4. Verify backup was created with timestamp

Status: All services restored and operational.

POST-MORTEM: Why www Broke and How to Prevent It

What Actually Caused the www Conflict

When the Flask backends were running before, `www.ldmathes.cc` stopped working and showed errors (Error 1033, 502, or 404). Here's what was happening:

The Root Cause: Mandatory Catch-All Rule

Cloudflare Tunnel configurations **require** a catch-all rule as the last ingress entry:

```
yaml
```

```
ingress:
```

```
- hostname: api.ldmathes.cc
```

```
  service: http://localhost:5000
```

```
- hostname: api-edit.ldmathes.cc
```

```
  service: http://localhost:5001
```

```
- service: http_status:404 # MANDATORY - can't be removed
```

How the catch-all intercepted www traffic:

1. Traffic to `www.ldmathes.cc` somehow reached the tunnel (via DNS misconfiguration)
2. Tunnel checked ingress rules:
 - Not `api.ldmathes.cc` ✗
 - Not `api-edit.ldmathes.cc` ✗
 - **Catch-all rule matched** ✓ → Returned 404 or error
3. GitHub Pages never received the traffic

Contributing Factors

One or more of these allowed www traffic to reach the tunnel:

1. **Wildcard DNS Record:** A CNAME for `*.ldmathes.cc` pointing to the tunnel
2. **Root Domain Route:** A route for `ldmathes.cc` (root) that caught www
3. **CLI Auto-Created Routes:** Running `cloudflared tunnel route dns` may have created extra DNS records
4. **Tunnel Public Hostname for Wildcard:** Adding `*.ldmathes.cc` in Zero Trust dashboard

The Solution: Explicit Route Control

Use remote management via Cloudflare Zero Trust Dashboard because:

1. ✓ You explicitly add only `api` and `api-edit` routes
2. ✓ No automatic wildcard or root routes are created
3. ✓ Visual confirmation of what's configured
4. ✓ Easy to audit and prevent conflicts
5. ✓ The catch-all rule still exists (mandatory) but www traffic never reaches it

DNS Must Be:

- ✓ CNAME | `www` | `dennyrgood.github.io` | Proxied (orange cloud)
- ✓ CNAME | `api` | `[tunnel-id].cfargotunnel.com` | Proxied
- ✓ CNAME | `api-edit` | `[tunnel-id].cfargotunnel.com` | Proxied
- ✗ NO root (`@`) pointing to tunnel
- ✗ NO wildcard (`*`) pointing to tunnel

Tunnel Routes Must Be:

- ✓ `api.ldmathes.cc` → `localhost:5000`
- ✓ `api-edit.ldmathes.cc` → `localhost:5001`
- ✗ NO `www.ldmathes.cc`
- ✗ NO `ldmathes.cc` (root)
- ✗ NO `*.ldmathes.cc` (wildcard)

Prevention Checklist

Before starting the tunnel, verify:

- ☐ www DNS CNAME points to `dennyrgood.github.io` (NOT tunnel)
- ☐ Tunnel only has explicit routes for `api` and `api-edit`
- ☐ No wildcard or root routes in tunnel config
- ☐ Test www BEFORE starting Flask backends

If www breaks at any stage:

1. Stop immediately
2. Check DNS records in Cloudflare dashboard
3. Check tunnel routes in Zero Trust dashboard
4. Remove any wildcard, root, or www routes from tunnel
5. Ensure www DNS points directly to GitHub Pages

Key Lesson

The tunnel catch-all is mandatory and cannot be removed. The only way to prevent it from catching www traffic is to ensure www traffic never reaches the tunnel at the DNS level. Remote management via dashboard makes this much easier to control and verify.



TROUBLESHOOTING

Services Won't Start

Check Task Scheduler:

```
powershell

# View task details
Get-ScheduledTask -TaskName "Flask Excel Backend" | Select-Object *
Get-ScheduledTask -TaskName "Flask Full Edit Backend" | Select-Object *
Get-ScheduledTask -TaskName "Cloudflared Tunnel" | Select-Object *
```

Manually start to see errors:

```
powershell

# Test Flask backends
cd D:\OneDrive\MS\
python excel_backend.py
# Watch for errors

# In new PowerShell window:
python excel_backend_full_edit.py
# Watch for errors

# Test Cloudflare Tunnel
D:\OneDrive\MS\cloudflared.exe tunnel run excel-backend
# Watch for errors
```

Ports Already in Use

```
powershell

# Find what's using the port
netstat -ano | findstr :5000
# Note the PID (last column)

# Kill the process
taskkill /PID [PID] /F

# Restart the service
Start-ScheduledTask -TaskName "Flask Excel Backend"
```

Tunnel Not Connecting

```
powershell

# Check tunnel config
Get-Content C:\Users\[YOUR-USERNAME]\.cloudflared\config.yml

# Verify credentials exist
Test-Path C:\Users\[YOUR-USERNAME]\.cloudflared\*.json

# Restart tunnel
Stop-ScheduledTask -TaskName "Cloudflared Tunnel"
Start-Sleep -Seconds 5
Start-ScheduledTask -TaskName "Cloudflared Tunnel"
```

GitHub Pages Not Loading

- Check repo at: <https://github.com/dennyrgood/excel-web-interface>
- Verify GitHub Pages is enabled: Settings → Pages
- Wait 1-2 minutes for deployment
- Hard refresh browser: Ctrl + F5

 QUICK REFERENCE

Current Configuration

Service	Port	Tunnel URL	Task Name
Quick-Add Backend	5000	api.ldmathes.cc	Flask Excel Backend
Full Editor Backend	5001	api-edit.ldmathes.cc	Flask Full Edit Backend
Cloudflare Tunnel	N/A	N/A	Cloudflared Tunnel

File Locations

Item	Location
Python	D:\Misc\Python313\python.exe
Backend Scripts	D:\OneDrive\MS\
Quick-Add	D:\OneDrive\MS\excel_backend.py
Full Editor	D:\OneDrive\MS\excel_backend_full_edit.py
Excel File	D:\OneDrive\MS\MoviesShows.xlsx
Cloudflared	D:\OneDrive\MS\cloudflared.exe
Tunnel Config	C:\Users\[USERNAME]\.cloudflared\config.yml
GitHub Repo	D:\excel-web-interface\

Key URLs

Purpose	URL
Landing Page	https://dennyrgood.github.io/
Quick-Add Form	https://dennyrgood.github.io/excel-web-interface/
Full Editor	https://dennyrgood.github.io/MoviesShowsFullEdit/
Port 5000 Health	https://api.ldmathes.cc/api/health
Port 5001 Health	https://api-edit.ldmathes.cc/api/health

OPTIONAL: COMPLETE REMOVAL

If you want to permanently remove the project (not just pause):

1. Delete Task Scheduler Tasks

```
powershell

Unregister-ScheduledTask -TaskName "Cloudflared Tunnel" -Confirm:$false
Unregister-ScheduledTask -TaskName "Flask Excel Backend" -Confirm:$false
Unregister-ScheduledTask -TaskName "Flask Full Edit Backend" -Confirm:$false
```

2. Delete Cloudflare Tunnel

```
powershell

# List tunnels
D:\OneDrive\MS\cloudflared.exe tunnel list

# Delete tunnel
D:\OneDrive\MS\cloudflared.exe tunnel delete excel-backend
```

Note: This permanently deletes the tunnel. If you restart later, you'll need to create a new tunnel with a new ID.

3. Remove DNS Records

Option A: Via Cloudflare Dashboard (Recommended)

1. Go to **Cloudflare Zero Trust Dashboard**: <https://one.dash.cloudflare.com/>
2. Navigate to **Networks** → **Tunnels**
3. Click on **excel-backend** tunnel
4. Go to **Public Hostname** tab
5. Delete both routes:
 - `api.ldmathes.cc`
 - `api-edit.ldmathes.cc`
6. Click the 3 dots (:) next to the tunnel name → **Delete tunnel**

Option B: Via Cloudflare DNS Dashboard

1. Go to **Cloudflare Dashboard**: <https://dash.cloudflare.com/>
2. Select your domain: `ldmathes.cc`
3. Go to **DNS** → **Records**
4. Find and delete CNAME records:
 - `api` → `[tunnel-id].cfargotunnel.com`
 - `api-edit` → `[tunnel-id].cfargotunnel.com`

4. Delete GitHub Pages (Optional)

- Go to <https://github.com/dennyrgood/excel-web-interface/settings/pages>
- Select "None" under Source
- Or delete entire repository

5. Clean Up Local Files (Optional)

```
powershell

# Remove backend scripts
Remove-Item D:\OneDrive\MS\excel_backend.py
Remove-Item D:\OneDrive\MS\excel_backend_full_edit.py
Remove-Item D:\OneDrive\MS\cloudflared.exe

# Remove tunnel credentials
Remove-Item -Recurse C:\Users\[USERNAME]\.cloudflared\

# Remove GitHub repo clone
Remove-Item -Recurse D:\excel-web-interface\
```



NOTES

- **Pause vs. Remove:** Disabling tasks = pause (can restart easily). Deleting tasks/tunnel = permanent removal.
- **Excel File Safety:** Always remains local at `D:\OneDrive\MS\MoviesShows.xlsx`, never deleted in any scenario.
- **GitHub Pages:** Stays live even when backends are stopped (forms just won't work).
- **Domain:** `ldmathes.cc` remains active and registered—only the subdomains stop working.
- **Security:** When shutdown, nothing is exposed to the internet. Your machine is secure.



LESSONS LEARNED

Great things accomplished in this project:

- Set up Cloudflare Tunnel for secure external access
- Built Flask REST APIs with full CRUD operations
- Preserved Excel formulas and formatting programmatically
- Created responsive web forms with GitHub Pages
- Implemented multi-port backend architecture
- Automated services with Windows Task Scheduler
- Built configuration-driven, modular frontend

This was a successful experiment. You learned a ton about web architecture, Flask, Cloudflare, Excel automation, and system administration. Well done! 🎉