

# PlexAnimeTools v2.0.0

A comprehensive PowerShell module for organizing anime, TV shows, cartoons, and movies for Plex Media Server. Automatically fetches metadata from MyAnimeList (Jikan API) and TMDb, renames files with proper episode titles, creates season folders, and downloads artwork.

---

## Documentation

### NEW: Interactive HTML Documentation!

We now provide **beautiful, interactive HTML documentation** with the best user experience:

```
powershell
```

```
# Open the interactive HTML documentation
```

```
Start-Process "PlexAnimeTools.html"
```

```
# Or simply double-click PlexAnimeTools.html in Windows Explorer
```

### Why use the HTML documentation?

-  **Interactive command cards** with copy-to-clipboard buttons
-  **Beautiful purple gradient design** that's easy on the eyes
-  **Mobile-responsive** layout works on any device
-  **Quick navigation** menu to jump to any section
-  **Live examples** with syntax highlighting
-  **One-click command copying** - no more typos!
-  **Organized by command type** for easy reference
-  **Visual feature cards** showing all capabilities

### Multiple Documentation Formats:

Choose the format that works best for you:

1. **PlexAnimeTools.html**  - Interactive HTML (RECOMMENDED - Best Experience!)
  2. **README.md** - Complete markdown documentation (this file)
  3. **README.pdf** - Printable PDF version
-

## Features

- **Automatic Content Detection** - Identifies anime, TV shows, cartoons, and movies
  - **API Integration** - Fetches metadata from Jikan (MyAnimeList) and TMDb
  - **Dual MAL Support** - Uses both Jikan (unofficial) and MAL Official APIs
  - **Episode Title Fetching** - Renames files with accurate episode titles
  - **Plex-Compatible Naming** - S01E01 format with customizable templates
  - **Season Folders** - Automatically creates proper folder structure
  - **Artwork Download** - Fetches and saves poster images
  - **Multiple Config Profiles** - Default, Plex-strict, and Fansub-chaos presets
  - **WhatIf Preview Window** - Visual preview of all changes with approval system
  - **GUI & CLI Modes** - Use graphical interface or command line
  - **Comprehensive Logging** - Detailed logs with full transcript capture
  - **Pipeline Support** - Process multiple folders efficiently
  - **Interactive Launcher** - Easy-to-use menu system
  - **Interactive HTML Documentation** - Beautiful, searchable command reference
- 

## Installation

### Method 1: Manual Installation

1. Download or clone the module to one of your PowerShell module paths:

```
powershell  
$env:PSModulePath -split ';'
```

2. Common locations:

- C:\Users\<YourName>\Documents\PowerShell\Modules\PlexAnimeTools
- C:\Program Files\PowerShell\Modules\PlexAnimeTools

3. Verify installation:

```
powershell
```

```
Import-Module PlexAnimeTools  
Get-Command -Module PlexAnimeTools
```

## Method 2: Import from Current Directory

```
powershell
```

```
Import-Module .\PlexAnimeTools.psd1
```

## Module Structure

```
PlexAnimeTools/  
    ├── PlexAnimeTools.psd1      # Module manifest  
    ├── PlexAnimeTools.psm1      # Module loader  
    ├── Start-PlexAnimeTools.ps1  # Launcher script  
    ├── README.md                # Markdown Documentation  
    ├── README.pdf               # PDF Documentation  
    ├── PlexAnimeTools.html      # ★ Interactive HTML Documentation (NEW!)  
    └── Config/  
        ├── default.json        # Default configuration  
        ├── plex-strict.json      # Strict Plex naming  
        └── fansub-chaos.json     # Preserve fansub tags  
    └── Public/  
        ├── Invoke-AnimeOrganize.ps1 # Main organization function  
        ├── Get-AnimeInfo.ps1       # Anime info retrieval  
        ├── Test-PlexScan.ps1      # Plex compatibility testing  
        └── Start-PlexGUI.ps1      # GUI launcher (UPDATED!)  
    └── Private/  
        ├── Detection.ps1        # Content type detection  
        ├── Jikan.ps1              # Jikan API (MAL unofficial)  
        ├── MALOfficial.ps1        # MAL Official API  
        ├── TMDb.ps1               # TMDb API functions  
        ├── Naming.ps1              # File naming utilities  
        ├── Plex.ps1                # Plex-specific functions  
        ├── Threading.ps1          # Logging & utilities  
        └── Preview.ps1             # WhatIf preview window  
    └── Logs/                  # Auto-created log directory  
        ├── PlexAnimeTools_*.log   # Main application logs  
        ├── Transcript_*.log     # Full console transcripts  
        └── PlexScan_Errors_*.log  # Error reports
```

---

## Quick Start

### View Documentation (Best Way to Learn!)

```
powershell

# Open the beautiful interactive HTML documentation
Start-Process "PlexAnimeTools.html"

# Browse all commands, examples, and features with one-click copy buttons!
```

### Launch with Start Script (Easiest)

```
powershell

# Navigate to PlexAnimeTools folder
cd C:\Path\To\PlexAnimeTools

# Run the launcher
.\Start-PlexAnimeTools.ps1
```

The launcher provides an interactive menu with options for:

- GUI mode
- CLI mode
- Quick start guide
- Plex library testing
- Module information
- README viewer

### GUI Mode (Recommended for Beginners)

```
powershell

Start-PlexGUI
```

### GUI Features (v2.0.0):

-  Process - Main organization function

- Anime Info - Search anime database
- Test Plex - Validate library compatibility
- View Main Log - Open latest log file
- View Transcript - Full console output
- Clear Old Logs - Remove old log files
- Test MAL API - Check MAL API status
- Export Errors - Generate error report
- Get Log Path - Copy log path to clipboard
- Error Summary - View error breakdown
- Open Logs - Open logs folder
- Help/README - Open documentation

## CLI Mode

```
powershell

# Preview mode with visual window (safe - no changes made)
Invoke-AnimeOrganize -Path "D:\Downloads\Anime" -OutputPath "D:\Plex\Anime" -WhatIf

# Process single folder
Invoke-AnimeOrganize -Path "D:\Downloads\Attack on Titan" -OutputPath "D:\Plex\Anime"

# Process multiple folders via pipeline
Get-ChildItem "D:\Downloads\Anime" -Directory |
  Invoke-AnimeOrganize -OutputPath "D:\Plex\Anime"

# Use strict Plex naming
Invoke-AnimeOrganize -Path "D:\Downloads" -OutputPath "D:\Plex" -ConfigProfile plex-strict
```

---

## WhatIf Preview System

When you run with `-WhatIf`, a visual preview window opens showing all planned changes:

### Preview Window Features:

- **Summary Statistics** - Total shows, files to rename, folders to create
- **Color-Coded Table:**
  -  Light Blue = Folders being created

-  Light Green = New file creation
-  Light Yellow = File moves/renames
- **Detailed View** - Shows current filename → new filename for every file
- **Export to CSV** - Save preview for offline review
- **Two-Step Approval:**
  1. Review all changes in the preview window
  2. Click **Proceed** to execute, or **Cancel** to abort

### **Example:**

```
powershell

# Run in preview mode
Invoke-AnimeOrganize -Path "I:\Anime\Pokemon" -OutputPath "I:\Anime" -WhatIf

# Preview window opens showing all planned changes
# Click "Proceed" button to execute all changes
# Click "Cancel" button to abort with no changes
```

The preview system ensures you never accidentally rename or move files incorrectly!

---

## **Configuration Profiles**

### **default.json**

- Balanced settings for most users
- Removes common fansub tags
- Uses English titles when available
- Standard Plex naming conventions

### **plex-strict.json**

- Maximum Plex compatibility
- Strips ALL tags and quality markers
- English titles only
- Minimal file extensions (mkv, mp4 only)
- Stops on first error

## **fansub-chaos.json**

- Preserves release group tags
  - Keeps quality information in filenames
  - Romaji titles preferred
  - Backs up original filenames
  - Processes all extensions
- 

## **API Configuration**

### **TMDb API Key (Required for TV Shows/Movies)**

1. Get a free API key: <https://www.themoviedb.org/settings/api>
2. Open your config file (e.g., `Config/default.json`)
3. Replace `"YOUR_TMDB_API_KEY_HERE"` with your actual key:

```
json
```

```
"TMDbAPIKey": "abc123yourkeyhere456"
```

### **Jikan API (MyAnimeList - Unofficial)**

- No API key required
- Free public API
- Automatically rate-limited to respect API guidelines
- Used by default for all anime lookups

### **MAL Official API (Optional - Enhanced Features)**

The module can also use MyAnimeList's official API for enhanced features:

1. **Create a MAL API Application (free):**
  - Go to: <https://myanimelist.net/apiconfig>
  - Click "Create ID"
  - App Name: `PlexAnimeTools` (or any name)
  - App Type: `web`
  - App Redirect URL: `http://localhost`

- Fill in other required fields
- Click "Submit"

## 2. Copy your Client ID and add it to your config:

```
json
```

```
"MALClientId": "your_client_id_here"
```

## 3. Test the connection:

```
powershell
```

```
Test-MALOfficialAPI
```

## Benefits of Official MAL API:

- More detailed anime information
- Personal list integration (your anime list)
- Real-time updates
- Higher rate limits
- Official support

---

## Available Functions

For detailed documentation with interactive examples, [open PlexAnimeTools.html](#) in your browser!

### Invoke-AnimeOrganize

Main function to organize media files.

#### Parameters:

- `-Path` - Source folder(s) containing media files
- `-OutputPath` - Destination Plex library location
- `-ConfigProfile` - Configuration to use (default, plex-strict, fansub-chaos)
- `-ForceType` - Override content detection (Auto, Anime, TV Series, Cartoon, Movie)
- `-WhatIf` - Preview mode - shows what would happen without making changes

## Examples:

```
powershell

# Preview changes with visual window
Invoke-AnimeOrganize -Path "D:\Downloads\Anime" -OutputPath "D:\Plex\Anime" -WhatIf

# Process with strict naming
Invoke-AnimeOrganize -Path "D:\Downloads" -OutputPath "D:\Plex" -ConfigProfile plex-strict

# Pipeline processing
Get-ChildItem "D:\Downloads" -Directory | Invoke-AnimeOrganize -OutputPath "D:\Plex"
```

## Get-AnimeInfo

Retrieves anime information from MyAnimeList.

## Examples:

```
powershell

# Search by title
Get-AnimeInfo -Title "Attack on Titan"

# Get details with episodes
Get-AnimeInfo -MalId 16498 -IncludeEpisodes

# Export episode list
Get-AnimeInfo -Title "One Piece" -IncludeEpisodes |
    Select-Object -ExpandProperty EpisodeList |
        Export-Csv "episodes.csv"
```

## Test-PlexScan

Validates Plex compatibility of organized media.

## Examples:

```
powershell
```

```
# Test single show  
Test-PlexScan -Path "D:\Plex\Anime\Attack on Titan"  
  
# Test all shows in library  
Get-ChildItem "D:\Plex\Anime" -Directory | Test-PlexScan
```

## Start-PlexGUI

Launches the graphical user interface.

### Example:

```
powershell  
  
Start-PlexGUI
```

## Utility Functions

### Show-LatestLog

Opens the most recent log file.

```
powershell  
  
Show-LatestLog -Type Main    # View main application log  
Show-LatestLog -Type Transcript # View full console output  
Show-LatestLog -Type Error    # View error report
```

### Get-LatestLog

Gets path to the most recent log file.

```
powershell  
  
$logPath = Get-LatestLog -Type Main
```

### Clear-OldLogs

Removes log files older than specified days.

```
powershell
```

```
Clear-OldLogs -Days 30 # Clean up logs older than 30 days
```

## Test-MALOfficialAPI

Tests if MAL Official API is configured and working.

```
powershell
```

```
Test-MALOfficialAPI
```

## Best Practices

### 1. Read the HTML Documentation First

```
powershell
```

```
Start-Process "PlexAnimeTools.html"
```

The interactive documentation has everything you need with copy-paste examples!

### 2. Always test with -WhatIf first

```
powershell
```

```
Invoke-AnimeOrganize -Path "..." -OutputPath "..." -WhatIf
```

Review the visual preview window before proceeding.

### 3. Use the launcher script for easy access

```
powershell
```

```
.\Start-PlexAnimeTools.ps1
```

### 4. Monitor log files

```
powershell
```

```
Show-LatestLog -Type Main
```

## 5. Regular log cleanup

```
powershell
```

```
Clear-OldLogs -Days 30
```

## Examples & Use Cases

### Organize Downloaded Anime

```
powershell
```

```
# Preview first
```

```
Invoke-AnimeOrganize -Path "D:\Downloads\[SubsPlease] Attack on Titan" `  
-OutputPath "D:\Plex\Anime" -WhatIf
```

```
# Process after verification
```

```
Invoke-AnimeOrganize -Path "D:\Downloads\[SubsPlease] Attack on Titan" `  
-OutputPath "D:\Plex\Anime"
```

### Batch Process Multiple Shows

```
powershell
```

```
# Get all anime folders and process
```

```
Get-ChildItem "D:\Downloads\Anime" -Directory |  
Invoke-AnimeOrganize -OutputPath "D:\Plex\Anime" -WhatIf
```

### Export Anime Information

```
powershell
```

```
# Get comprehensive anime info
```

```
$info = Get-AnimeInfo -Title "Demon Slayer" -IncludeEpisodes
```

```
# Export episode list
```

```
$info.EpisodeList | Export-Csv "demon_slayer_episodes.csv" -NoTypeInformation
```

For more examples, see the [PlexAnimeTools.html](#) interactive documentation!

## Requirements

- **PowerShell 5.1 or higher**
  - **Windows** (Uses Windows Forms for GUI)
  - **.NET Framework** (Included with Windows)
  - **Internet connection** (For API access)
  - **TMDb API Key** (For TV shows/movies - free)
- 

## FAQ

**Q: Do I need API keys?** A: TMDb API key required for TV shows/movies. Jikan (anime) is free without key.

**Q: Will this delete my original files?** A: By default, files are moved (not copied). Use `-WhatIf` to preview. Enable `BackupOriginalNames` in config for safety.

**Q: Can I undo changes?** A: No built-in undo. Always test with `-WhatIf` first and keep backups.

**Q: How do I view the HTML documentation?** A: Double-click `PlexAnimeTools.html` or run `Start-Process "PlexAnimeTools.html"` from PowerShell.

**Q: Where can I find detailed examples?** A: Open the interactive HTML documentation for comprehensive examples with copy buttons!

---

## Troubleshooting

For detailed troubleshooting, open **PlexAnimeTools.html** for interactive help!

### Quick Fixes:

#### "No results found" Error:

- Check folder name is recognizable
- Try manual search with `Get-AnimeInfo -Title "Show Name"`
- Verify internet connection

#### View Detailed Logs:

```
powershell
```

```
# View latest main log  
Show-LatestLog -Type Main  
  
# View console transcript  
Show-LatestLog -Type Transcript
```

---

## Version History

### v2.0.0 (Current)

- Complete rewrite as PowerShell module
  - Added GUI interface with 13 command buttons
  - Multiple configuration profiles
  - TMDb integration for TV/movies
  - WhatIf support with visual preview window
  - Pipeline support
  - Comprehensive logging with transcript capture
  - Episode title fetching
  - Artwork download
  - Plex compatibility testing
  - **NEW: Interactive HTML documentation** ★
  - **NEW: Enhanced GUI with all utility commands**
  - **NEW: Export error reports**
  - **NEW: Quick log access buttons**
- 

## Support

-  [Open PlexAnimeTools.html](#) for interactive documentation (RECOMMENDED!)
- Check logs in Logs directory for errors
- Use `(-WhatIf)` to preview changes
- Test with small batches first
- Review README.md for examples
- Use `(Get-ErrorSummary)` for detailed error analysis

---

Happy organizing! 📺🎬🎊

---

## Links

- [Interactive HTML Documentation](#) ⭐ (Open this first!)
- [Jikan API Documentation](#)
- [TMDb API](#)
- [MyAnimeList API Config](#)
- [Plex Naming Guidelines](#)