

MVidarr Enhanced

Professional Music Video Management System - Complete Overview & Features

 **Production Ready**

 **Mobile Optimized**

 **Docker Ready**

 **High Performance**

Professional Music Video Management System

Complete Documentation Suite

Generated: July 23, 2025

Version 2.0 Enhanced Edition

🎵 MVidarr Enhanced - Professional Music Video Management System

Status Production Ready

Production Ready

Features Complete

All Features Complete

Docker Ready

Docker Ready

Python 3.12

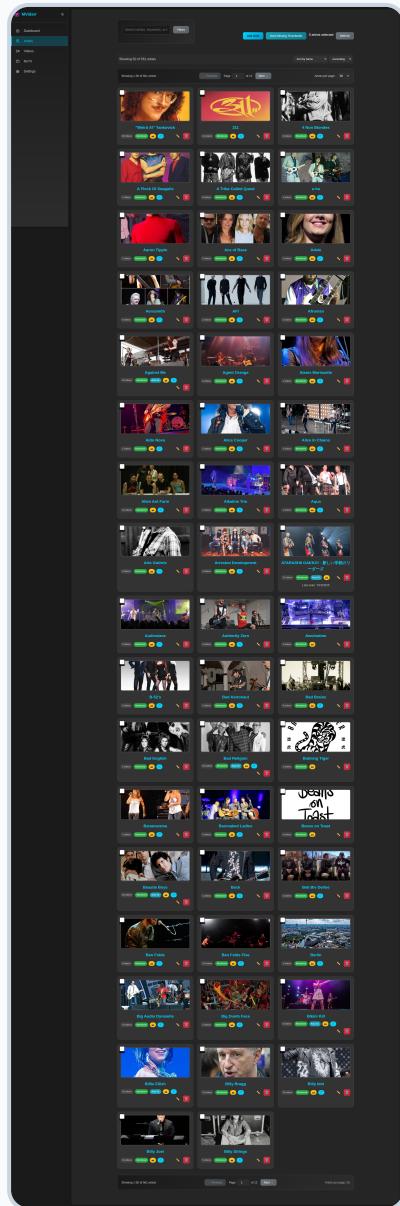
Python 3.12

MariaDB 11.4+

[MariaDB](#)

⌚ Professional music video management system inspired by Lidarr

Comprehensive artist-centric platform for downloading, organizing, and streaming music video collections.



MVidarr Enhanced Dashboard

The MVidarr Enhanced main dashboard showing your music video collection at a glance



Quick Start (30 seconds)

Docker Deployment (Recommended)

```
# 1. Get the code  
git clone <repository-url> && cd mvidarr  
  
# 2. Configure deployment  
cp docker-config.yml.sample docker-config.yml  
nano docker-config.yml # Edit paths and API keys  
  
# 3. Create storage directories  
mkdir -p /path/to/your/{musicvideos,database,thumbnails}  
sudo chown -R 1001:1001 /path/to/your/storage/  
  
# 4. Deploy with Docker  
docker-compose --env-file docker-config.yml -f docker-compose.production.yml up -d  
  
# 5. Access your system  
open http://localhost:5000
```



That's it! Your professional music video management system is running.



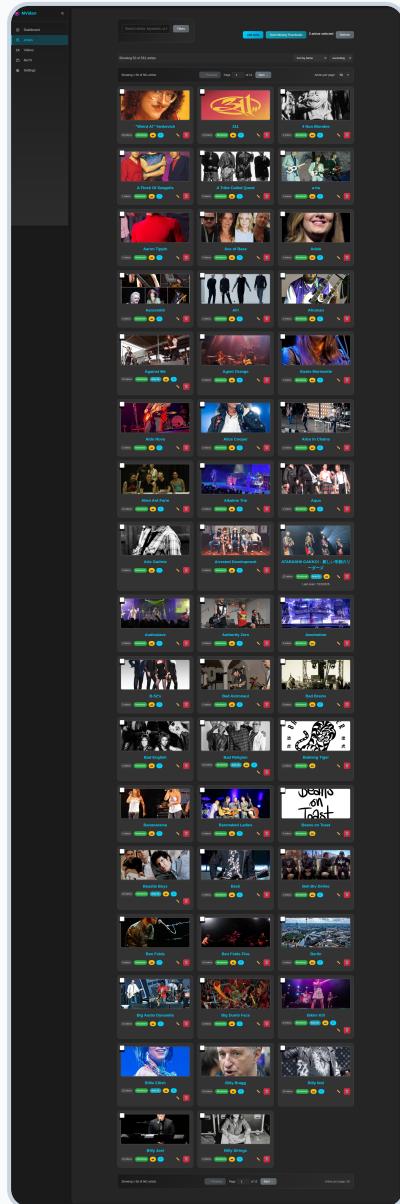
Why MVidarr Enhanced?

Built for Music Video Enthusiasts

Unlike generic media managers, MVidarr Enhanced is **specifically designed** for music video collections with:

- **Artist-centric organization** - Manage by artist, not just individual files
- **Dual-source discovery** - IMVDb + YouTube integration for comprehensive coverage
- **Professional metadata** - Rich video information with genres, directors, and more

- **Smart automation** - Automatic discovery and organization of new releases



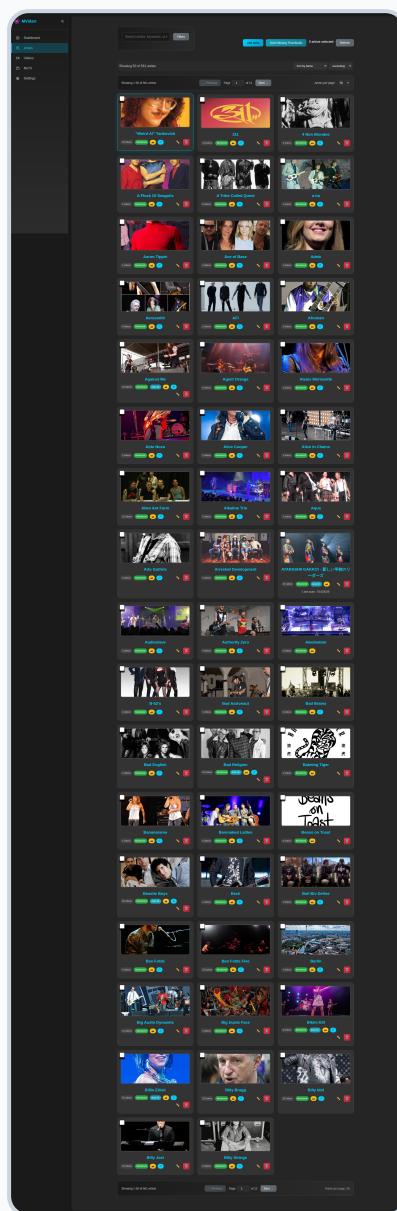
Artist Management Interface

Comprehensive artist management with metadata, thumbnails, and bulk operations

Complete Video Management Workflow

1.  **Discovery** - Find videos from your tracked artists automatically
2.  **Download** - Queue and monitor high-quality video downloads
3.  **Organization** - Intelligent file naming and folder structure

-
- 4.  **Streaming** - Built-in player with progress tracking
 - 5.  **Thumbnails** - Beautiful artwork from multiple sources



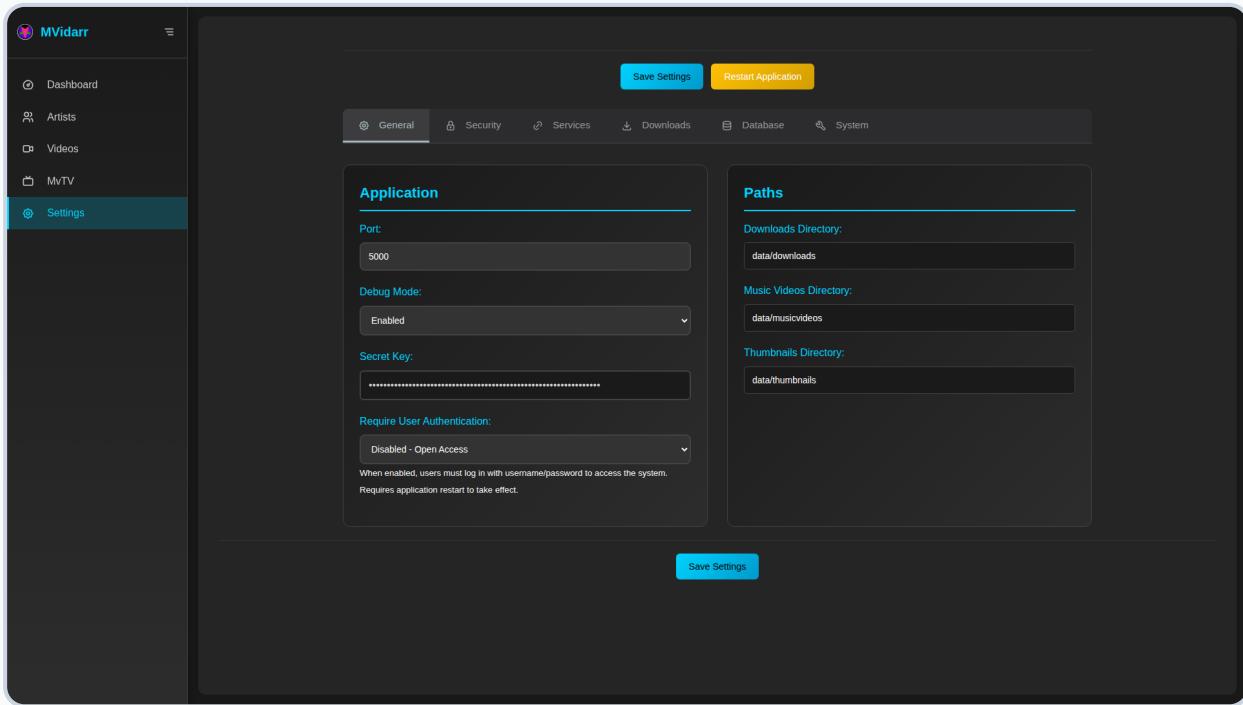
Video Discovery

Discover and import videos from YouTube and IMVDb with intelligent filtering

Key Features

Modern, Responsive Interface

- **Dark/Light themes** with 10+ color schemes
- **Mobile-optimized** responsive design
- **Real-time updates** and progress tracking
- **Professional UI** with smooth animations



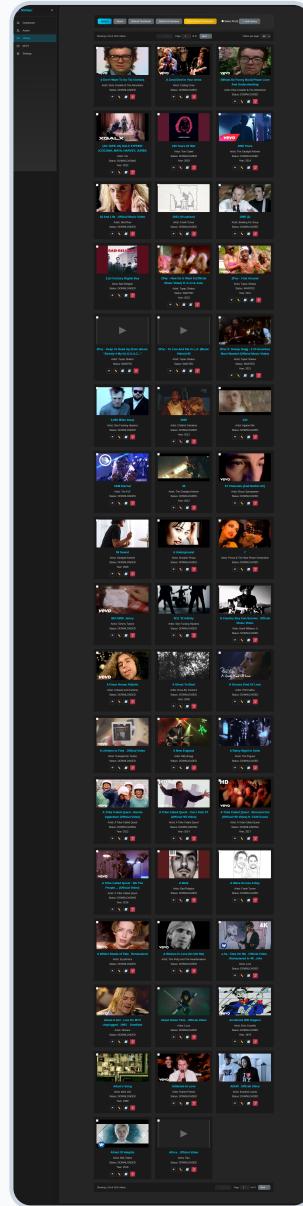
The screenshot shows the MVidarr Enhanced application's settings page. On the left is a sidebar with links: Dashboard, Artists, Videos, MyTV, and Settings (which is selected). The main area has tabs for General, Security, Services, Downloads, Database, and System. Under General, there are sections for Application (Port: 5000, Debug Mode: Enabled, Secret Key: masked, Require User Authentication: Disabled - Open Access) and Paths (Downloads Directory: data/downloads, Music Videos Directory: data/musicvideos, Thumbnails Directory: data/thumbnails). At the bottom are Save Settings and Restart Application buttons.

Theme Selector

Choose from multiple themes including dark mode and color variants

Advanced Search & Discovery

- **Real-time search** with autocomplete suggestions
- **Multi-criteria filtering** (genre, year, status, quality)
- **Bulk operations** with professional multi-select interface
- **Smart pagination** for large collections



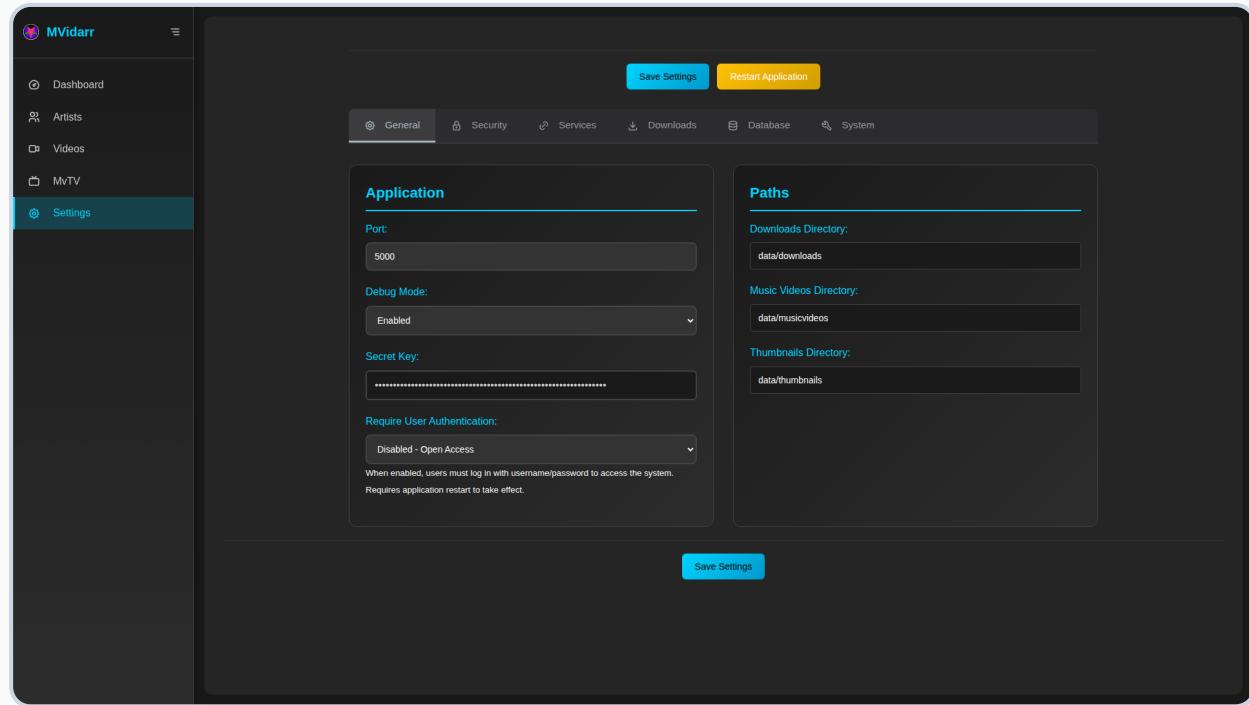
Advanced Search

Powerful search with real-time filtering and bulk selection



Professional Thumbnail Management

- **Multi-source search** (IMDb, YouTube, Wikipedia)
- **Manual upload** with drag-and-drop
- **Auto-cropping** and multiple sizes
- **Intelligent caching** with WebP optimization



Thumbnail Management

Professional thumbnail management with multiple sources and sizes

System Health & Monitoring

- **Comprehensive diagnostics** for all system components
- **Download progress tracking** with detailed statistics
- **Performance metrics** and system resource monitoring
- **Automated recovery** for failed operations

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

System Health

Monitor system health, performance, and download statistics

Installation Methods

Method 1: Docker (Recommended)

 **Easiest setup, production-ready, includes database**

 **Docker Installation Guide**

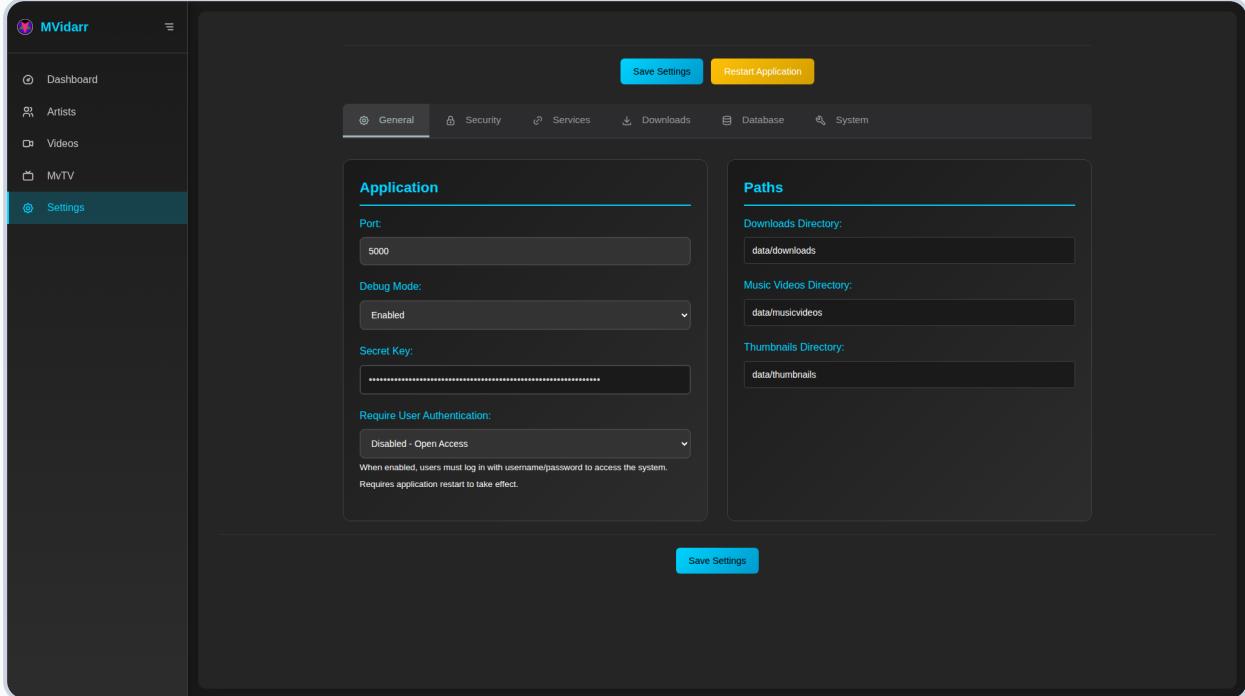
Prerequisites

- Docker and Docker Compose installed
- 2GB+ free disk space
- API keys (optional but recommended)

Step 1: Download and Configure

```
# Clone the repository
git clone <repository-url>
cd mvidarr

# Copy configuration template
cp docker-config.yml.sample docker-config.yml
```



Docker Config

Step 2: Edit Configuration

```
nano docker-config.yml
```

Required settings:

```
# Storage paths (create these directories)
MUSIC_VIDEOS_PATH=/home/youruser/musicvideos
DATABASE_FOLDER=/home/youruser/mvidarr/database

# Security (change these!)
DB_PASSWORD=your_secure_password
SECRET_KEY=your_long_random_secret_key

# API Keys (get from respective services)
IMVDB_API_KEY=your_imvdb_api_key
YOUTUBE_API_KEY=your_youtube_api_key
```

Step 3: Create Directories

```
# Create storage directories
sudo mkdir -p /home/youruser/{musicvideos,mvidarr/database,mvidarr/thumbnails}

# Set proper permissions (container runs as user 1001)
sudo chown -R 1001:1001 /home/youruser/musicvideos /home/youruser/mvidarr/
```

Step 4: Deploy

```
# Start all services
docker-compose --env-file docker-config.yml -f docker-compose.production.yml up -d

# Check status
docker-compose -f docker-compose.production.yml ps
```

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Docker Deployment

Step 5: Access

- **Web Interface:** `http://localhost:5000`
- **Database** (if needed): `localhost:3306`

Method 2: Local Installation

 **Full control, manual setup required**

 **Local Installation Guide**

Prerequisites

- Python 3.12+
- MariaDB 11.4+
- FFmpeg
- Git

Step 1: System Dependencies

```
# Ubuntu/Debian
sudo apt update && sudo apt install -y \
    python3.12 python3.12-venv python3-pip \
    mariadb-server mariadb-client \
    ffmpeg git curl

# CentOS/RHEL
sudo yum install -y python3 python3-venv python3-pip \
    mariadb-server mariadb \
    ffmpeg git curl
```

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

System Dependencies

Step 2: Database Setup

```
# Start MariaDB
sudo systemctl start mariadb
sudo systemctl enable mariadb

# Secure installation
sudo mysql_secure_installation

# Create database
sudo mysql -u root -p
```

```
CREATE DATABASE mvidarr_enhanced
    CHARACTER SET utf8mb4
    COLLATE utf8mb4_unicode_ci;

CREATE USER 'mvidarr'@'localhost'
    IDENTIFIED BY 'your_secure_password';

GRANT ALL PRIVILEGES ON mvidarr_enhanced.*
    TO 'mvidarr'@'localhost';

FLUSH PRIVILEGES;
EXIT;
```

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Database Setup

Step 3: Application Setup

```
# Clone and setup
git clone <repository-url>
cd mvidarr

# Create virtual environment
python3.12 -m venv venv
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt
```

Step 4: Configuration

```
# Copy environment template  
cp .env.example .env  
nano .env
```

Edit **.env** with your settings:

```
# Database  
DB_HOST=localhost  
DB_PORT=3306  
DB_NAME=mvidarr_enhanced  
DB_USER=mvidarr  
DB_PASSWORD=your_secure_password  
  
# Application  
SECRET_KEY=your_very_long_random_secret_key  
PORT=5000  
DEBUG=false  
  
# API Keys  
IMVDB_API_KEY=your_imvdb_api_key  
YOUTUBE_API_KEY=your_youtube_api_key  
  
# Paths  
MUSIC_VIDEOS_PATH=/home/youruser/musicvideos
```

Step 5: Initialize and Start

```
# Initialize database  
python -c "from src.database.connection import DatabaseManager; DatabaseManager().create()  
  
# Start application  
python app.py
```

Not Found

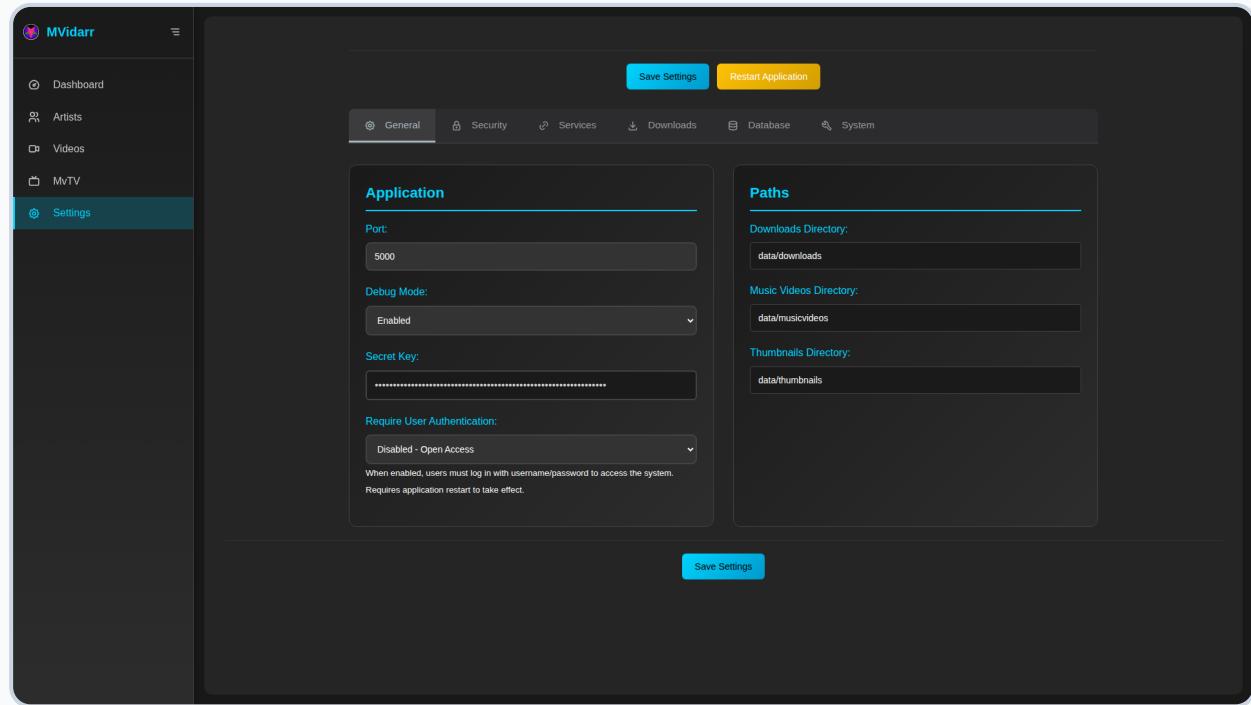
The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Local Installation

Getting Started Guide

1. Initial Setup

After installation, access <http://localhost:5000> for first-time setup:



Initial Setup

Complete the initial setup wizard to configure your system

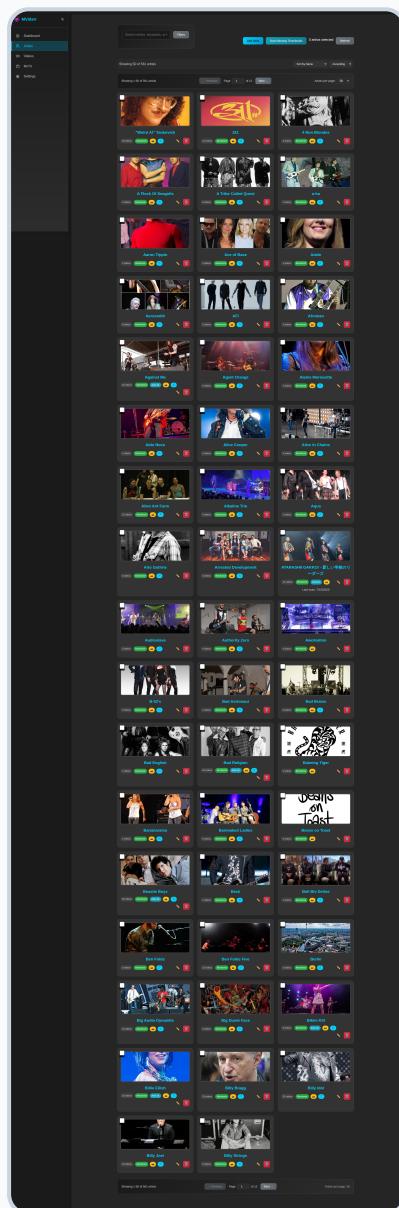
1. Configure API Keys (Settings → External Services)

- IMVDb API: Get from <https://imvdb.com/developers/api>
- YouTube API: Get from <https://console.developers.google.com>

2. Set Download Paths (Settings → Downloads)

- Music videos storage location
- Quality preferences
- Organization structure

2. Add Your First Artist



Add Artist

Step-by-step artist addition process

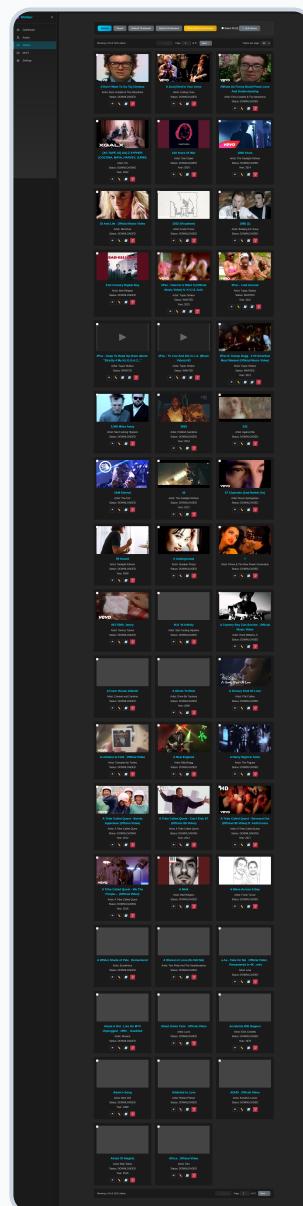
1. **Navigate to Artists** → Click “Add Artist”
2. **Search for Artist** - Use IMVDb search for best results
3. **Select Match** - Choose the correct artist from results

4. Configure Monitoring:

- Monitor new releases
- Automatically download official videos
- Search for existing videos

5. Click "Add Artist" - Artist will be added and videos discovered

3. Discover and Download Videos

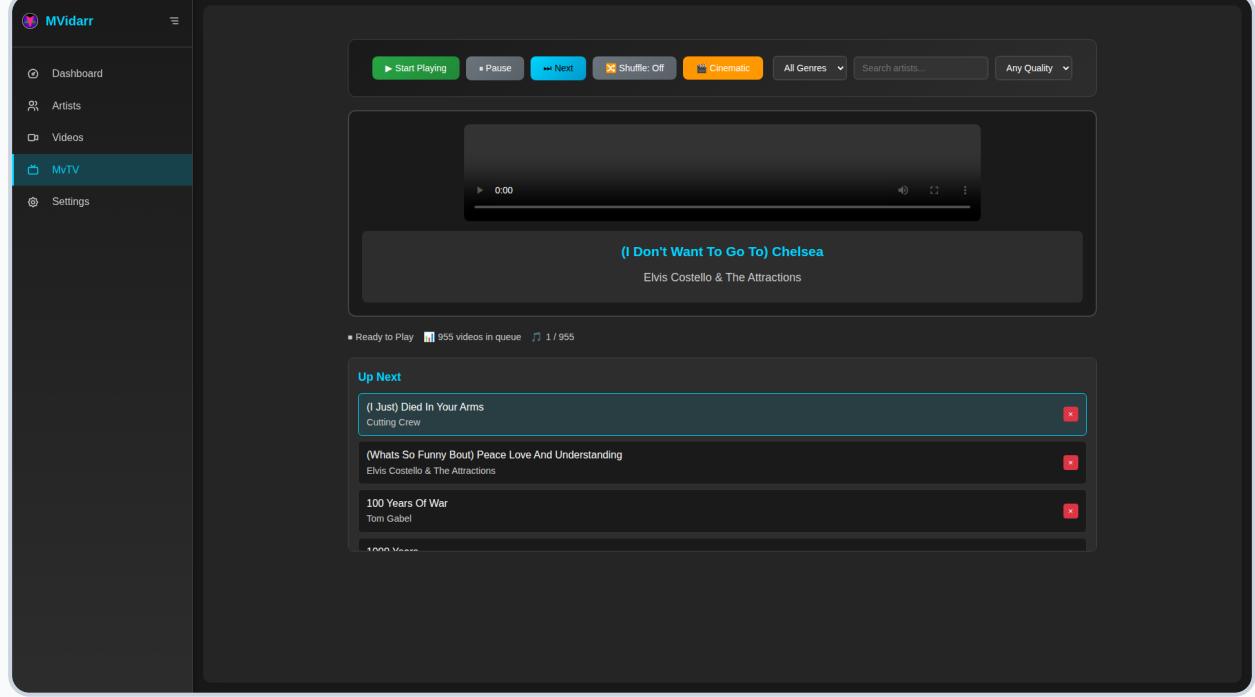


Video Management

Complete video discovery and download workflow

1. **Video Discovery** - Automatic discovery runs for new artists
2. **Review Videos** - Go to Videos page to see discovered content
3. **Filter Content** - Use filters to find specific types:
 -  Official Music Videos
 -  Live Performances
 -  Lyric Videos
 -  Behind the Scenes
4. **Queue Downloads** - Select videos and click “Download Selected”
5. **Monitor Progress** - Track downloads in real-time

4. Organize and Stream



The screenshot shows the MVidarr Enhanced streaming interface. On the left is a sidebar with navigation links: Dashboard, Artists, Videos, MvTV (which is selected), and Settings. The main area features a video player at the top with controls for Start Playing, Pause, Next, Shuffle Off (set to Cinematic), All Genres, Search artists..., and Any Quality. Below the player is a video frame showing a scene from a music video. The video information is: (I Don't Want To Go To) Chelsea by Elvis Costello & The Attractions. At the bottom of the video frame, it says "Ready to Play" and "955 videos in queue". Below the video frame is a "Up Next" section containing three items: (I Just) Died In Your Arms by Cutting Crew, (Whats So Funny Bout) Peace Love And Understanding by Elvis Costello & The Attractions, and 100 Years Of War by Tom Gabel.

Streaming Interface

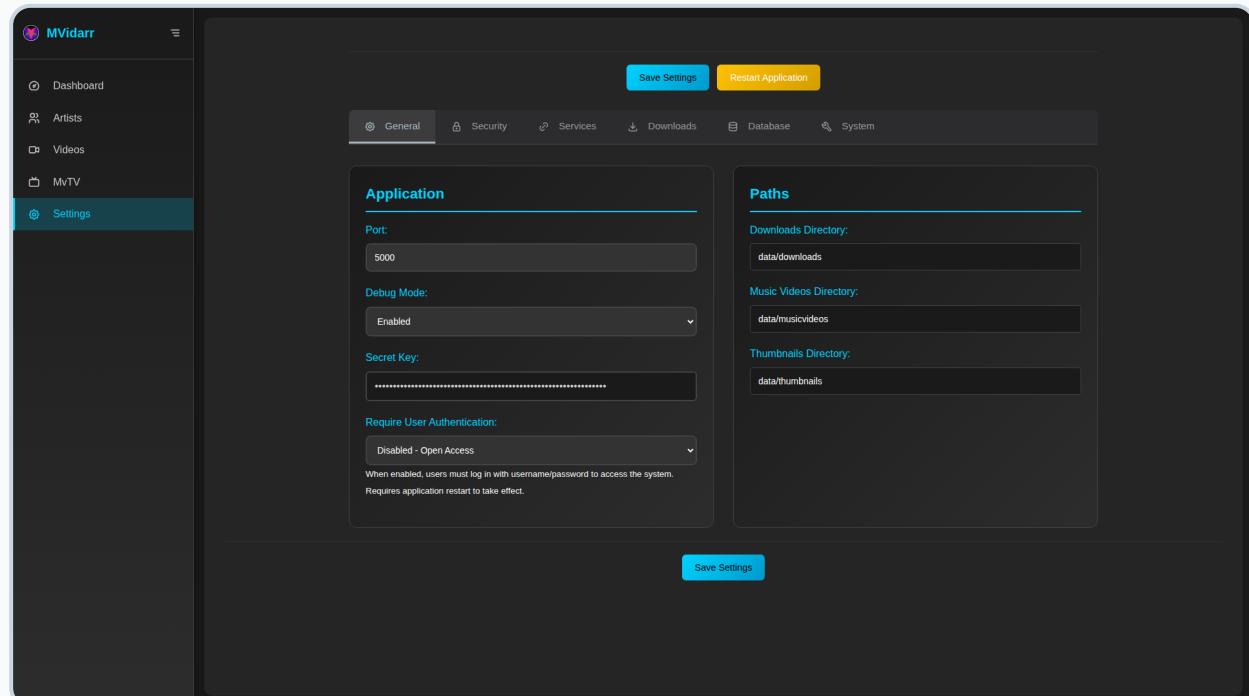
Built-in video player with continuous play features

- **Automatic Organization** - Videos organized by artist/album

- **Built-in Player** - Stream directly from the interface
- **Continuous Play** - MvTV mode for uninterrupted viewing
- **Progress Tracking** - Resume where you left off

🔧 Configuration Guide

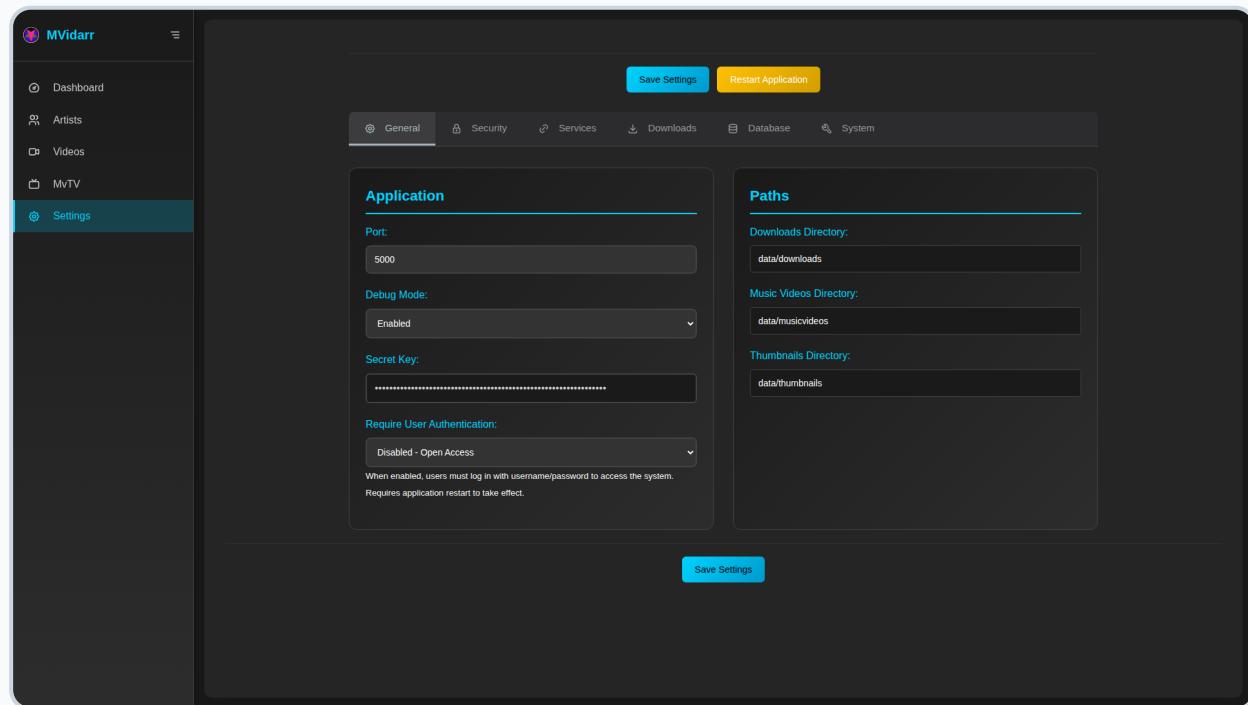
Essential Settings



Settings Overview

Comprehensive settings interface organized by category

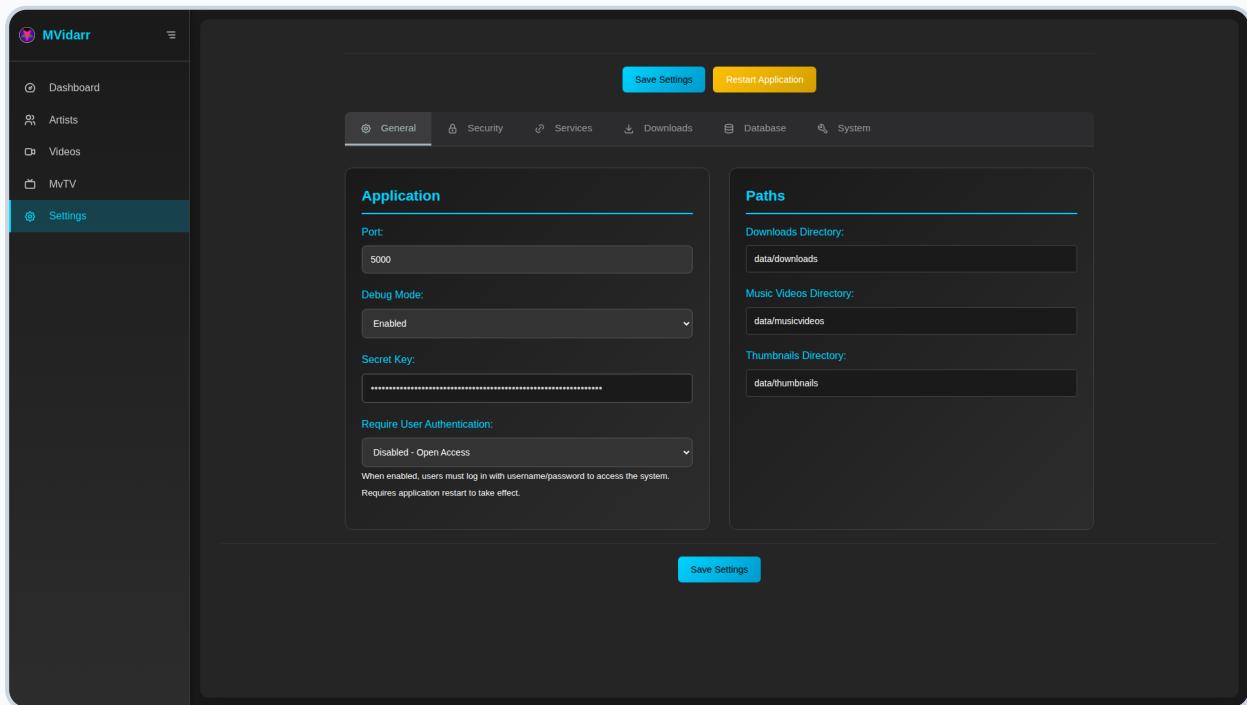
1. External Services



External Services

- **IMVDb API Key** - For video metadata and discovery
- **YouTube API Key** - For enhanced YouTube integration
- **API Rate Limits** - Automatic rate limiting configuration

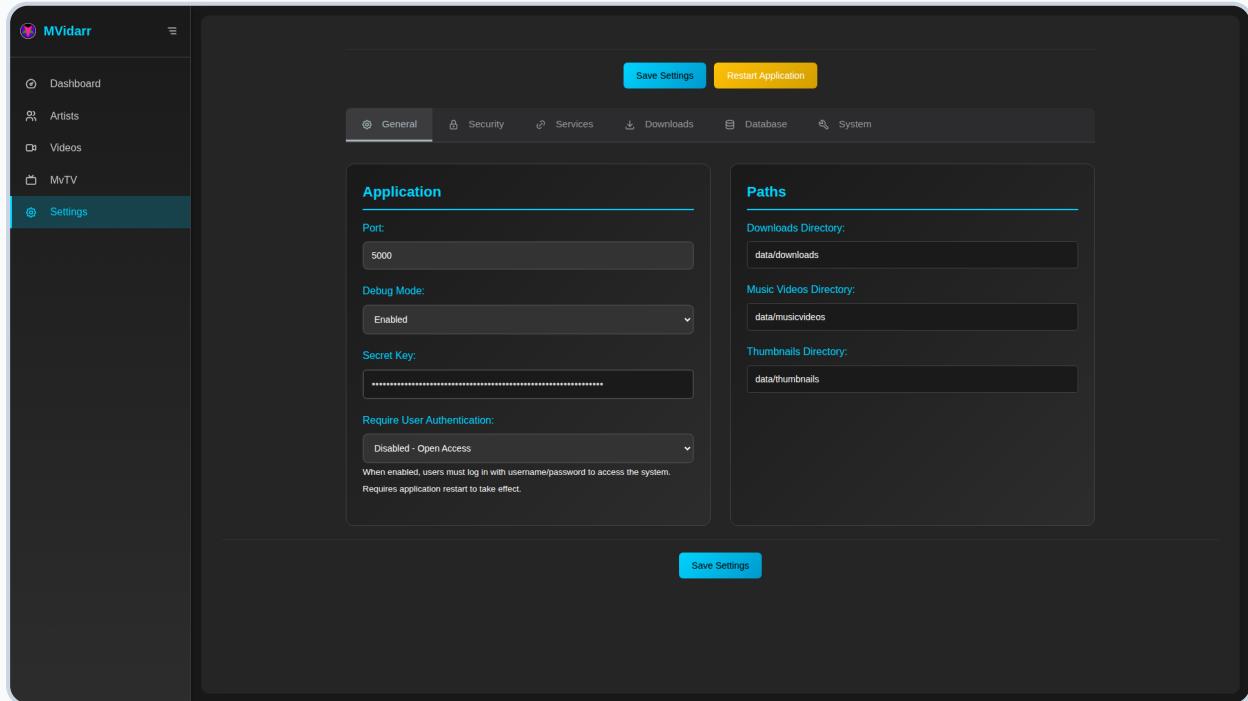
2. Download Configuration



Download Settings

- **Quality Preferences** - Video resolution and format selection
- **Storage Paths** - Where videos are stored and organized
- **Naming Templates** - Customize file naming conventions

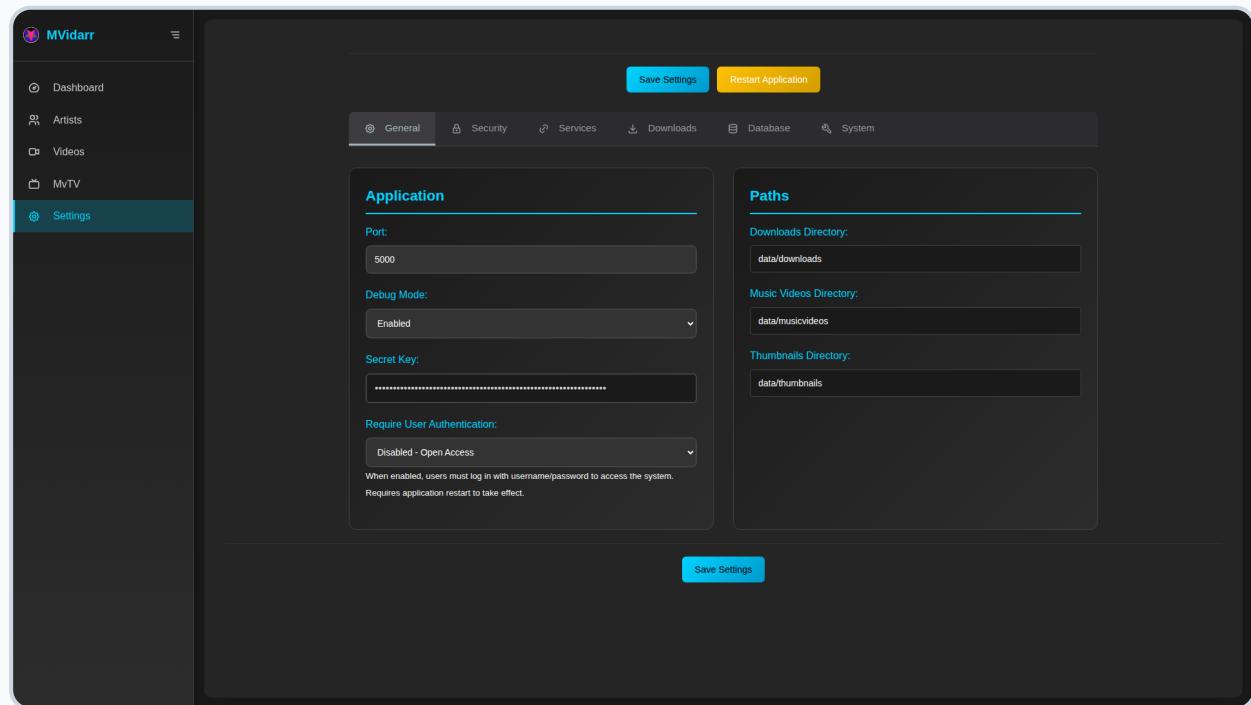
3. User Interface



UI Settings

- **Theme Selection** - Choose from 10+ themes
- **Page Sizes** - Customize pagination
- **Animation Preferences** - UI responsiveness settings

4. System Settings

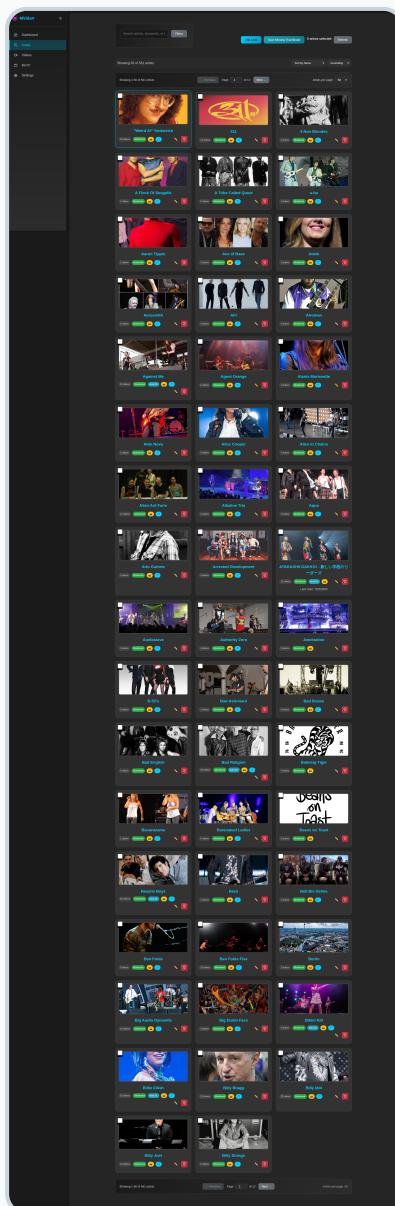


System Settings

- **Database Configuration** - Connection and performance settings
- **Logging Levels** - Debugging and monitoring configuration
- **Backup Settings** - Automated backup configuration

Interface Showcase

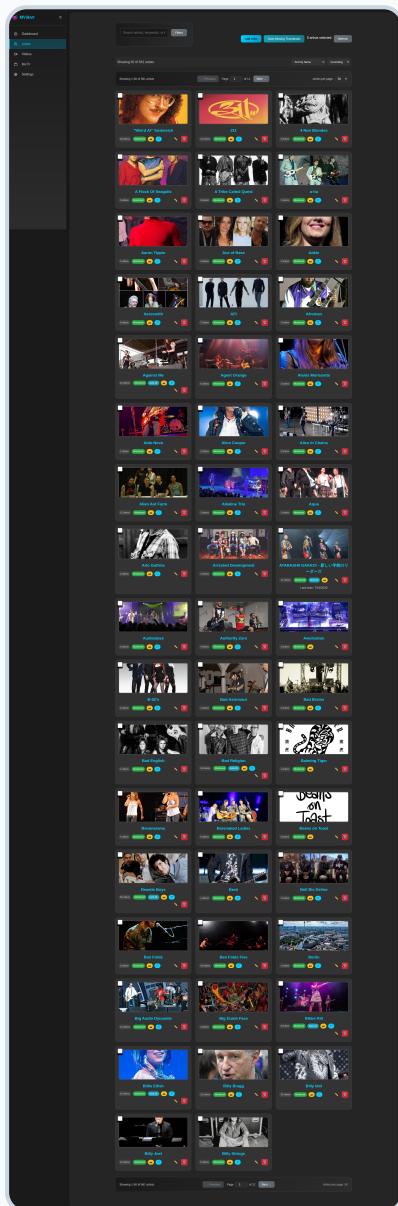
Dashboard Overview



Dashboard

Clean, informative dashboard with at-a-glance statistics

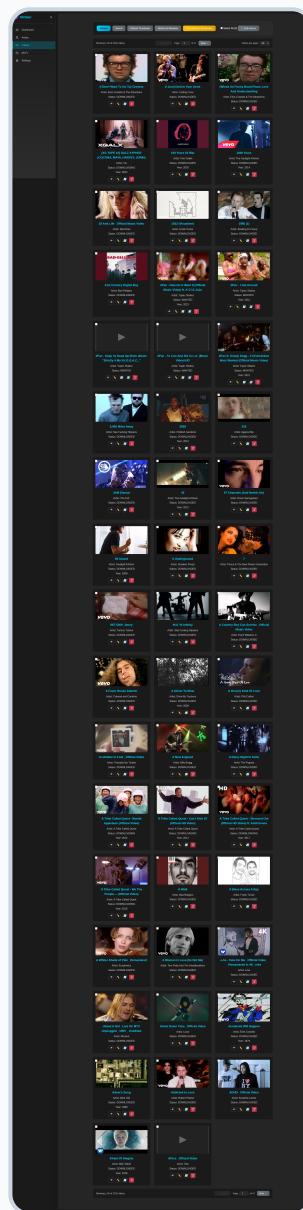
Artist Management



Artist List

Professional artist management with search, filtering, and bulk operations

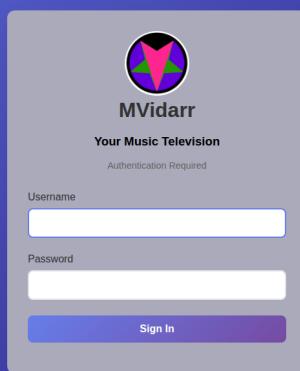
Video Library



Video Library

Comprehensive video library with advanced filtering and metadata

Download Manager



Download Manager

Real-time download monitoring with progress tracking and queue management

Mobile Interface



Mobile Interface

Fully responsive design optimized for mobile devices

Troubleshooting Guide

Common Issues

Database Connection Failed

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Database Error

Symptoms: Application won't start, database connection errors

Solutions:

1. Check MariaDB Status

```
sudo systemctl status mariadb  
sudo systemctl start mariadb # if not running
```

1. Verify Credentials

```
mysql -u mvidarr -p mvidarr_enhanced # test connection
```

2. Check Configuration

- Verify `DB_PASSWORD` in docker-config.yml or .env
- Ensure database name matches (`mvidarr_enhanced`)

3. Docker Specific

```
docker-compose logs mariadb # check database logs
```

🚫 Video Downloads Not Working

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Download Error

Symptoms: Downloads stuck, failed downloads, permission errors

Solutions:

1. Check yt-dlp Installation

```
docker exec mvidarr-enhanced yt-dlp --version  
# or locally: yt-dlp --version
```

1. Verify Directory Permissions

```
ls -la /path/to/your/musicvideos  
sudo chown -R 1001:1001 /path/to/your/musicvideos # for Docker
```

2. Check Logs

```
# Docker  
docker-compose logs mvidarr  
  
# Local  
tail -f data/logs/mvidarr.log
```

🚫 API Connection Issues

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

API Error

Symptoms: No video discovery, API key errors, rate limiting

Solutions:

1. Verify API Keys

- IMVDb: <https://imvdb.com/developers/api>
- YouTube: <https://console.developers.google.com>

1. Test API Connectivity

- Go to Settings → System Health
- Check API status indicators

2. Rate Limiting

- APIs have built-in rate limiting
- Wait 1-5 minutes between large operations

Performance Optimization

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Performance Monitor

Built-in performance monitoring and optimization recommendations

For Large Collections (1000+ videos):

1. Increase Database Buffer Size

```
SET GLOBAL innodb_buffer_pool_size = 1073741824; # 1GB
```

1. Enable Caching

- Thumbnail caching (enabled by default)
- Search result caching
- Metadata caching

2. Docker Resource Limits

```
services:  
  mvidarr:  
    deploy:  
      resources:  
        limits:  
          memory: 2G  
          cpus: '1.0'
```



System Monitoring

Health Dashboard

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Health Dashboard

Comprehensive system health monitoring with real-time status

Monitor your MVidarr Enhanced system:

-  **Component Status** - Database, API connections, storage
-  **Performance Metrics** - Response times, memory usage, disk space
-  **Alerts** - Automatic notifications for issues
-  **Analytics** - Usage statistics and trends

Log Management

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Log Viewer

Built-in log viewer with filtering and search capabilities

Access logs through:

- **Web Interface:** Settings → System Logs
- **Docker:** `docker-compose logs -f mvidarr`
- **Local:** `tail -f data/logs/mvidarr.log`

API Documentation

RESTful API

MVidarr Enhanced provides a complete REST API for integration:

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

API Documentation

Interactive API documentation with live testing capabilities

Base URL: `http://localhost:5000/api`

Key Endpoints:

- `GET /artists` - List all artists with filtering
- `POST /artists/{id}/discover` - Discover new videos for artist
- `GET /videos/search` - Advanced video search
- `POST /videos/{id}/download` - Queue video for download
- `GET /health` - System health status

Authentication: Optional API key authentication available



Production Deployment

Docker Swarm/Kubernetes

For production deployments at scale:

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Production Architecture

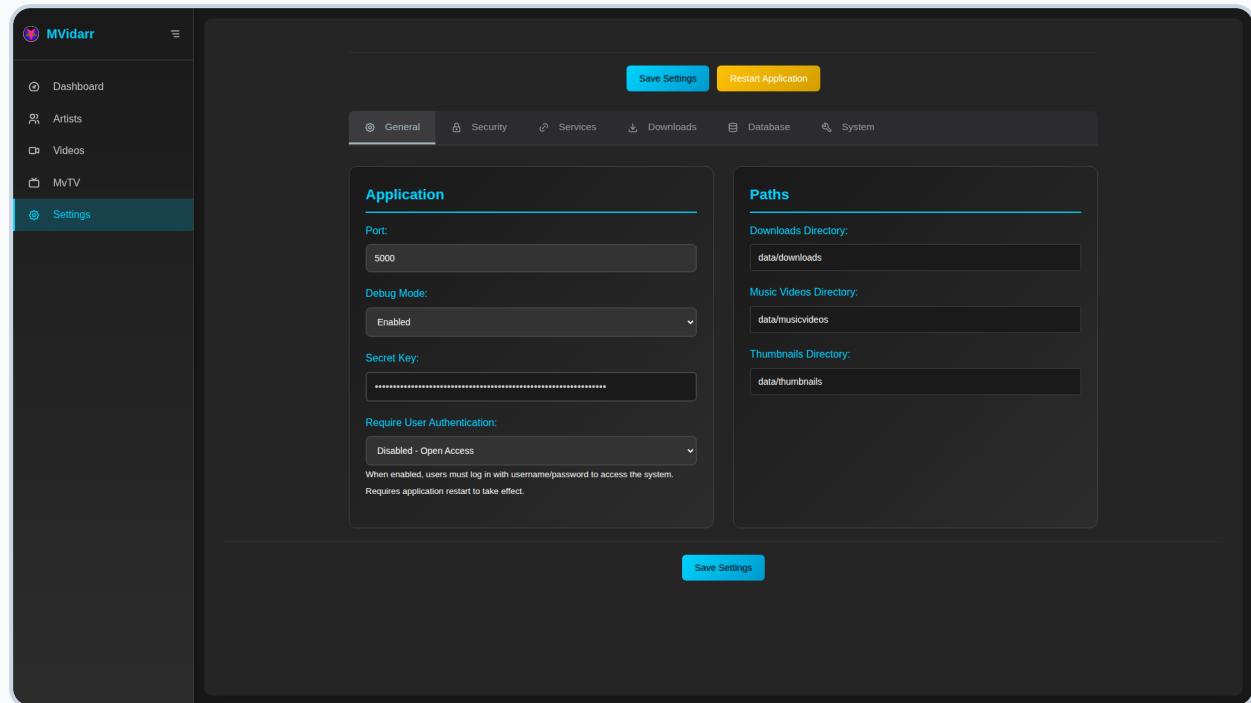
Scalable production architecture with load balancing and high availability

Docker Swarm Example:

```

version: '3.8'
services:
  mvidarr:
    image: mvidarr-enhanced:latest
    deploy:
      replicas: 2
      resources:
        limits:
          memory: 2G
    volumes:
      - nfs-storage:/app/data/downloads
  
```

Reverse Proxy Setup



Nginx Configuration

Sample Nginx configuration for SSL termination and load balancing

Nginx Configuration:

```
upstream mvidarr {
    server localhost:5000;
}

server {
    listen 80;
    server_name your-domain.com;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name your-domain.com;

    ssl_certificate /path/to/cert.pem;
    ssl_certificate_key /path/to/key.pem;

    location / {
        proxy_pass http://mvidarr;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```



Security Best Practices

Security Dashboard

Security configuration and monitoring interface

Essential Security Measures

1. Strong Authentication

- Change default passwords
- Use long, random SECRET_KEY
- Enable two-factor authentication (if configured)

2. Network Security

- Use reverse proxy for SSL termination
- Restrict database access to localhost
- Configure firewall rules

3. Regular Updates

- Keep Docker images updated
- Monitor security advisories
- Regular backup validation

4. Access Control

- Limit administrative access
- Monitor access logs
- Use strong API keys

Support & Community

Getting Help

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Support Resources

Comprehensive support resources and community links

-  **Documentation:** Complete guides and API reference
-  **GitHub Issues:** Bug reports and feature requests
-  **Built-in Diagnostics:** Settings → System Health
-  **Log Analysis:** Detailed logging with error tracking

Contributing

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Contributing Guide

Guidelines for contributing to MVidarr Enhanced development

1. **Fork the Repository**
2. **Create Feature Branch:** `git checkout -b feature/amazing-feature`
3. **Make Changes:** Follow code style guidelines
4. **Add Tests:** Ensure comprehensive test coverage
5. **Submit PR:** Detailed description of changes

🎯 What's Next?

Upcoming Features

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Roadmap

Development roadmap and planned features

-  **AI-Powered Recommendations** - Smart video discovery
-  **Mobile App** - Native iOS/Android applications
-  **Multi-Language Support** - Internationalization
-  **Audio Extraction** - Extract audio tracks from videos
-  **Advanced Analytics** - Detailed collection statistics



License & Acknowledgments

License: MIT License - see [LICENSE.md](#)

Acknowledgments:

- Inspired by [Lidarr](#) - Excellent music management
- [IMVDb](#) - Comprehensive music video database
- [yt-dlp](#) - Reliable video downloading
- Community contributors and testers

🎉 Ready to Get Started? [

 DEPLOY WITH DOCKER

Deploy with Docker

](docs/DOCKER-QUICKSTART.md) [

 VIEW DOCUMENTATION

View Documentation

](docs/) [

 JOIN COMMUNITY

Join Community

](<https://github.com/your-repo/discussions>) **MVidarr Enhanced** - Professional music video management made simple.

Last updated: July 22, 2025