Getting Started

Setup steps and first-run workflow

PSScript Manager Docs

PSScript - Getting Started Guide

This document provides a comprehensive guide to setting up and using the PSScript application.

System Overview

PSScript is a PowerShell script management system that allows users to: - Upload and manage PowerShell scripts - Organize scripts by categories and tags - Search scripts using vector search capabilities - Analyze scripts for security and best practices - Chat with an AI assistant about PowerShell topics

Prerequisites

- Node.js (v14+)
- PostgreSQL (v12+)
- Redis (optional, for enhanced caching)

Quick Start

1. Clone the repository

2. Install dependencies: `bash npm install`

3. Set up the environment variables: ```bash # Copy the example environment file cp .env.example .env

Edit the .env file with your configuration ```

1. Initialize the database: `bash node setup-database.js`

2. Start the application: ```bash # Start all services npm run start:all

Or start components individually npm run start:backend npm run start:frontend npm run start:ai ```

1. Access the application:
2. Frontend: http://localhost:3000
3. Backend API: http://localhost:4001
4. AI Service: http://localhost:5001

Default Admin Account

The system comes with a default administrator account: - Email:
`admin@psscript.com` - Password: `ChangeMe1!`

It's recommended to change the default password after first login.

Database Setup

PostgreSQL Configuration

1. Create a new database: `sql CREATE DATABASE psscript;`

2. Enable the pgvector extension (if using vector search): `sql \c psscript CREATE EXTENSION IF NOT EXISTS vector;`

3. Run the database initialization script: `bash node setup-database.js`

Redis Configuration (Optional)

Configure Redis in the .env file:

```
REDIS_URL=redis://localhost:6379
```

The system will automatically fall back to in-memory caching if Redis is unavailable.

Key Features

1. Script Management

- **Upload Scripts**: Upload PowerShell scripts through the web interface
- **Script Categories**: Organize scripts by categories
- **Tagging**: Apply tags to scripts for better organization
- **Version Control**: Track changes to scripts over time
- **Bulk Operations**: Perform actions on multiple scripts at once

2. Script Analysis

- **Security Analysis**: Automatically analyze scripts for security issues
- **Best Practices**: Check scripts against PowerShell best practices
- **Syntax Validation**: Verify script syntax is correct
- **Command Details**: View detailed information about PowerShell commands used

3. AI Integration

- **Chat Interface**: Ask questions about PowerShell
- **Script Generation**: Generate scripts based on natural language descriptions
- **Script Explanation**: Get plain-language explanations of script functionality
- **Documentation Search**: Search PowerShell documentation

4. Search Capabilities

- **Full-Text Search**: Search script content and metadata
- **Vector Search**: Find similar scripts using semantic search (requires pgvector)
- **Advanced Filters**: Filter by category, tags, author, date, etc.
- **Sorting Options**: Sort results by relevance, date, popularity, etc.

Troubleshooting

Database Connectivity Issues

If you experience database connection problems:

1. Check your database credentials in the .env file
2. Verify PostgreSQL is running: `bash pg_isready -h localhost -p 5432`
3. Run the database test script: `bash node src/backend/test-db.js`

Login Issues

If you're having trouble logging in:

1. Verify the database is properly set up
2. Reset the admin password: `bash node add-admin-user.js`
3. Check the auth logs: `bash tail -f logs/authentication.log`

Performance Issues

If the application is running slowly:

1. Ensure Redis is configured correctly for caching
2. Check database indexes are created properly
3. Verify pgvector extension is installed if using vector search

Additional Resources

- **Database Documentation**: See DATABASE-IMPROVEMENTS.md
- **Authentication Documentation**: See AUTHENTICATION-IMPROVEMENTS.md
- **Vector Search**: See README-VECTOR-SEARCH.md
- **Voice API Integration**: See README-VOICE-API.md

Support

For support, please open an issue in the GitHub repository or contact the development team.