

Database Improvements

This document summarizes the improvements made to the database connectivity and infrastructure of the PSScript application.

Improvements Made

1. Database Connection Enhancements

- Implemented robust database connection handler with proper error typing and exponential backoff
- Added intelligent error categorization (authentication, network, timeout, resource, etc.)
- Improved connection pooling configuration with optimized settings
- Added support for both individual parameters and DATABASE_URL connection string

2. Migration Management

- Created a robust migration tracking system with `schema_migrations` table
- Added transaction support for safer migrations
- Fixed migration ordering and dependency issues
- Implemented comprehensive error handling for migration failures

3. pgvector Integration

- Added proper pgvector extension installation and verification
- Created vector indexes for both script embeddings and chat history
- Added error handling for systems without pgvector

4. Redis and Caching

- Implemented a fault-tolerant caching mechanism with Redis
- Added graceful fallback to in-memory cache when Redis is unavailable
- Implemented health monitoring and automatic fallback/recovery
- Enhanced cache interface with type-safe operations

5. Testing and Diagnostics

- Improved database connection tests with more comprehensive diagnostics
- Enhanced error handling and reporting for both PostgreSQL and Redis
- Added detailed logging for connection issues and migrations

- Created combined setup and test script for easier verification

Setup Instructions

To set up the database from scratch:

```
# Initialize database schema and tables  
node setup-database.js  
  
# Run migrations for any additional schema changes  
node run-migration.js  
  
# Check pgvector extension status  
node check-pgvector.js  
  
# Full setup and testing (recommended)  
node setup-and-test.js
```

Configuration

Database configuration can be specified in two ways:

1. **Individual Parameters:** DB_HOST=localhost DB_PORT=5432

```
DB_NAME=psscript DB_USER=postgres DB_PASSWORD=postgres  
DB_SSL=false
```

2. **Connection String:**

```
DATABASE_URL=postgresql://postgres:postgres@localhost:5432/psscript
```

Redis Configuration

Redis can be configured with the following parameters:

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

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

Testing

To test the database and caching system:

```
# Test PostgreSQL connectivity  
cd src/backend && node test-db.js  
  
# Test Redis/caching  
cd src/backend && node test-redis.js
```

Additional Notes

- The application will automatically create the database schema if it doesn't exist
- Migrations are now tracked in the database to prevent duplicate runs
- Vector search capabilities are available when pgvector is installed
- The caching system will automatically use the best available option (Redis or in-memory)

Generated 2026-01-16 21:23 UTC