

# Vector Search and File Integrity

---

This document describes the implementation of vector search and file integrity verification features in the PSScript platform.

## Overview

---

We've added two key features to enhance the platform's capabilities:

1. **File Integrity Verification:** Ensures script files are not duplicated and maintains data integrity through MD5 hash verification.
2. **Vector Search:** Enables semantic similarity search for PowerShell scripts using vector embeddings.

## File Integrity Features

---

### Key Components

- **Hash Calculation:** MD5 hash generation for file content to uniquely identify scripts.
- **Duplicate Detection:** Prevents uploading identical scripts by comparing file hashes.
- **Integrity Verification:** Ensures scripts haven't been tampered with by comparing stored and calculated hashes.

### Implementation Files

- `src/backend/src/utils/fileIntegrity.ts` : Utility functions for file hash calculation and verification.
- `src/db/migrations/add_file_hash_to_scripts.sql` : Database migration to add file hash column.
- `run-file-hash-migration.sh` : Script to run the migration and update existing scripts.

# Vector Search Features

---

## Key Components

- **Embedding Generation:** Creates vector embeddings for script content using OpenAI's embedding models.
- **Similarity Search:** Finds semantically similar scripts using cosine similarity.
- **Hybrid Search:** Combines vector similarity with keyword search for better results.

## Implementation Files

- `src/backend/src/utils/vectorUtils.ts`: Utility functions for vector operations and search.
- `src/backend/src/controllers/ScriptController.ts`: Updated controller with vector search integration.

## Database Changes

---

The following changes were made to the database schema:

```
-- Add file_hash column to scripts table
ALTER TABLE scripts ADD COLUMN IF NOT EXISTS file_hash VARCHAR(255);

-- Add index on file_hash for faster duplicate detection
CREATE INDEX IF NOT EXISTS idx_scripts_file_hash ON scripts(file_hash);

-- Add embedding column for vector search
ALTER TABLE scripts ADD COLUMN IF NOT EXISTS embedding vector(1536);

-- Create vector index for faster similarity search
CREATE INDEX IF NOT EXISTS idx_scripts_embedding ON scripts USING GIN(embedding);
```

## Usage Examples

---

## File Integrity

```
// Calculate hash for a file
const fileHash = calculateBufferMD5(fileBuffer);

// Check if a file with the same hash exists
const existingScriptId = await checkFileExists(fileHash, sequelize)
if (existingScriptId) {
  // Handle duplicate file
}

// Verify file integrity
const isValid = verifyFileIntegrity(content, expectedHash);
```

## Vector Search

```
// Find similar scripts
const similarScripts = await findSimilarScripts(scriptId, 5, 0.7);

// Perform hybrid search with keywords and vectors
const searchResults = await hybridSearch(query, 10, 0.7, {
  categoryId: 5,
  isPublic: true
});
```

## Testing

---

Use the `test-file-hash-vector.sh` script to test the implementation:

```
./test-file-hash-vector.sh
```

This script tests: 1. File hash migration 2. Hash calculation 3. Vector search functionality

## Performance Considerations

---

- Vector search operations can be computationally intensive. The implementation uses PostgreSQL's vector operators with indexes to optimize performance.
- File hash calculation is performed once during upload and stored for future reference.
- Batch operations are provided for updating file hashes and embeddings for existing scripts.

## Future Improvements

---

- Implement periodic integrity checks to ensure all scripts maintain their integrity.
- Add more sophisticated vector search algorithms like Hierarchical Navigable Small World (HNSW).
- Explore using different embedding models for different types of PowerShell scripts.
- Implement caching for frequently accessed vector search results.