File Hash Deduplication for PowerShell Scripts

This document explains the implementation of file hash deduplication for PowerShell scripts in the PSScript application.

Overview

The file hash deduplication feature prevents duplicate scripts from being uploaded to the database. When a user attempts to upload a script, the system calculates an MD5 hash of the file content and checks if a script with the same hash already exists in the database. If a match is found, the upload is rejected with a 409 Conflict response, and the user is informed that a script with identical content already exists.

Implementation Details

The file hash deduplication feature is implemented in the following files:

1. `src/backend/src/models/Script.ts`: Added a `fileHash` field to the Script model to store the MD5 hash of the script content.
2. `src/backend/src/utils/fileIntegrity.ts`: Contains utility functions for calculating and verifying file hashes.
3. `src/backend/src/controllers/ScriptController.ts`: Uses the utility functions to calculate the hash of uploaded files and check for duplicates.

Script Model

The Script model was updated to include a `fileHash` field:

```
public fileHash?: string;

// In the model initialization
fileHash: {
  type: DataTypes.STRING(32),
  allowNull: true,
  field: 'file_hash'
}
```

File Integrity Utilities

The `fileIntegrity.ts` file provides the following functions:

- `calculateBufferMD5`: Calculates the MD5 hash of a file buffer.
- `calculateStringMD5`: Calculates the MD5 hash of a string.
- `checkFileExists`: Checks if a file with the same hash already exists in the database.
- `verifyFileIntegrity`: Verifies the integrity of a file by comparing its hash.
- `updateFileHash`: Updates the file hash in the database.
- `batchUpdateFileHashes`: Batch updates file hashes for scripts without hashes.

Script Controller

The ScriptController uses these utilities to implement the deduplication logic:

1. When a script is uploaded, the controller calculates the MD5 hash of the file content.
2. It then checks if a script with the same hash already exists in the database.
3. If a match is found, the upload is rejected with a 409 Conflict response.
4. If no match is found, the script is saved to the database with its hash.

Testing

The file hash deduplication feature can be tested using the following scripts:

- `test-upload-script.sh`: Uploads the test-script.ps1 file to the server.
- `test-upload-new-script.sh`: Uploads the test-script-new.ps1 file to the server.

When attempting to upload the same script multiple times, the server should return a 409 Conflict response with a message indicating that a script with identical content already exists.

Benefits

The file hash deduplication feature provides the following benefits:

1. **Storage Efficiency**: Prevents duplicate scripts from consuming storage space.
2. **Data Integrity**: Ensures that each script in the database is unique.
3. **User Experience**: Informs users when they attempt to upload a script that already exists.
4. **Search Optimization**: Makes it easier to find scripts by eliminating duplicates from search results.

Future Enhancements

Possible future enhancements to the file hash deduplication feature include:

1. **Similarity Detection**: Implement fuzzy matching to detect scripts that are similar but not identical.
2. **Version Control**: Allow users to update existing scripts instead of creating duplicates.
3. **Duplicate Management**: Provide tools for administrators to manage and merge duplicate scripts.
4. **Hash Algorithms**: Support additional hash algorithms for improved security and performance.

Generated 2026-01-16 23:34 UTC