DBexecute

Here's a detailed documentation for the provided Streamlit code:

Overview

This Python script uses Streamlit to create a web application for MySQL database operations, specifically backup and upgrade. Users can navigate between two tabs: "Backup" and "Upgrade." The script handles user inputs, performs MySQL backups, and executes SQL files for database upgrades.

Dependencies

- `streamlit`: For creating the web application interface.
- `os`: For file and directory operations.
- `subprocess`: To execute shell commands (e.g., `mysqldump` and `mysql`).
- `datetime`: For generating timestamps.

Streamlit Interface

- **Title**: 'DB Executor'
- **Sidebar**: Navigation with two options: 'Backup' and 'Upgrade'.

Sections

Backup

User Inputs:

- **MySQL Server IP**: `backup_server_ip` (string)
- **MySQL Server Port**: `backup_server_port` (string, default '3306')
- **Username**: `backup_username` (string)
- **Password**: `backup_password` (password field)
- **Database**: `backup_database` (string)

- **Backup Folder Path**: `backup_main_file` (string)

Functionality:

- **Button**: 'Backup' initiates the backup process.
- **Validation**: Checks if all required fields are filled.
- **Backup File Creation**: Uses `mysqldump` to create a backup file with a timestamp.
- **Directory Handling**: Ensures the backup directory exists.
- **Error Handling**: Logs errors if the backup process fails.

Backup Process:

- **Command**: `mysqldump -h {backup_server_ip} -P {backup_server_port} -u {backup_username} -p{backup_password} {backup_database} > "{backup_file_path}"`
- **Exception Handling**: Catches and prints errors during the backup.

Upgrade

User Inputs:

- **MySQL Server IP**: `upgrade_server_ip` (string)
- **MySQL Server Port**: `upgrade_server_port` (string, default '3306')
- **Username**: `upgrade_username` (string)
- **Password**: `upgrade_password` (password field)
- **Database**: `upgrade_database` (string)
- **Folder Path**: `upgrade_main_file` (string)

Functionality:

- **Button**: 'Upgrade' initiates the SQL file execution.
- **Validation**: Checks if all required fields are filled.
- **File Execution**: Executes SQL files from the specified folder.
- **Error Handling**: Logs errors if file execution fails or no SQL files are found.

Upgrade Process:

- **Command**: `mysql -h {upgrade_server_ip} -P {upgrade_server_port} -u {upgrade_username} -p{upgrade_password} {upgrade_database} < "{file_path}"`
- **File Handling**: Iterates over `.sql` files in the specified folder and executes them.
- **Success/Error Notifications**: Displays success or error messages based on file execution status.

Code Details

- **Backup Section**:
  - Uses `mysqldump` to back up the specified MySQL database.
  - Ensures the backup directory exists before executing the backup command.
- **Upgrade Section**:
  - Uses `mysql` to execute SQL files for upgrading the database.
  - Searches for `.sql` files in the specified folder and executes each.

Execution Flow

1. **User Selection**: Users select either 'Backup' or 'Upgrade' from the sidebar.
2. **Input Validation**: Checks if all required fields are filled in each section.
3. **Execution**: Depending on the section, either backs up the database or executes SQL files.
4. **Feedback**: Provides feedback to the user through success or error messages.
