Script Management System Documentation

Overview

This document describes a Flask-based application that allows users to upload and manage scripts associated with their datasets. Users can easily replace existing scripts by removing them first and then uploading new versions.

Features

- Upload scripts in various formats.
- Remove existing scripts before uploading new ones.
- Error handling for missing files and unsupported operations.
- Organizes scripts in a designated directory.

Installation

1. Ensure you have Python installed on your machine.
2. Install Flask using pip:

```
pip install Flask
```

Application Code

Below is the complete code for the script management application:

```
from flask import Flask, request, jsonify
import os

app = Flask(__name__)

# Set the script upload folder
SCRIPT_UPLOAD_FOLDER = 'scripts/'
os.makedirs(SCRIPT_UPLOAD_FOLDER, exist_ok=True)

@app.route('/upload_script', methods=['POST'])
def upload_script():
  if 'script' not in request.files:
      return jsonify({'error': 'No script part'}), 400

  script = request.files['script']

  if script.filename == '':
      return jsonify({'error': 'No selected script'}), 400
```

```
  # Save the script
  script_path = os.path.join(SCRIPT_UPLOAD_FOLDER, script.filename)
  script.save(script_path)
  return jsonify({'message': 'Script successfully uploaded', 'filename': script.filename}), 200

@app.route('/remove_script/', methods=['DELETE'])
def remove_script(filename):
  script_path = os.path.join(SCRIPT_UPLOAD_FOLDER, filename)
  if os.path.exists(script_path):
      os.remove(script_path)
      return jsonify({'message': 'Script successfully removed'}), 200
  return jsonify({'error': 'Script not found'}), 404

if __name__ == '__main__':
  app.run(debug=True)
```

How to Use the Script Management System

1. Save the script in a file, e.g., script_manager.py.
2. Run the script:

   ```
   python script_manager.py
   ```

3. Use a tool like Postman to send a POST request to http://localhost:5000/upload_script with the script file.
4. To remove a script, send a DELETE request to http://localhost:5000/remove_script/<filename>.

Example HTML Form for Uploading Scripts

Below is an example of a simple HTML form to test the upload functionality:

```
<!DOCTYPE html>
<html>
<head>
  <title>Upload Script</title>
</head>
<body>
  <h1>Upload Script</h1>
  <form action="http://localhost:5000/upload_script" method="post" enctype="multipart/form-data">
      <input type="file" name="script" required>
      <input type="submit" value="Upload">
  </form>
```

```
  <h2>Remove Script</h2>
  <form action="http://localhost:5000/remove_script" method="post">
     <input type="text" name="filename" placeholder="Enter script filename" required>
     <input type="submit" value="Remove">
  </form>
</body>
</html>
```

Conclusion

This script management system provides a straightforward way to handle scripts associated with datasets, allowing for easy uploads and replacements. For further customization or additional features, please refer to the Flask documentation or consult the community.