

Dataset Upload Script Documentation

Overview

This document describes a Python script that allows users to upload datasets in various formats. The script is built using the Flask web framework and includes validation for file types and size.

Features

- Supports multiple file formats including CSV, JSON, XML, and more.
- Restricts uploads to a maximum file size of 2 GB.
- Provides error handling for unsupported file types and empty uploads.
- Saves uploaded files in a designated directory.

Installation

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

```
pip install Flask
```

Script Code

Below is the complete code for the dataset upload script:

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

app = Flask(__name__)

# Set the upload folder
UPLOAD_FOLDER = 'uploads/'
ALLOWED_EXTENSIONS = {
    'zip', 'gz', 'gzip', 'csv', 'json', 'txt', 'sql', 'xml', 'tsv',
    'ebs', 'avro', 'orc', 'parquet', 'hdf5', '7z', 'tbz2', 'iso',
    'tar', 'bz2', 'z', 'xls', 'xlsx', 'graph', 'properties', 'offsets',
    'flac', 'ogg', 'wav', 'aac', 'mp3', 'gif', 'jpg', 'jpeg', 'png',
    'avi', 'mov', 'mp4', 'mpg', 'm4v', 'yaml', 'dat', 'mat', 'fig'
}

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['MAX_CONTENT_LENGTH'] = 2 * 1024 * 1024 * 1024 # 2 GB limit

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

```

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

    file = request.files['file']

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

    if file and allowed_file(file.filename):
        # Save the file
        os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
        file.save(file_path)
        return jsonify({'message': 'File successfully uploaded', 'filename': file.filename}), 200

    return jsonify({'error': 'File type not allowed'}), 400

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

```

How to Use the Script

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

```
python upload_service.py
```

3. Use a tool like Postman or a simple HTML form to send a POST request to `http://localhost:5000/upload` with the file.

Example HTML Form for Upload

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

```

<!DOCTYPE html>
<html>
<head>
  <title>Upload Dataset</title>
</head>
<body>
  <h1>Upload Dataset</h1>

```

```
<form action="http://localhost:5000/upload" method="post" enctype="multipart/form-data">
  <input type="file" name="file" required>
  <input type="submit" value="Upload">
</form>
</body>
</html>
```

Conclusion

This script provides a basic framework for uploading datasets in various formats while adhering to specified constraints. For further customization or additional features, please refer to the Flask documentation or consult the community.