

Implementare l'upload di file multipli con Node.js

Caricare file multipli è una funzionalità essenziale per molte applicazioni web, come piattaforme di condivisione di documenti, gallerie fotografiche o gestori di file. Node.js offre diversi modi per implementare questa funzionalità. In questo articolo, vedremo come implementare l'upload di file multipli utilizzando **Express.js** e il middleware **Multer**.

Prerequisiti

- Node.js installato sul tuo sistema.
- Un progetto Node.js configurato con Express.js.
- Conoscenze di base di JavaScript e Node.js.

Configurazione del progetto

```
mkdir file-upload-app
cd file-upload-app
npm init -y
npm install express multer
```

Crea una struttura base del progetto:

```
file-upload-app/
|-- uploads/
|-- app.js
```

Implementazione del codice

1. Importare le dipendenze

```
const express = require('express');
const multer = require('multer');
const path = require('path');

const app = express();
const PORT = 3000;
```

2. Configurare Multer

```
const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, 'uploads/');
  },
  filename: (req, file, cb) => {
    const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);
    cb(null,
`${file.fieldname}-${uniqueSuffix}${path.extname(file.originalname)}`);
  }
});

const upload = multer({ storage: storage });
```

3. Creare il percorso per l'upload di file multipli

```
app.post('/upload', upload.array('files', 10), (req, res) => {
  try {
    const files = req.files;
    if (!files || files.length === 0) {
      return res.status(400).send('Nessun file caricato.');
```

```
    }

    res.status(200).send({
      message: 'File caricati con successo!',
      files: files.map(file => ({
        filename: file.filename,
        path: file.path
      }))
    });
  } catch (error) {
    res.status(500).send({ message: 'Errore durante il caricamento dei file.' });
  }
});
```

4. Avviare il server

```
app.listen(PORT, () => {
  console.log(`Server in esecuzione su http://localhost:${PORT}`);
});
```

Test dell'applicazione

Per testare l'applicazione, puoi utilizzare strumenti come **Postman** o **cURL**.

Esempio di comando cURL:

```
curl -X POST -F "files=@file1.jpg" -F "files=@file2.jpg"
http://localhost:3000/upload
```

Applicazioni Correlate



-

Node.js Placeholder Image

Applicazione per la generazione con Node.js di immagini segnaposto.
DockerDocker ComposeNode.jsJavaScriptExpressJS



-

Node.js URL Shortener

Implementazione in Node.js di un sistema per l'abbreviazione degli URL.

DockerDocker ComposeNode.jsJavaScriptExpressJSMongoDB

A large, bold, black 'JS' logo is centered on a solid yellow rectangular background.

-

JavaScript App Hash Change

Applicazione che sfrutta gli hash degli URL per gestire contenuto dinamico in JavaScript.

DockerDocker ComposeNode.jsJavaScriptExpressJSMYSQL