

GABRIELE ROMANATO

Come implementare l'upload di file multipli con PHP

L'upload di file multipli è una funzionalità comune nelle applicazioni web moderne. PHP offre strumenti semplici ed efficaci per gestire questa operazione. In questo articolo, vedremo come creare un modulo per caricare file multipli e gestirli sul server.

Creazione del Modulo HTML

Per consentire agli utenti di caricare file multipli, è necessario utilizzare il campo `<input>` con l'attributo `multiple`. Ecco un esempio di modulo HTML:

```
<form action="upload.php" method="post" enctype="multipart/form-data">
  <label for="files">Scegli i file da caricare:</label>
  <input type="file" name="files[]" id="files" multiple>
  <button type="submit">Carica</button>
</form>
```

Nota l'uso dell'attributo `name="files[]"`: l'uso delle parentesi quadre permette di inviare un array di file al server.

Gestione dell'Upload in PHP

Il codice PHP per gestire i file caricati controlla la variabile globale `$_FILES`. Ecco un esempio di script per gestire i file multipli:

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Creare una cartella di destinazione se non esiste
    $uploadDir = 'uploads/';
    if (!is_dir($uploadDir)) {
        mkdir($uploadDir, 0755, true);
    }

    // Iterare sui file caricati
    foreach ($_FILES['files']['tmp_name'] as $key => $tmpName) {
        if ($_FILES['files']['error'][$key] === UPLOAD_ERR_OK) {
            $fileName = basename($_FILES['files']['name'][$key]);
            $destination = $uploadDir . $fileName;

            // Spostare il file caricato nella cartella di destinazione
            if (move_uploaded_file($tmpName, $destination)) {
                echo "File $fileName caricato con successo.<br>";
            }
        }
    }
}
```

```
        } else {
            echo "Errore nel caricamento del file $fileName.<br>";
        }
    } else {
        echo "Errore nel file " . $_FILES['files'][$key] . ".
<br>";
    }
}
} else {
    echo "Nessun file caricato.";
}
?>
```

Validazione e Sicurezza

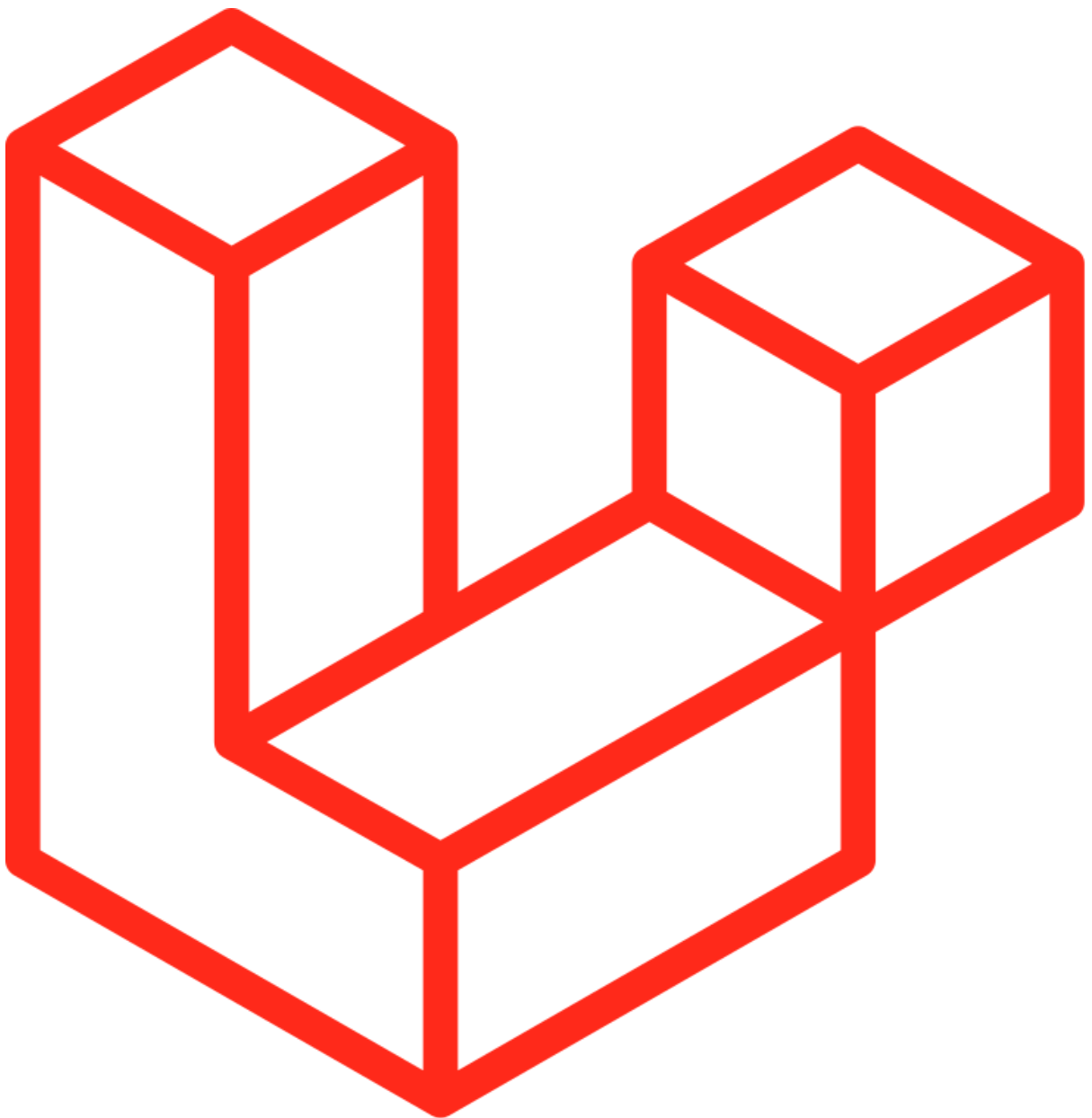
Durante l'upload, è importante effettuare delle verifiche per garantire la sicurezza. Ecco alcune best practice:

- Controllare il tipo MIME dei file con `mime_content_type()`.
- Limitare la dimensione dei file usando `$_FILES['files']['size']`.
- Sanitizzare i nomi dei file per evitare rischi di path traversal.
- Impostare permessi adeguati alla cartella di destinazione.

Conclusione

Implementare l'upload di file multipli con PHP è semplice e può essere fatto in pochi passaggi. Assicurati sempre di seguire le best practice per la sicurezza per evitare vulnerabilità nella tua applicazione.

Applicazioni Correlate

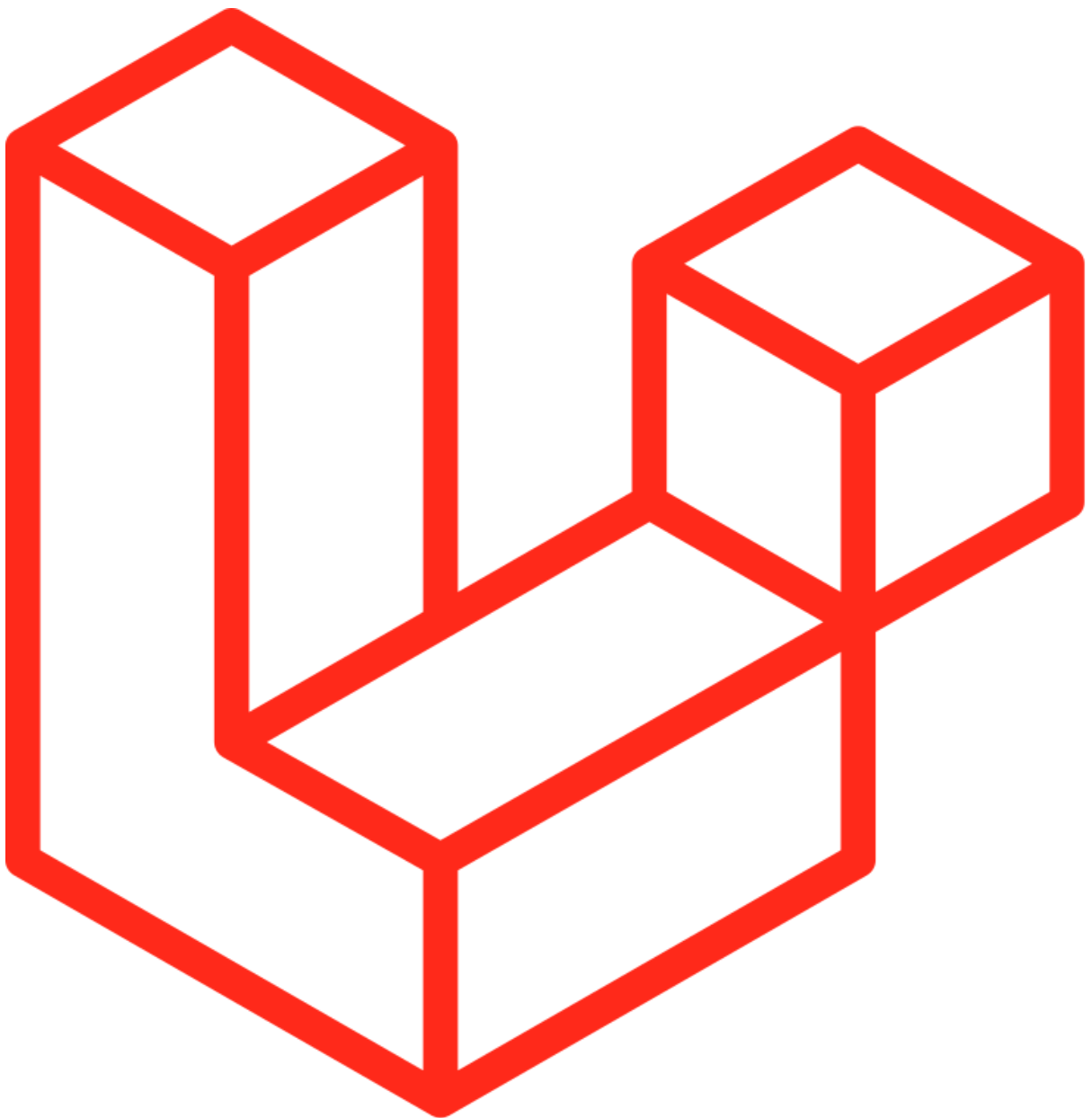


•

Laravel Placeholder Image

Applicazione in Laravel per creare immagini segnaposto.

Docker Docker Compose Composer PHP Laravel JavaScript

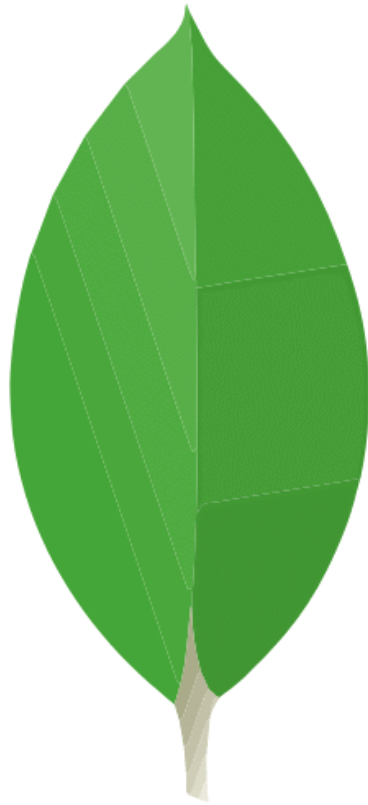


•

Laravel Shopping Cart

Applicazione che fa parte del progetto Laravel E-commerce e implementa la gestione del carrello in Laravel.

Docker Docker Compose Composer PHP Laravel



-

PHP MongoDB App

Applicazione basata su MongoDB con il driver PHP ufficiale.
DockerDocker ComposeComposerPHPMongoDB