

Laravel Mastering File Upload

Laravel File's Important Methods

```
<!-- Single File Input Field -->
<form action="{{ route('upload.file') }}" method="post" enctype="multipart/form-data">
    @csrf
    <label for="image">Choose a file:</label>
    <input type="file" name="image" id="image">
</form>

<!-- Multiple File Input Field -->
<form action="{{ route('upload.file') }}" method="post" enctype="multipart/form-data">
    @csrf
    <label for="image">Choose a file:</label>
    <input type="file" name="images[]" id="image">
</form>

//=====
// Getting Information
//=====
// Get the original name of the uploaded file.
$request->file('image')->getClientOriginalName();

// Get the original extension of the uploaded file.
$request->file('image')->getClientOriginalExtension();

// Get the size of the uploaded file in bytes.
$request->file('image')->getSize();

// Get the MIME type of the uploaded file.
$request->file('image')->getMimeType();

//=====
// Moving and Storing
//=====
// Store the uploaded file on disk (using Laravel's storage system).
$request->file('image')->store($path, $disk);

// Move the uploaded file to a new location.
$request->file('image')->move($directory, $name);

//=====
// Checking
//=====

// Check if the request has a file with the key 'image'
```

```

$request->hasFile('image'))

    // Check if the uploaded file is valid.
$request->file('image')->isValid();

    // Check if the uploaded file is a regular file.
$request->file('image')->isFile();

    // Check if the uploaded file is an image.
$request->file('image')->isImage();

    //=====
    // Manipulating
    //=====

    // Store the uploaded file on disk with a custom name.
$request->file('image')->storeAs($path, $name, $disk);

    // Get the original name of the file without the extension.
$request->file('image')->getClientOriginalNameWithoutExtension();

    // Get a unique name for the uploaded file.
$request->file('image')->hashName();

    //=====
    // Customizing
    //=====

    // Store the uploaded file publicly.
$request->file('image')->storePublicly($path, $disk);

    // Store the uploaded file publicly with a custom name.
$request->file('image')->storePubliclyAs($path, $name, $disk);

    //=====
    // Deleting
    //=====

    // Delete the uploaded file.
$request->file('image')->delete();

    //=====
    // Custom Validation
    //=====

    // Perform custom validation on the file.
$request->file('image')->isValid();

    // Custom validation with a MIME type constraint.
$request->file('image')->isValid('image/*');

    //=====
    // Stream Access

```

```
//=====

// Get the contents of the uploaded file as a string.
$request->file('image')->get();

// Get the real path to the uploaded file.
$request->file('image')->getRealPath();

//=====
// Client Disposition
//=====

// Get the client-provided name of the file.
$request->file('image')->getClientOriginalName();
```

Upload Single Image in Storage > App > Public

```
//Storage Single Image Upload
Route::get( '/store-single-upload', [ImageController::class, 'storage_single'] );
Route::post( '/store-single-upload', [ImageController::class, 'storage_single_store']
)->name( 'storage.single.store' );
```

```
// Storage Single Image Upload
public function storage_single(){
    return view('image-upload.storage-single-upload');
}

public function storage_single_store(Request $request){
    // Validation
    $this->validate( $request, [
        'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
    ],
    [
        //Custom Error Message
        'image.required' => 'Please Select Image',
    ]
    );

    //Upload the image
    //Check if the file exist

    if ( $request->hasFile( 'image' ) ) {
        //Get filename with the extension
        $image = $request->file( 'image' );
        //Filename to store
        $fileNameToStore = md5(uniqid()) . '-' . time() . '.' . $image->getClientOriginalExtension();
        //Upload Image
        $image->storeAs( 'public/images', $fileNameToStore );
```

```

}
return redirect()->back();
}

```

Upload Multiple Image in Storage > App > Public

```

//Storage Multiple Images Upload
Route::get( '/store-multiple-upload', [ImageController::class, 'storage_multiple'] );
Route::post( '/store-multiple-upload', [ImageController::class, 'storage_multiple_store'] )->name( 'storage.multiple.store' );

```

```

public function storage_multiple(){
    return view('image-upload.storage-multiple-upload');
}

public function storage_multiple_store(Request $request){
    // Validation
    $this->validate( $request, [
        'images' => 'required',
        'images.*' => 'image|mimes:jpeg,png,jpg,gif,svg|max:2048',
    ] );

    if ( $request->hasFile( 'images' ) ) {

        $images = $request->file( 'images' );

        foreach ( $images as $imagefile ) {
            //Filename to store
            $fileNameToStore = md5(uniqid()) . "-" . time(). "." . $imagefile->getClientOriginalExtension();
            //Upload Image
            $imagefile->storeAs( 'public/images', $fileNameToStore );
        }
    }
    return redirect()->back();
}

```

Upload Single Image in Public

```

//Public Single Image Upload
Route::get( '/pub-single-upload', [ImageController::class, 'pub_single'] );
Route::post( '/pub-single-upload', [ImageController::class, 'pub_single_store'] )->name( 'pub.single.store' );

```

```
// Public Single
public function pub_single(){
    return view('image-upload.public-single-upload');
}

public function pub_single_store(Request $request){

// Validation
$this->validate( $request, [
    'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
],
[
    //Custom Error Message
    'image.mimes' => 'Your uploaed image does not match the requirements',
]);

    if ( $request->hasFile( 'image' ) ) {
        $image = $request->file( 'image' );
        //Filename to store
        $fileNameToStore = md5(uniqid()) . '-' . time() . '.' . $image->getClientOriginal
Extension();
        //Upload Image
        $image->move(public_path('assets/uploads'), $fileNameToStore);
    }
    return redirect()->back();
}
```

Upload Multiple Images in Public

```
//Public Multiple Images Upload
Route::get( '/pub-multiple-upload', [ImageController::class, 'pub_multiple'] );
Route::post( '/pub-multiple-upload', [ImageController::class, 'pub_multiple_store'] )->name( 'pub.multiple.store' );
```

```
public function pub_multiple(){
    return view('image-upload.public-multiple-upload');
}

public function pub_multiple_store(Request $request){
    // Validation
    $this->validate( $request, [
        'images' => 'required',
        'images.*' => 'image|mimes:jpeg,png,jpg,gif,svg|max:2048',
    ] );

    if ( $request->hasFile( 'images' ) ) {
        $images = $request->file( 'images' );
        foreach ( $images as $imagefile ) {
```

```
        //Filename to store
        $fileNameToStore = md5(uniqid()) . "-" . time() . "." . $imagefile->getClientOriginalExtension();
        //Upload Image
        $imagefile->move(public_path('assets/uploads'), $fileNameToStore);

    }
}
return redirect()->back();
}
```