https://www.codesolutionstuff.com/laravel-9-dropzone-image-upload-example-step-by-step?expand_article=1

Laravel 9 Dropzone Image Upload Example Step By Step

14 Apr 2023 CodeSolutionStuff **Laravel**

The most well-known, free, and open-source library for drag-and-drop file uploads with image previews is Dropzone. I'll be using Laravel 9 in this example.

Laravel Dropzone Image Upload

- To begin, use dropzone to upload several photos.
- Adding photos to the database with alternative file names.
- Removing photos from the preview box of the dropzone.

Step 1: Download Laravel Project

Type the following command to create a Laravel project.

```
composer create-project --prefer-dist laravel/laravel
dropzonefileupload
```

Step 2: Set up a MySQL database

```
//.env

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=**********
DB_USERNAME=root
DB_PASSWORD=
```

Step 3: Compose a model and migration file

In your cmd, type the following command.

```
php artisan make:model ImageUpload -m
```

There will be two files created.

- ImageUpload.php model.
- create_image_uploads_table migration file.

For the image upload table, we'll need to develop a Schema. Go to `Laravel >> database >> migrations >> create_image_uploads_table` to get started.

```
//create_image_uploads_table

 public function up()
    {
        Schema::create('image_uploads', function (Blueprint $table) {
            $table->increments('id');
            $table->text('filename');
            $table->timestamps();
        });
    }
```

Step 4: Create a view file

Create an imageupload.blade.php file in `resources >> views >> imageupload.php`. Put the code below in there. We'll add a dropzone for file uploading in this file.

```
<!-- imageupload.blade.php -->
<!DOCTYPE html>
<html>
<head>
    <title>Laravel Multiple Images Upload Using Dropzone</title>
    <meta name="_token" content="{{csrf_token()}}" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/dropzone/5.4.0/min/dropzo
ne.min.css">
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></sc
ript>
```

```
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/dropzone/5.4.0/dropzone.js
"></script>
</head>
<body>
<div class="container">

    <h3 class="jumbotron">Laravel Multiple Images Upload Using
Dropzone</h3>
    <form method="post" action="{{url('image/upload/store')}}"
enctype="multipart/form-data"
                class="dropzone" id="dropzone">
    @csrf
</form>
</body>
</html>
```

First, we'll include our bootstrap.min.css and dropzone.min.css files in this file. After that, we'll include jquery.js and dropzone.js. After that, we'll make a form and add the dropzone class to it.

In addition, we have some text in our upload box. Also, if the image is successfully uploaded, it will display a tick otherwise it displays a cross and error.

Step 5: Configure Dropzone

Now we'll write all of the Dropzone setups. So, in a view file, add the following code.

```
<!-- imageupload.blade.php -->
<!DOCTYPE html>
<html>
<head>
    <title>Laravel Multiple Images Upload Using Dropzone</title>
    <meta name="_token" content="{{csrf_token()}}" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/dropzone/5.4.0/min/dropzo
ne.min.css">
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></sc
ript>
```

```
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/dropzone/5.4.0/dropzone.js
"></script>
</head>
<body>
<div class="container">

    <h3 class="jumbotron">Laravel Multiple Images Upload Using
Dropzone</h3>
    <form method="post" action="{{url('image/upload/store')}}"
enctype="multipart/form-data"
                class="dropzone" id="dropzone">
    @csrf
</form>
<script type="text/javascript">
        Dropzone.options.dropzone =
         {
            maxFilesize: 12,
            renameFile: function(file) {
                var dt = new Date();
                var time = dt.getTime();
               return time+file.name;
            },
            acceptedFiles: ".jpeg,.jpg,.png,.gif",
            addRemoveLinks: true,
            timeout: 5000,
            success: function(file, response)
            {
                console.log(response);
            },
            error: function(file, response)
            {
               return false;
            }
};
</script>
</body>
</html>
```

We're adding Dropzone setup options to the file above. Any of the setting options are documented in the Dropzone Documentation

Let's take a look at each choice one by one.

- **maxFilesize** is set to 12 by default. Dropzone will only accept photos that are smaller than 12MB in size. You can make it smaller or larger depending on your needs.
- Before the file is uploaded to the server, the **renameFile** function is called, which renames the file.
- **acceptedFiles** compares the mime type or extension of the file to this list. The terms.jpeg,.jpg,.png, and.gif are defined. You have the option to alter according on your requirements.
- The value of **addRemoveLinks** is set to true. Dropzone will show the Remove button, which we may use to delete the file we just uploaded.
- The **timeout** is set to 5000 seconds.

Step 6: Create one controller and route

```
php artisan make:controller ImageUploadController
```

`ImageUploadController.php` will be created, and we'll register routes in the `routes >> web.php` file. So let's get started.

```
//web.php

Route::get('image/upload','ImageUploadController@fileCreate');
Route::post('image/upload/store','ImageUploadController@fileStore');
Route::post('image/delete','ImageUploadController@fileDestroy');
```

The next step is to add some code to the **fileCreate()** function in the `ImageUploadController.php file`.

```
// ImageUploadController.php

public function fileCreate()
    {
        return view('imageupload');
    }
```

We are simply returning the imageupload that we have made in the `create()` method.

Step 7: Save File into Database

To store the filename in the database, we must code the `fileStore()` procedure in sequence.

```
// ImageUploadController.php

use App\ImageUpload;

public function fileStore(Request $request)
    {
        $image = $request->file('file');
        $imageName = $image->getClientOriginalName();
        $image->move(public_path('images'),$imageName);

        $imageUpload = new ImageUpload();
        $imageUpload->filename = $imageName;
        $imageUpload->save();
        return response()->json(['success'=>$imageName]);
    }
```

Step 8: Remove File From Database

The **removedFile()** function has now been added to the dropzone configuration.

```
<!-- imageupload.blade.php -->
<!DOCTYPE html>
<html>
<head>
    <title>Laravel Multiple Images Upload Using Dropzone</title>
    <meta name="_token" content="{{csrf_token()}}" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/dropzone/5.4.0/min/dropzo
ne.min.css">
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></sc
ript>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/dropzone/5.4.0/dropzone.js
"></script>
</head>
<body>
<div class="container">
```

```
    <h3 class="jumbotron">Laravel Multiple Images Upload Using
Dropzone</h3>
    <form method="post" action="{{url('image/upload/store')}}"
enctype="multipart/form-data"
                class="dropzone" id="dropzone">
    @csrf
</form>
<script type="text/javascript">
        Dropzone.options.dropzone =
         {
            maxFilesize: 12,
            renameFile: function(file) {
                var dt = new Date();
                var time = dt.getTime();
               return time+file.name;
            },
            acceptedFiles: ".jpeg,.jpg,.png,.gif",
            addRemoveLinks: true,
            timeout: 50000,
            removedfile: function(file)
            {
                var name = file.upload.filename;
                $.ajax({
                    headers: {
                                'X-CSRF-TOKEN':
$('meta[name="_token"]').attr('content')
                            },
                    type: 'POST',
                    url: '{{ url("image/delete") }}',
                    data: {filename: name},
                    success: function (data){
                        console.log("File has been successfully
removed!!");
                    },
                    error: function(e) {
                        console.log(e);
                    }});
                    var fileRef;
                    return (fileRef = file.previewElement) != null ?

fileRef.parentNode.removeChild(file.previewElement) : void 0;
            },

            success: function(file, response)
            {
```

```
                console.log(response);
            },
            error: function(file, response)
            {
               return false;
            }
};
</script>
</body>
</html>
```

To delete a file from the database, add the **fileDestroy()** function. In FileUploadController, add the following code.

```
//ImageUploadController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\ImageUpload;

class ImageUploadController extends Controller
{
    public function fileCreate()
    {
        return view('imageupload');
    }
    public function fileStore(Request $request)
    {
        $image = $request->file('file');
        $imageName = $image->getClientOriginalName();
        $image->move(public_path('images'),$imageName);

        $imageUpload = new ImageUpload();
        $imageUpload->filename = $imageName;
        $imageUpload->save();
        return response()->json(['success'=>$imageName]);
    }
    public function fileDestroy(Request $request)
    {
        $filename =  $request->get('filename');
        ImageUpload::where('filename',$filename)->delete();
        $path=public_path().'/images/'.$filename;
```

```
        if (file_exists($path)) {
            unlink($path);
        }
        return $filename;
    }
}
```

Our Laravel Dropzone Image Upload is now complete. Thank you for taking the time to do so.