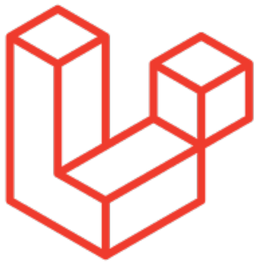
atec



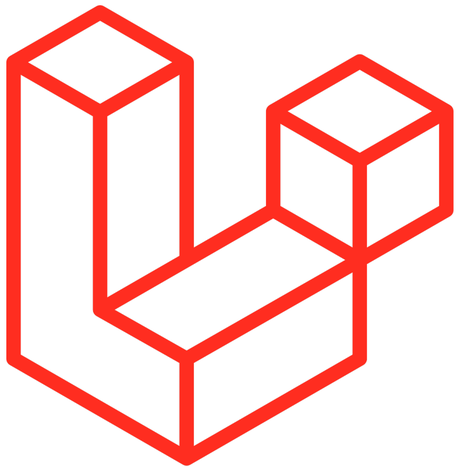
Laravel

Programação para a WEB - servidor (server-side)

Sérgio da Silva Nogueira

PHP

Files - Controller - Migration

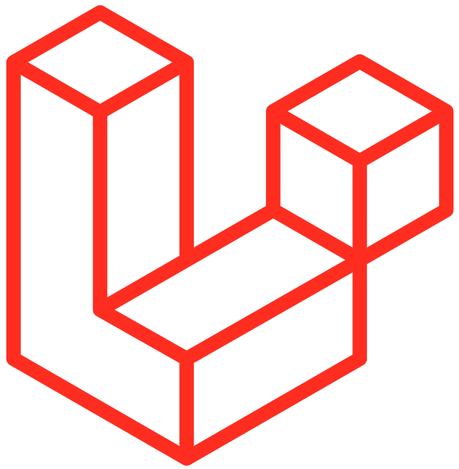


https://laravel.com/docs/7.x/filesystem#file-uploads

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePlayersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('players', function (Blueprint $table) {
            $table->id();
            $table->string('image')->nullable();
            $table->string('name');
            $table->string('address');
            $table->text('description')->nullable();
            $table->boolean('retired')->default(1);
            $table->timestamps();
            $table->softDeletes();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('players');
    }
}
```

PHP

Files - Create View

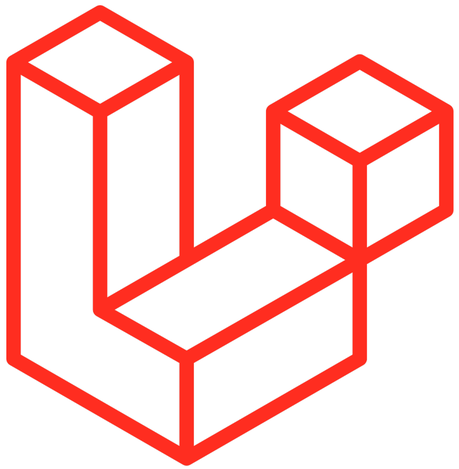


https://laravel.com/docs/7.x/filesystem#file-uploads

```
<form method="POST" action="{{ url('players') }}" enctype="multipart/form-data">
    @csrf

    <div class="form-group">
        <label for="name">Image</label>
        <input
            type="file"
            id="image"
            name="image"
            autocomplete="image"
            class="form-control
            @error('image') is-invalid @enderror"
            value="{{ old('image') }}"
            required>

        @error('image')
        <span class="invalid-feedback" role="alert">
                <strong>{{ $message }}</strong>
            </span>
        @enderror
    </div>
```

PHP

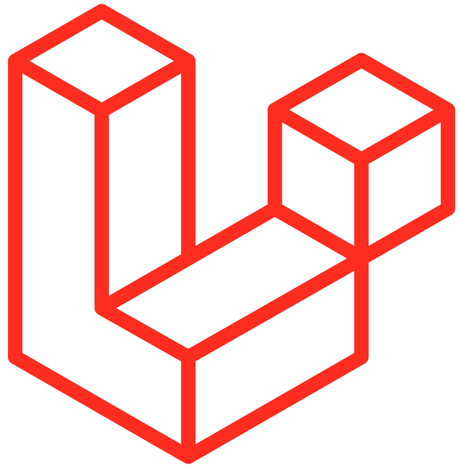
Files - Controller - Store



https://laravel.com/docs/7.x/filesystem#file-uploads

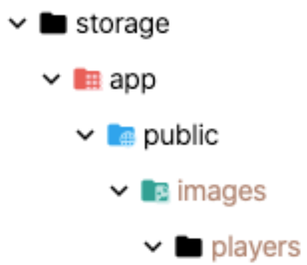
```php
/**
 * Store a newly created resource in storage.
 *
 * @param  \Illuminate\Http\Request  $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $this->validate($request, [
        'name'        => 'required',
        'address'     => 'required',
        'description' => 'required',
        'retired'     => 'required',
        'image'       => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
    ]);

    $player              = new Player();
    $player->name        = $request->name;
    $player->address     = $request->address;
    $player->description = $request->description;
    $player->retired     = $request->retired;
    $player->save();

    //If we have an image file, we store it, and move it in the database
    if ($request->file('image')) {

        // Get Image File
        $imagePath = $request->file('image');

        // Define Image Name
        $imageName =  $player->id . '_' .  time() . '_' .  $imagePath->getClientOriginalName();

        // Save Image on Storage
        $path = $request->file('image')->storeAs('images/players/' . $player->id, $imageName, 'public');

        //Save Image Path
        $player->image = $path;
    }

    $player->save();

    return redirect('players')->with('status','Item created successfully!');
}
```

Files - Create Symbolic link



https://laravel.com/docs/7.x/filesystem#the-public-disk

```
php artisan storage:link
```



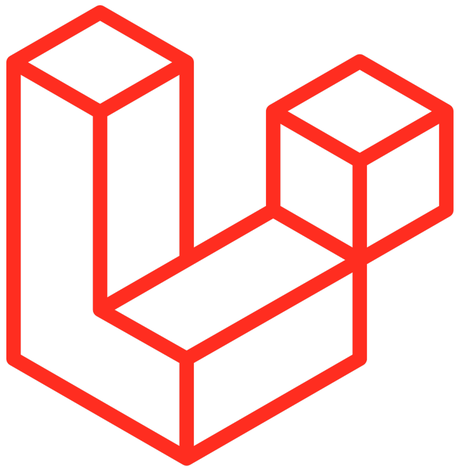
PHP

Files - List View

```
<table class="table table-striped table-bordered">
    <thead>
    <tr>
        <th scope="col">#</th>
        <th scope="col">Image</th>
        <th scope="col">Name</th>
        <th scope="col">Address</th>
        <th scope="col">Retired</th>
        <th scope="col">Actions</th>
    </tr>
    </thead>
    <tbody>

    @forelse($players as $player)

        <tr>
            <th scope="row"> {{ $player->id  }} </th>
            <td>
                @if ($player->image)
                    <img class="w-100 img-responsive" src="{{ asset('storage/'.$player->image) }}" alt="" title=""></a>
                @else
                    <p>
                        No Image
                    </p>
                @endif
            </td>
```

PHP

Files - Controller - Destroy



```php
use Illuminate\Support\Facades\Storage;


/**
 * Remove the specified resource from storage.
 *
 * @param  \App\Player  $player
 * @return \Illuminate\Http\Response
 */
public function destroy(Player $player)
{

    Storage::deleteDirectory('public/images/players/' . $player->id);
    //Storage::delete('public/' . $player->image);

    $player->delete();

    return redirect('players')->with('status','Item deleted successfully!');
}
```

https://laravel.com/docs/7.x/filesystem#deleting-files