

Redirect

Chuyển hướng home đến trang mặc định danh sách sinh viên

```
Route::redirect('/', route("students.index"), 301);
```

Paging

```
$students = Student::where("name", "LIKE", "%$search%")->paginate(2);
```

Hiển thị paging trong view: {{ \$students->links() }}

Để laravel sử dụng framework bootstrap khi tạo paging, vào

```
app\Providers\AppServiceProvider.php, add dòng
```

```
use Illuminate\Pagination\Paginator;
```

Tiếp theo thêm dòng `Paginator::useBootstrap();` vào hàm `boot()`

Để hiển thị số lượng tổng cộng số lượng sinh viên dùng `$students->total()`

Để paging có kèm thêm parameter trên thanh địa chỉ của website thì thêm

```
->withQueryString() vào paginate(2)
```

```
Cụ thể: Student::where("name", "LIKE", "%$search%")->paginate(2)->withQueryString()
```

Validate data

Sử dụng `$this->validate($request,$pattern, $messenger, $customName);`

`$request`: Là biến chứa dữ liệu đầu vào.

`$pattern`: Là mảng chứa rule kiểm tra dữ liệu đầu vào.

`$messenger`: Là mảng chứa nội dung báo lỗi.

`$customName`: Là mảng ánh xạ tên field trong form thành văn bản dễ hiểu đối với người dùng

```
protected $pattern = [
    'name' => 'required|regex:/^[a-zA-ZÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðñòóôõöùúûüýþÿ\']+$|/i|max:50',
    'birthday' => 'required|date',
    'gender' => 'bail|required|numeric|between:0,2'
];

protected $messenger = [
    'required' => ':attribute không được để trống',
    'regex' => ':attribute không được chứa số hoặc ký tự đặc biệt',
```

```
'date' => ':attribute không nhập sai định dạng ngày tháng năm',
'max' => ':attribute không được lớn hơn :max ký tự',
'between' => ':attribute phải là số từ :min đến :max',
'numeric' => ':attribute phải là số',
];

protected $customName = [
    'name' => 'Tên đăng nhập',
    'birthday' => 'Ngày sinh',
    'gender' => 'Giới tính',
];

$this->validate($request,$this->pattern, $this->messenger, $this->customName);
```

Lưu ý rằng nếu validate failed, nó sẽ tự động redirect về form (previous link)

Nâng cao:

Nếu muốn chủ động redirect đến view mình muốn thì dùng Validator::make().

Dùng trong trường hợp gom các validate lại để dùng nhiều lần (create, edit)

Hiển thị lỗi ở view

```
@if ($errors->any())
<div class="alert alert-danger">
    <ul>
        @foreach ($errors->all() as $error)
            <li>{{ $error }}</li>
        @endforeach
    </ul>
</div>
@endif
```

Thêm class is-invalid vào input/select nếu có lỗi

```
{{ $errors->has("name") ? "is-invalid" : "" }}
```

Hiển thị dữ liệu đã nhập trên form dùng hàm: old("name")

Tách error ra file view riêng

Dùng @include('error') để include file error vào view

Cache view

Nhằm tăng tốc ứng dụng (không phải compile lại view nếu code không có sự thay đổi)

View được cache trong thư mục: storage/framework/views

Xóa cache view bằng lệnh: php artisan view:clear

Session

Khi tạo thành công sinh viên. Thêm dòng code bên dưới vào controller để keep message trong session

```
$request->session()->put('success', 'Đã tạo sinh viên thành công');
```

Để hiển thị message ra trong view ta dùng:

```
request()->session()->pull('success')
```

(hàm *request()* trả về đối tượng *Request*)

(hàm *pull* sẽ lấy session ra và xóa luôn session này)

Hàm kiểm tra xem session có tồn tại không:

```
request()->session()->has('success')
```

Để lấy session và không delete session ta dùng:

```
request()->session()->get('success')
```

Ngoài ra còn 3 cách để truy xuất session:

```
session()->put('success', 'Đã tạo sinh viên thành công') hoặc
```

```
session(['success' => 'Đã tạo sinh viên thành công'])
```

```
Session::put(['success', 'Đã tạo sinh viên thành công'])
```

Phải thêm dòng `use Illuminate\Support\Facades\Session;`

Session thì được lưu trữ ở đường dẫn: storage/framework/session

Bắt lỗi liên quan đến thao tác dữ liệu dùng try/catch

Phải dùng `use Illuminate\Database\QueryException;`

```
try {
    $student->forceDelete();
    request()->session()-
>put('success', 'Đã xóa sinh viên thành công');
}
catch(QueryException $e) {
    if ($e->getCode() == 23000) {
        request()->session()-
>put('error', "Sinh viên đã đăng ký môn học, bạn không thể xóa");
    }
}
```

```
    }  
    else {  
        request()->session()->put('error', $e->getMessage());  
    }  
  
}  
return redirect()->route("students.index");
```

Tương tự như vậy triển khai subject

Active menu có nhiều cách

`request()->segment(1) == "students" //(recommened)`

hoặc

`request()->is("students*")`

hay

`$routeName = Route::currentRouteName();`

`strpos($routeName, "students") === 0`