Day 8

pada pertemuan ini, kita akan belajar bagaimana memberikan informasi atau notifikasi ketika kita sukses melakukan sebuah event. berhasil menambah data, berhasil merubah data & berhasil menghapus data.

```php
use Session;
class AuthorsController extends Controller
{
  public function store(Request $request)
  {
  $this->validate($request, ['name' => 'required|unique:authors']);
    $author = Author::create($request->only('name'));

    Session::flash("flash_notification", [
        "level"=>"success",
        "message"=>"Berhasil menyimpan $author->name"
    ]);
    return redirect()->route('authors.index');
  }
}

```

Pada syntax diatas kita membuat flash data dengan nama flash_notification dan mengisinya dengan array yang memiliki key level dan message. Key level akan kita gunakan untuk menampilkan jenis alert yang akan muncul di bootstrap. Ada 4 jenis alert yang bisa dibuat di bootstrap yaitu success, info, warning dan danger. Key message akan kita gunakan untuk mengisi pesan pada alert yang digenerate. Untuk menampilkan flash data ini, kita akan menggunakan partial view dire- sources/views/layouts/_flash.blade.php:

```php
@if (session()->has('flash_notification.message')) <div class="container">
    <div class="alert alert-{{ session()->get('flash_notification.level') }}">
      <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
      {!! session()->get('flash_notification.message') !!}
    </div>
  </div>
@endif
```

Pada view diatas, kita menggunakan session()->has() untuk mengecek apakah ada data session dengan key yang kita inginkan. Disini, kita mencari data session dengan key flash_notification.message. Setelah kita dapatkan, kita tampilkan alert dan pesannya. Agar flash message ini bisa digunakan di semua view, kita harus memanggil partial view diatas dari resources/views/layouts/app.blade.php:

```php
@include('layouts._flash')
```

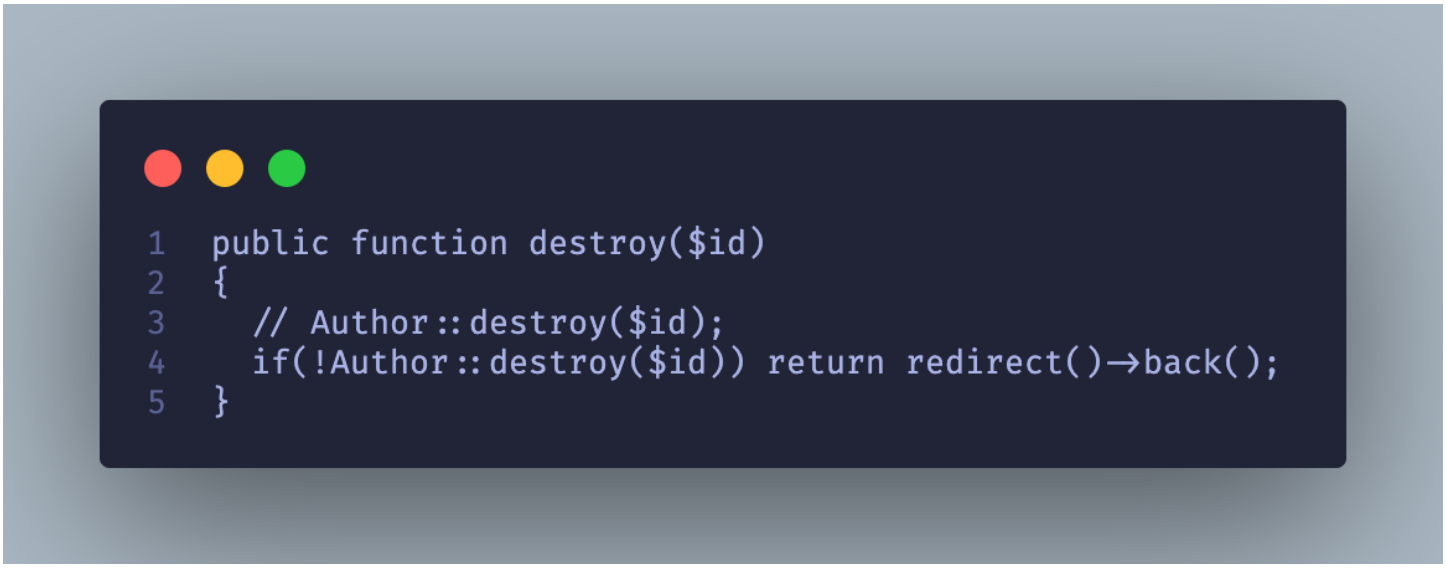
Penggunaan Model Event

Secara default, ketika kita menghapus data parent, semua data child tersebut akan dihapus. Kita akan mengubah logic ini, agar kita bisa menghentikan proses penghapusan data parent jika masih terdapat data child tersebut. Untuk membuat fitur ini, kita akan menggunakan fitur Model Event di Laravel. Dengan Model Event, kita dapat menjalankan logic pada berbagai proses CRUD yang dilakukan pada model. Pada kasus ini, kita akan menggunakan model event dengan nama deleting. Event ini akan dieksekusi ketika kita melakukan proses delete pada sebuah model. Jika kita memberikan nilai false pada event ini, maka proses penghapusan model akan dibatalkan.

```php
use Illuminate\Support\Facades\Session;
class Author extends Model
{
    public static function boot()
    {
        parent::boot();
        self::deleting(function($author) {
            // mengecek apakah penulis masih punya buku
            if ($author->books->count() > 0) {
                // menyiapkan pesan error
                $html = 'Penulis tidak bisa dihapus karena masih memiliki buku : ';
                $html .= '<ul>';
                    foreach ($author->books as $book) {
                    $html .= "<li>$book->title</li>";
                    }
                    $html .= '</ul>';
                Session::flash("flash_notification", [
                    "level"=>"danger",
                    "message"=>$html
                ]);
                // membatalkan proses penghapusan
                return false;
            }
        });
    }
}
```

Pada syntax diatas, kita menggunakan method boot untuk melakukan hook ke event deleting. Pada event tersebut, kita mengecek apakah Penulis yang sedang dihapus masih memiliki buku. Jika ya, kita set flash session dengan tipe danger dan pesan berisi detail dari buku yang masih dimiliki oleh penulis tersebut. Terakhir, kita return false agar proses penghapusan dibatalkan.

Kita harus melakukan perubahan pada method destroy di AuthorsController seperti berikut:



```php
public function destroy($id)
{
  // Author::destroy($id);
  if(!Author::destroy($id)) return redirect()->back();
}
```

Pada perubahan ini, ketika method destroy() untuk menghapus penulis menghasilkan nilai false, kita akan arahkan user ke halaman sebelumnya. Tentunya, akan muncul feedback berdasarkan flash data yang kita set pada event deleting