

## ■ جزوه آموزشی جلسه ۳۴ – One Request, One Controller

### 🎯 هدف جلسه

- مرتب‌سازی کدهای مربوط به درخواست‌های مختلف (GET, POST, DELETE) و (...).
- جداسازی وظایف هر کنترلر (اصل یک درخواست → یک کنترلر).
- ساده‌سازی ساختار پروژه و حذف شرط‌های اضافی داخل فایل‌ها.

---

### 1 حذف شرط بررسی متد درخواست

قبلاً در فایل‌های `show.php` و `destroy.php` شرط‌هایی مثل `if ($_SERVER["REQUEST_METHOD"] == "POST")` داشتیم. در این جلسه، به جای شرط گذاشتن داخل فایل، از **Router** برای تعریف متد مناسب استفاده می‌کنیم.

---

### 2 ایجاد فایل `destroy.php`

است (note) مسئول حذف کردن یادداشت.

کد نهایی:

```
<?php
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$note = $db->query(
    "SELECT * FROM notes WHERE user_id = :user AND id = :id",
    ["user" => 3, "id" => $_POST['id']]
)->findOrFail();
```

```

Authorize($note['user_id'] === 3);

$db->query(
    "DELETE FROM notes WHERE user_id = :user AND id = :id",
    ["user" => 3, "id" => $_POST['id']]
)->fetch();

header("Location: /laracast-php/public/notes");
exit();

```

نکات:

- تابع `Authorize` از فایل `functions.php` صدا زده می‌شود.
- بررسی کاربر و مالکیت یادداشت قبل از حذف.
- ریدایرکت به صفحه لیست یادداشت‌ها بعد از حذف.

### 3 تغییرات در فایل `show.php`

کد بررسی متد درخواست حذف شد. حالا فقط یادداشت را می‌گیرد و نمایش می‌دهد:

```

<?php
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$note = $db->query(
    "SELECT * FROM notes WHERE user_id = :user AND id = :id",
    ["user" => 3, "id" => $_GET['id']]
)->findOrFail();

$current_userid = 3;
Authorize($note['user_id'] === $current_userid);

```

```
view("note/show.view.php", ["note" => $note]);
```

---

#### 4) اضافه کردن مسیر (Route) حذف یادداشت

در فایل routes.php:

```
$router->delete('/laracast-php/public/notes/note',  
"controller/note/destroy.php");
```

---

#### 5) مدیریت ایجاد یادداشت (Create Note)

قبلاً با شرط متد (GET/POST) در یک فایل بود.  
الان جدا می‌کنیم:

- `create.php` فقط فرم را نمایش می‌دهد.
- `store.php` عملیات ثبت یادداشت جدید.

فایل `create.php`

```
<?php  
view("note/create.view.php");
```

فایل `store.php`

```
<?php  
use core\Validator;  
use core\Database;  
  
$config = require_once  
base_path("controller/config.php");  
$error = [];  
  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    if (Validator::Is_Empty($_POST['body'])) {  
        $error["body"] = "Body is required";  
    }  
}
```

```

    } elseif (!Validator::Count_Char($_POST["body"], 1,
1000)) {
        $error["body"] = "Body is too long";
    }

    if (empty($error)) {
        $db = new Database($config['database']);
        $db->query(
            "INSERT INTO notes (body, user_id) VALUES
(:body, :user_id)",
            ['body' => $_POST['body'], 'user_id' => 3]
        );
        $_POST['body'] = null;

        header('Location: /laracast-php/public/notes');
        exit();
    } else {
        view("note/create.view.php", ['error' =>
$error]);
    }
}

```

### تغییر فرم در `create.view.php`

```

<form method="post" action="/laracast-
php/public/notes/store">
    ...
</form>

```

## نکته مهم (اصل جلسه)

به جای ترکیب چند وظیفه در یک فایل:

- یک فایل فقط نمایش می‌دهد. (`create.php`)
- یک فایل فقط پردازش می‌کند. (`store.php`)
- یک فایل فقط حذف می‌کند. (`destroy.php`)
- و همگی این‌ها با **Router** و متد درست مدیریت می‌شوند.

## ✓جمع‌بندی

- شرط بررسی متد از کنترلرها حذف شد.
- برای هر درخواست (GET, POST, DELETE) یک کنترلر جدا نوشتیم.
- اصل **One Request = One Controller** پیاده‌سازی شد.
- کد ساده‌تر و ساختار پروژه شفاف‌تر شد.

یک فایل با نام destroy.php ایجاد کرده و کد زیر را داخلش مینویسیم:

تعریف تابع Authorize رو در فایل function.php مینویسیم و از این فایل و فایل show.php کار پاک میکنیم و چون در ابتدای فایل index فایل function را require میکنیم از این تابع به راحتی در فایل‌های destroy و show استفاده میکنیم.

شرط بررسی اینکه درخواست اگر post بود را از فایل destroy پاک کرد و هم چنین قسمت else ش را پس کد باقی مانده در destroy به صورت زیر هست:

```
<?php
    use core\Database;
    $config = require_once
        base_path("controller/config.php");
    $db = new Database($config['database'], "root",
        "");

    $note = $db->query("SELECT * FROM notes where
        user_id = :user and
        id=:id", ["user"=>3, "id"=>$_POST['id']])->findOrFail();
    Authorize($note['user_id'] === 3);

    $db->query("DELETE FROM notes where user_id = :user
        and id=:id", ["user"=>3, "id"=>$_POST['id']])->fetch();
    header("location:/laracast-php/public/notes");
    exit();
```

بعد به فایل show.php رفته و دستور if که برای بررسی متد درخواستی گذاشته بودیم پاک میکنیم پس محتوای این فایل هم به صورت زیر میشود:

```
<?php
    use core\Database;
    $config = require_once
        base_path("controller/config.php");
    $db = new Database($config['database'], "root",
        "");
    $note = $db->query("SELECT * FROM notes where
        user_id = :user and
        id=:id", ["user"=>3, "id"=>$_GET['id']])->findOrFail();
    $current_userid = 3;
    Authorize($note['user_id'] === $current_userid);
    view("note/show.view.php", ["note"=>$note]);
```

به فایل routes.php رفته و مسیر فایل کنترلر destroy با url و متد مورد نظر را اضافه میکنیم:

```
$router->delete('/laracast-php/public/notes/note' ,
    "controller/note/destroy.php");
```

حالا برای اضافه کردن یادداشت یک راه مثل جلسه قبل بود که get,post را با هم میفرستادیم وقتی post میشد اضافه کردن یادداشت درست میشد حالا ما یک فایل با نام store.php در پوشه controller/note/ ایجاد میکنیم. و درونش یه کد var\_dump(\$\_POST); نوشته به فایل create.view.php رفته و در قسمت فرم مقدار action را برابر با urlی که برای فایل store.php در فایل routes تعریف کردیم یعنی /laracast-php/public/notes/store قرار میدهیم با توجه به url ای که در فایل routes برای این عملیات تعریف کردیم حالا با کلیک بر روی دکمه ی ذخیره در فرم وارد این صفحه شده و مقدار var\_dump نمایش داده میشود.

به فایل create.php رفته و همهی کدهایش را برای فایل store.php بر میداریم تنها کدی که در این فایل باقی میماند require کردن فایل create.view.php هست پس محتوای فایل create.php به صورت زیر میباشد:

```
<?php  
view("note/create.view.php");
```

و اما محتوای فایل store.php به صورت زیر میباشد

```
<?php  
  
        use core\Validator;  
        use core\Database;  
        $config = require_once  
        base_path("controller/config.php");  
        $error = [];  
        if($_SERVER["REQUEST_METHOD"] == "POST"){  
  
            if validator::Is_Empty($_POST['body'])){  
                $error["body"] = "Body is required";  
            }else  
            if(!validator::Count_Char($_POST["body"],1,1000)){  
                $error["body"] = "body is too long ";  
            }  
  
            if(empty($error)){  
                $db = new Database($config['database']);  
                $db->query("INSERT INTO notes  
(body,user_id) VALUES (:body ,:user_id)" , ['body' =>  
                $_POST['body'], 'user_id' => 3 ]);  
                $_POST['body'] = null;  
                header('Location: /laracast-  
                php/public/notes');  
                exit();  
            }else  
            {  
  
                view("note/create.view.php", ['error'=>$error]);  
            }  
        }  
    }  
}
```

و محتوای فایل create.view.php هم به صورت زیر میباشد البته در این فایل فقط مقدار  
action تغییر کرد:

```
<?php view("partials/nav.php"); ?>  
<?php view("partials/header.php", [
```





```
                <button type="submit"
class="rounded-md bg-indigo-600 px-3 py-2 text-sm font-
                semibold text-white shadow-xs hover:bg-indigo-500
focus-visible:outline-2 focus-visible:outline-offset-2
                focus-visible:outline-indigo-600">Save</button>
                </div>
            </form>
        </div>

    </div>

<?php    view("partials/footer.php"); ?>
```