

## ◆ سؤال ۱

Refactoring یعنی چی؟

✓جواب:

بازنویسی و تمیزکاری کد بدون تغییر عملکرد نهایی، برای اینکه کد خواناتر، قابل استفاده مجدد و توسعه پذیرتر بشه.

---

## ◆ سؤال ۲

چرا جفری گفت فرمها (Form Classes) رو نباید داخل پوشه‌ی core بذاریم؟

✓جواب:

چون core مخصوص بخش‌های عمومی و زیرساختی پروژه است که در تمامی پروژه‌ها می‌شه استفاده کرد (مثل routing ، database ، validation). فرمها خاص همین اپلیکیشن هستن.

---

## ◆ سؤال ۳

جفری در نهایت فرمها و کنترلرها رو کجا قرار داد؟

✓جواب:

درون پوشه‌ی Http (داخل پروژه) چون مربوط به بخش ورود کاربر و لاجیک‌های خاص همین برنامه هستن.

---

## ◆ سؤال ۴

وقتی کنترلرها رو جابه‌جا کرد چرا ارور گرفتیم؟

✓جواب:

چون مسیر کنترلرها در فایل `routes.php` و `Router` هنوز به مسیر قبلی اشاره می‌کرد و باید تغییر داده می‌شد.

---

#### ◆سؤال ۵

چطور مشکل مسیر کنترلرها حل شد؟

✓جواب:

- در `routes.php` مسیر کامل حذف شد و فقط نام فایل کنترلر نوشته شد.
  - در `Router.php` تغییر داده شد تا همیشه کنترلرها رو از مسیر `Http/controllers` بخونه.
- 

#### ◆سؤال ۶

وظیفه‌ی کلاس `LoginForm` چی بود؟

✓جواب:

مدیریت اعتبارسنجی (Validation) فرم ورود و نگهداری خطاها (Errors) مربوط به آن.

---

#### ◆سؤال ۷

چرا جفری گفت متد `validate` نباید View برگردونه؟

✓جواب:

چون وظیفه‌ی فرم فقط اعتبارسنجی است. نمایش View جزو مسئولیت فرم نیست بلکه وظیفه‌ی Controller است.

---

## ◆ سؤال ۸

متد `validate` در نهایت چی برمی‌گردوند؟

✓ جواب:

یک مقدار  $\rightarrow$  Boolean (true/false) اگر آرایه‌ی خطاها خالی بود ( `true` اعتبارسنجی موفق)، در غیر این صورت `false`.

---

## ◆ سؤال ۹

چطور خطاها (Errors) در فرم ذخیره شدند؟

✓ جواب:

در یک پراپرتی `protected $errors = []`؛ ذخیره شدند.

---

## ◆ سؤال ۱۰

چرا از Getter برای خطاها استفاده شد؟

✓ جواب:

چون بهتره `errors` مستقیم `public` نباشه. با متد `errors()` فقط خواندن آن ممکن میشه و اگر در آینده نیاز به تغییر یا پردازش اضافه داشتیم، جای مناسبی برای اضافه کردن اون داریم.

## تمرینات

### ◆ سؤال ۱

کلاس LoginForm رو بساز که دو ورودی email و password بگیره و متد validate() داشته باشه.

✓ جواب:

```
<?php

namespace Http\Forms;

class LoginForm
{
    protected $errors = [];

    public function validate($email, $password)
    {
        if (! filter_var($email,
FILTER_VALIDATE_EMAIL)) {
            $this->errors['email'] = 'Email is
invalid.';
        }

        if (strlen($password) < 7) {
            $this->errors['password'] = 'Password must
be at least 7 characters.';
        }

        return empty($this->errors);
    }

    public function errors()
    {
        return $this->errors;
    }
}
```

---

## ◆ سؤال ۲

چطور این کلاس رو در کنترلر sessions/store.php صدا می‌زنیم؟

✓ جواب:

```
use Http\Forms\LoginForm;

$form = new LoginForm();

if (! $form->validate($_POST['email'],
$_POST['password'])) {
    return view('sessions/create.view.php', [
        'errors' => $form->errors()
    ]);
}

// ادامه‌ی لاگین
```

---

## ◆ سؤال ۳

کدی بنویس که اگر خطای اعتبارسنجی وجود داشت، پیام‌ها رو در صفحه نمایش بده.

✓ جواب) در: view)

```
<?php if (!empty($errors)) : ?>
    <ul>
        <?php foreach ($errors as $error) : ?>
            <li><?= htmlspecialchars($error) ?></li>
        <?php endforeach; ?>
    </ul>
<?php endif; ?>
```

---

## ◆ سؤال ۴

فرض کن می‌خواهی بررسی کنی که پسورد حتماً شامل عدد هم باشد. چطور به کلاس LoginForm اضافه می‌کنی؟

✓ جواب:

```
if (! preg_match('/[0-9]/', $password)) {  
    $this->errors['password'] = 'Password must contain  
at least one number.';  
}
```

---

#### ◆ سؤال ۵

اگر بخوای متد validate به جای Boolean، خود آرایه‌ی errors رو برگردونه باید چطور تغییر بدی؟

✓ جواب:

```
public function validate($email, $password)  
{  
    // ... اعتبارسنجی‌ها  
  
    return $this->errors;  
}
```

---

#### ◆ سؤال ۶

چرا استفاده از Getter برای errors بهتر از public کردن پراپرتی \$errors است؟  
(جواب کدنویسی + توضیح کوتاه)

✓ جواب:

```
public function errors()  
{  
    return $this->errors;  
}
```

◆ چون با Getter می‌تونیم بعداً تغییرات اضافه کنیم (مثلاً فرمت کردن خطاها) بدون اینکه بقیه‌ی کد پروژه تغییر کنه.

---

#### ◆ سؤال ۷

کدی بنویس که اگر کاربر ایمیل وارد نکرد، خطای مخصوص نمایش بده.

✓ جواب:

```
if (trim($email) === '') {  
    $this->errors['email'] = 'Email is required.';  
}
```

---

#### ◆ سؤال ۸

اگر بخوای از LoginForm برای اعتبارسنجی یک فرم دیگه (مثلاً ثبت‌نام) استفاده کنی، چه تغییر ساده‌ای باید بدی؟

✓ جواب:

ایجاد یک کلاس جدید مثل RegisterForm که ساختارش مشابه LoginForm باشه ولی قوانین اعتبارسنجی مخصوص خودش رو داشته باشه.

---

#### ◆ سؤال ۹

چطور می‌تونی طول پسورد رو به صورت حداقل ۷ و حداکثر ۲۰ کاراکتر اعتبارسنجی کنی؟

✓ جواب:

```
if (strlen($password) < 7 || strlen($password) > 20) {  
    $this->errors['password'] = 'Password must be  
between 7 and 20 characters.';
```

}

---

### ◆ سؤال ۱۰

در کنترلر اگر فرم معتبر بود (یعنی true برگشت)، کدی بنویس که کاربر رو به صفحه‌ی اصلی (/) هدایت کنه.

✓ جواب:

```
if ($form->validate($_POST['email'],  
$_POST['password'])) {  
    header('Location: /');  
    exit;  
}
```