

آموزش کامل Session در - PHP جلسه ۳۷ Laracast

مقدمه

این جلسه شروع فصل جدیدی در مورد Session Handling و Authentication است. مدرس در ابتدا از شما به خاطر تداوم در یادگیری تشکر می‌کند و می‌گوید که خیلی از افراد خیلی زود تسلیم می‌شوند ولی شما همچنان ادامه داده‌اید.

تعریف Session

Session به معنای دوره‌ای است که به فعالیت خاصی اختصاص می‌یابد. در وبسایت، Session دوره‌ای است که کاربر با سایت شما تعامل می‌کند. هر کاربر Session مخصوص به خودش دارد که با Session های دیگران متفاوت است.

مثال عملی – قدم به قدم

گام ۱: ایجاد Session در کنترلر index

ابتدا وارد فایل کنترلر index.php شده و کد زیر را نوشتیم:

```
$_SESSION['name'] = "Mohammad";
```

گام ۲: خواندن Session در کنترلر about

سپس در فایل کنترلر about.php کد زیر را نوشتیم:

```
var_dump($_SESSION['name']);
```

گام ۳: خطای اولیه

با کلیک روی منوی about ، خطای زیر را دریافت کردیم:

```
Warning: Undefined global variable $_SESSION
Warning: Trying to access array offset on value of type
null
NULL
```

گام ۴: حل مشکل با session_start()

برای حل این مشکل، در فایل public/index.php کد زیر را اضافه کردیم:

```
session_start();
```

گام ۵: نتیجه موفق

حالا با اجرای مجدد پروژه و کلیک روی about ، خروجی زیر را دریافت کردیم:

```
string(8) "Mohammad"
```

نکات مهم در مورد Session ها

1. موقت بودن Session ها

- Session ها موقت هستند و برای همیشه ذخیره نمی‌شوند
- با بستن مرورگر معمولاً حذف می‌شوند
- یا بعد از مدت زمان مشخصی (مثل ۳۰ دقیقه یا یک روز) منقضی می‌شوند

2. session_start() ضروری است

- قبل از کار با \$_SESSION، حتماً باید session_start() را فراخوانی کنید
- بهتر است در ابتدای اپلیکیشن (مثل index.php) این کار را انجام دهید

مثال پیشرفته‌تر

نمایش نام کاربر در Banner

در فایل views/partials/banner.php:

```
<?php echo "Hello " . $_SESSION['name']; ?>
```

مدیریت حالت عدم وجود Session

```
<?php echo "Hello " . ($_SESSION['name'] ?? 'Guest');  
?>
```

این کد اگر Session وجود داشته باشد نام کاربر را نمایش می‌دهد، وگرنه "Guest" نمایش می‌دهد.

مکانیزم کارکرد Session ها

Session ها از دو قسمت تشکیل شده‌اند:

1. کوکی در مرورگر (Client Side)

- نام کوکی PHPSESSID :
- شامل شناسه Session است
- در Developer Tools > Storage > Cookies قابل مشاهده است

2. فایل در سرور (Server Side)

- فایلی در سرور ذخیره می‌شود که حاوی داده‌های Session است
- معمولاً در پوشه tmp / ذخیره می‌شود
- نام فایل با شناسه کوکی همخوانی دارد

یافتن محل ذخیره Session ها

برای پیدا کردن محل ذخیره فایل‌های Session:

```
php -i | grep "session.save_path"
```

مثال پیدا کردن فایل Session

۱. در Developer Tools ، شناسه Session مثل (u82k...) را پیدا کنید
۲. در پوشه temp سرور، فایل با همین نام را پیدا کنید
۳. محتوای فایل مثل این است:

```
name|s:7:"Jeffrey";
```

اضافه کردن داده‌های بیشتر

در کنترلر دیگر مثل `contact.php`

```
$_SESSION['last'] = "Way";
```

محتوای فایل Session تبدیل می‌شود به:

```
name|s:7:"Jeffrey";last|s:3:"Way";
```

حذف کوکی و تأثیر آن

- اگر کوکی `PHPSESSID` را حذف کنید، Session قدیمی از دست می‌رود
- PHP تشخیص می‌دهد که کوکی وجود ندارد و Session جدیدی ایجاد می‌کند
- فایل Session قدیمی همچنان در سرور باقی می‌ماند (تا زمان cleanup)

نکات امنیتی و عملی

مزایا:

- داده‌ها از صفحه‌ای به صفحه دیگر منتقل می‌شوند
- هر کاربر Session جداگانه دارد
- داده‌ها در سرور ذخیره می‌شوند (امن‌تر از کوکی‌های معمولی)

محدودیت‌ها:

- موقت هستند
- وابسته به کوکی مرورگر
- با بستن مرورگر یا انقضای زمان حذف می‌شوند

خلاصه

Session ها ابزار قدرتمندی برای نگهداری اطلاعات موقت کاربر در طول بازدید از وبسایت هستند. در جلسات بعدی، نحوه حذف Session ها و ساخت سیستم ورود کاربر پوشش داده خواهد شد.

نکته مهم: همیشه `session_start()` را قبل از کار با `$_SESSION` فراخوانی کنید!

جریان گردش Session و Cookie در PHP

۱. شروع درخواست (مثلاً اولین ورود به سایت):

- کاربر از مرورگر یک درخواست ارسال می‌کند (مثلاً ورود یا دیدن یک صفحه).
- سرور با اجرا شدن `session_start()` یک آیدی منحصر به فرد (`session id`) ایجاد می‌کند.
- فایل سشن جدید با این آیدی در مسیر مخصوص روی سرور (معمولاً `tmp/`) ساخته می‌شود و هر داده‌ای که در `$_SESSION` ذخیره کنیم، واقعاً داخل این فایل نوشته می‌شود.

۲. برگشت پاسخ به مرورگر:

- سرور آیدی جلسه را در قالب یک کوکی (`PHPSESSID`) به مرورگر می‌فرستد.
- مرورگر این کوکی را نگه می‌دارد و همراه درخواست‌های بعدی برای همین سایت، به سرور می‌فرستد.

۳. درخواست‌های بعدی (هر وقتی کاربر صفحه جدید را باز می‌کند):

- مرورگر همیشه کوکی `PHPSESSID` را در هدر درخواست ارسال می‌کند.
- سرور با توجه به مقدار این کوکی، فایل سشن منطبق را پیدا کرده و داده‌های قبلی آن کاربر را به `$_SESSION` وارد می‌کند.
- هر تغییر جدید باز هم در همان فایل ذخیره می‌شود.

۴. حذف یا انقضای سشن یا کوکی:

- اگر کاربر کوکی `PHPSESSID` را حذف کند یا تاریخ انقضای آن گذشته باشد یا مرورگر بسته شود (طبق تنظیمات)، در درخواست بعدی سرور متوجه می‌شود کاربر آیدی ندارد و مجدداً یک سشن جدید می‌سازد.

شکل شماتیک جریان (توضیح به زبان ساده)



نکته مهم نهایی:

- کوکی PHPSESSID فقط شناسه‌ی آیدی است و هیچ داده مهم دیگری را ذخیره نمی‌کند.
- تمام داده‌های اصلی شما فقط و فقط روی سرور و در فایل سشن ذخیره می‌شود (نه در مرورگر کاربر).
- اگر کوکی از بین برود، سشن جدید ساخته می‌شود و داده قبلی آن کاربر دیگر قابل دسترسی نیست.

۱. (مرورگر) ---- درخواست صفحه ----> (سرور)

۲. session_start()]]

۳. (سرور)

|

| -- آیا کوکی PHPSESSID موجود است؟

|

| --- بله: session مربوط را پیدا می‌کند، داده‌ها را می‌خواند.

|--- خیر: آیدی جدید می‌سازد، فایل جدید session روی سرور می‌سازد.

|

\--- داده‌ای در SESSION_\$ ذخیره کردی ← داخل فایل سشن ذخیره می‌شود.

۴. (سرور) ----پاسخ+کوکی (>----PHPSESSIDمرورگر)

۵. (مرورگر) برای هر درخواست جدید ← کوکی sessionid را همراه درخواست می‌فرستد.

۶. (سرور) طبق sessionid، فایل داده را می‌خواند و داده قبلی را بازیابی می‌کند.