

Web Technologies

Lecture#26

Muhammad Kamran

Sessions & Cookies

1. **Cookies**
2. **Session**
3. **Session in laravel**

What is a Cookie?

- ◆ A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values

Create Cookies With PHP

- ◆ A cookie is created with the `setcookie()` function.
- ◆ Syntax
 - ◆ `setcookie(name, value, expire, path, domain, secure, httponly);`
- ◆ Only the *name* parameter is required. All other parameters are optional.

PHP Create/Retrieve a Cookie

- ◆ The following example creates a cookie named "user" with the value "John Doe". The cookie will expire after 30 days ($86400 * 30$). The "/" means that the cookie is available in entire website (otherwise, select the directory you prefer).
- ◆ We then retrieve the value of the cookie "user" (using the global variable `$_COOKIE`). We also use the `isset()` function to find out if the cookie is set:

- ◆ `<?php`
- ◆ `$cookie_name = "user";`
- ◆ `$cookie_value = "John Doe";`
- ◆ `setcookie($cookie_name, $cookie_value, time() + (86400*30), "/"); // 86400 = 1 day`
- ◆ `?>`
- ◆ `<html>`
- ◆ `<body>`
- ◆ `<?php`
- ◆ `if(!isset($_COOKIE[$cookie_name])) {`
- ◆ `echo "Cookie named '" . $cookie_name . " not set!";`
- ◆ `} else {`
- ◆ `echo "Cookie '" . $cookie_name . "' is set!
";`
- ◆ `echo "Value is: " . $_COOKIE[$cookie_name];`
- ◆ `}`
- ◆ `?>`
- ◆ `</body>`
- ◆ `</html>`

- ◆ A session is a way to store information (in variables) to be used across multiple pages.
- ◆ Unlike a cookie, the information is not stored on the users computer.

What is a PHP Session?

- ◆ When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.
- ◆ Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.
- ◆ So; Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

- ◆ A session is started with the `session_start()` function.
- ◆ Session variables are set with the PHP global variable: `$_SESSION`.
- ◆ Now, let's create a new page
- ◆ called "demo_session1.php". In this page,
- ◆ we start a new PHP session and
- ◆ set some session variables:

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set";
?>
```

Note: The `session_start()` function must be the very first thing in your document. Before any HTML tags.

```
</body>
```

Get PHP Session Variable Values

- ◆ Next, we create another page called "demo_session2.php".
- ◆ From this page, we will access the session information we set on the first page ("demo_session1.php").
- ◆ Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (session_start()).
- ◆ Also notice that all session variable values are stored in the global `$_SESSION` variable:

Example -1

◆ `<?php`

```
session_start();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
// Echo session variables that were set on previous page
```

```
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
```

```
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
```

```
?>
```

```
</body>
```

```
</html>
```

Example -2

```
◆ <?php
    session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
    print_r($_SESSION);
?>

</body>
</html>
```

- ◆ If you want to remove certain session data, simply unset the corresponding key of the `$_SESSION` associative array, as shown in the following example:

- `<?php`
- `// Starting session`
- `session_start();`
- `// Removing session data`
- `if(isset($_SESSION["lastname"])){`
- `unset($_SESSION["lastname"]);`
- `} ?>`

session_set_cookie_params

- ◆ `session_set_cookie_params('3600');` // 1 hour
- ◆ `session_set_cookie_params('600');` // 10 minutes.
- ◆ Example:

 - ◆ `<?php`
 - ◆ `$lifetime=600;`
 - ◆ `session_set_cookie_params($lifetime);`
 - ◆ `session_start();`
 - ◆ `?>`

Session in Laravel

- ◆ Laravel provides various drivers like **file, cookie, apc, array, Memcached, Redis, and database** to handle session data.
- ◆ By default, file driver is used because it is lightweight. Session can be configured in the file stored at **config/session.php**.

Data can be stored in session using the `put()` method. The `put()` method will take two arguments, the “key” and the “value”.

```
$request->session()->put('key', 'value');
```

To access the session data, we need an instance of session which can be accessed via HTTP request. After getting the instance, we can use the **get()** method, which will take one argument, "**key**", to get the session data.

```
$value = $request->session()->get('key');
```

You can use **all()** method to get all session data instead of **get()** method.

The **forget()** method is used to delete an item from the session. This method will take “**key**” as the argument.

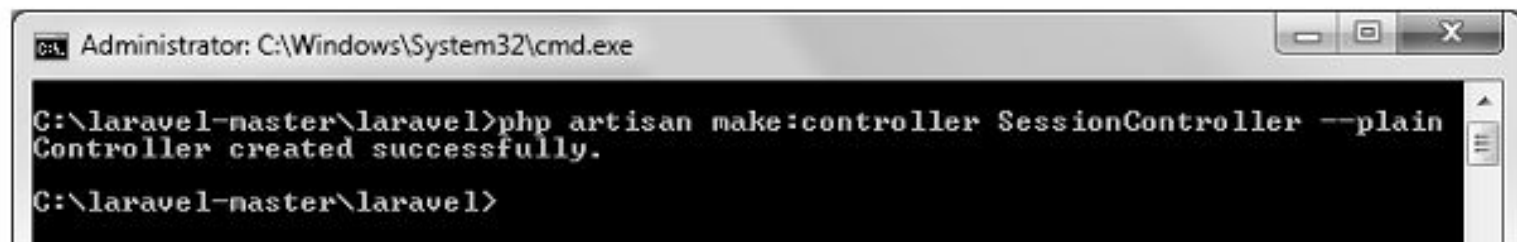
```
$request->session()->forget('key');
```

Use **flush()** method instead of **forget()** method to delete all session data. Use the **pull()** method to retrieve data from session and delete it afterwards. The **pull()** method will also take **key** as the argument. The difference between the **forget()** and the **pull()** method is that **forget()** method will not return the value of the session and **pull()** method will return it and delete that value from session.

Step 1 – Create a controller called **SessionController** by executing the following command.

```
php artisan make:controller SessionController --plain
```

Step 2 – After successful execution, you will receive the following output –



```
Administrator: C:\Windows\System32\cmd.exe

C:\laravel-master\laravel>php artisan make:controller SessionController --plain
Controller created successfully.

C:\laravel-master\laravel>
```

Step 3 – Copy the following code in a file at
`app/Http/Controllers/SessionController.php`.
`app/Http/Controllers/SessionController.php`

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;

class SessionController extends Controller {
    public function accessSessionData(Request $request) {
        if($request->session()->has('my_name'))
            echo $request->session()->get('my_name');
        else
            echo 'No data in the session';
    }
}
```


Step 4 – Add the following lines at **app/Http/routes.php** file.

app/Http/routes.php

```
Route::get('session/get', 'SessionController@accessSessionData');  
Route::get('session/set', 'SessionController@storeSessionData');  
Route::get('session/remove', 'SessionController@deleteSessionData');
```

Step 5 – Visit the following URL to **set data in session**.

```
http://localhost:8000/session/set
```

Step 6 – The output will appear as shown in the following image.



Data has been added to session

Step 7 – Visit the following URL to **get data from session**.

```
http://localhost:8000/session/get
```

Step 8 – The output will appear as shown in the following image.



Test

Step 9 – Visit the following URL to **remove session data**.

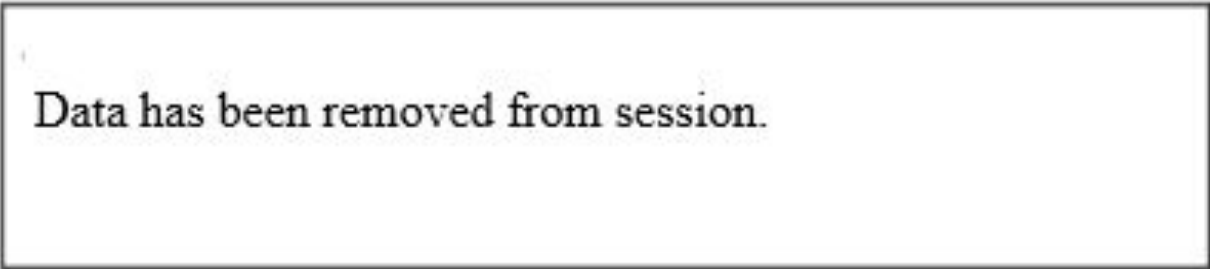
```
http://localhost:8000/session/remove
```

Step 10 – You will see a message as shown in the following image.

Step 9 – Visit the following URL to remove session data.

```
http://localhost:8000/session/remove
```

Step 10 – You will see a message as shown in the following image.



Data has been removed from session.



Questions?