A blog on PHP, JavaScript, and more

## Logout from every devices except the currently logged in one in Laravel

Laravel

PHP

**JavaScript** 

Git

React

Amit Merchant · May 5, 2019 · \* Laravel

through which you can choose to logout from all the devices you've been logged in from previously, except the current one. This is certainly a nice security feature that you can provide to your user without them even asking for it and if your app is built on top of Laravel 5.6, this comes out-of-the-box.

Here's how you can accomplish this. First of all, you need to uncomment the line \Illuminate\Session\Middleware\AuthenticateSession::class from \$middlewareGroups property in app/Http/Kernel.php file because this is the middleware that manages the user sessions in Laravel.

```
COPY
 * The application's route middleware groups.
 * @var array
protected $middlewareGroups = [
    'web' => [
        \Illuminate\Session\Middleware\AuthenticateSession::class,
    ],
];
```

user's password as its only argument. You can collect the password through an input form.

Next, you can use logoutOtherDevices method on the Auth facade which takes

```
use Illuminate\Support\Facades\Auth;
   Auth::logoutOtherDevices($password);
This is it! This will get your job done of logging out the user from all its active logged
```

in devices except the the current one.

## Now, Let's take a look at the logoutOtherDevices method's implementation which

**Behind the scenes** 

you can find in Illuminate\Auth\SessionGuard class.

```
/**
 * Invalidate other sessions for the current user.
 * The application must be using the AuthenticateSession middleware.
 * @param string $password
 * @param string $attribute
 * @return bool|null
public function logoutOtherDevices($password, $attribute = 'password')
   if (! $this->user()) {
       return;
   $result = tap($this->user()->forceFill([
       $attribute => Hash::make($password),
   ]))->save();
   $this->queueRecallerCookie($this->user());
   $this->fireOtherDeviceLogoutEvent($this->user());
   return $result;
```

As you can see, the method accepts two arguments. The first one being the password of the logged in user and the second one being an optional one which is

return;

Let's break down the function in chunks to understand what it's doing.

an attribute called 'password'. if (! \$this->user()) {

```
The method first checks if the user is logged in or not. If it's not logged in, it returns
immediately and continue execution if otherwise.
```

\$result = tap(\$this->user()->forceFill([ \$attribute => Hash::make(\$password),

```
]))->save();
The forceFill method here effectively ignores $fillable array on the User model
and will update the model with the array key-value pairs that you pass to it which is
```

in our case is 'password'. Then the result will be catched by tap method which is

then used to save it onto the **user** model. \$this->queueRecallerCookie(\$this->user()); It will then create a cookie for the existing session using queueRecallerCookie

```
method and queue it into the cookie jar.
   $this->fireOtherDeviceLogoutEvent($this->user());
```

invalidate user's other sessions entirely, meaning they will be "logged out" of all guards they were previously authenticated by. You can check this by looking at the fireOtherDeviceLogoutEvent method's implementation.

\* Fire the other device logout event if the dispatcher is set.

This will fire the other device logout event if the dispatcher is set and this will

```
* @param \Illuminate\Contracts\Auth\Authenticatable $user
 * @return void
protected function fireOtherDeviceLogoutEvent($user)
    if (isset($this->events)) {
        $this->events->dispatch(new Events\OtherDeviceLogout(
            $this->name, $user
        ));
Learn the fundamentals of PHP 8 (and 8.1), the latest version of PHP, and how to use it
today with my new book PHP 8 in a Nutshell. It's a no-fluff and easy-to-read guide to the
```

» Share: Twitter, Facebook, Hacker News

Beep! Beep! I'm also running a YouTube channel which I hope you're going to love!

latest features and nitty-gritty details of PHP 8. So, if you're looking for a quick and easy way

to PHP 8, this is the book for you.

**Accessing Laravel documentation** 

from the terminal

Grab your copy now!

```
Hi there! I'm Amit. I write articles about all things
web development. If you like what I write and want
me to continue doing the same, I would like you
buy me some coffees. I'd highly appreciate that.
Cheers!
                             Join my newsletter
    Buy me a coffee
    Comments?
```

More on similar topics **Browse articles by category** Discard Eloquent model changes in **Why Developers Choose Cloudways** For Laravel Hosting Laravel 9.x

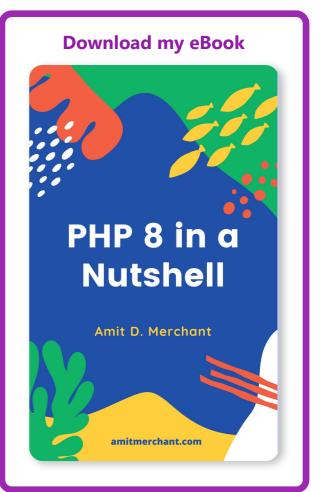
The artisan about command in

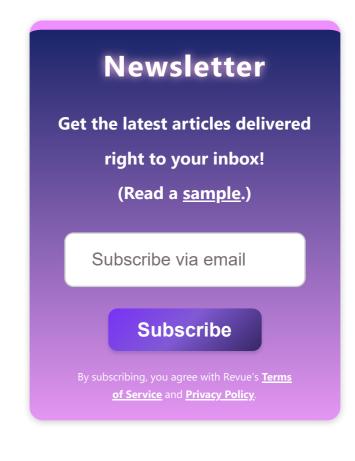
Laravel 9.x

Read a sample

**Focus on Laravel Not on Laravel** Hosting GET STARTED

Articles Projects Uses Art Stats About 🔎





Follow me everywhere



in Laravel 9.x **Why Developers Choose Cloudways For Laravel Hosting Accessing Laravel documentation** from the terminal The artisan about command in Laravel 9.x New invokable rule objects in Laravel 9.x







**About Contact Categories Guestbook** Copyright © 2022 — Amit Merchant