

Laravel - Pagination Customizations

Laravel includes a feature of pagination which helps a user or a developer to include a pagination feature. Laravel paginator is integrated with the query builder and Eloquent ORM. The paginate method automatically takes care of setting the required limit and the defined offset. It accepts only one parameter to paginate i.e. the number of items to be displayed in one page.

Laravel 5.7 includes a new pagination method to customize the number of pages on each side of the paginator. The new method no longer needs a custom pagination view.

The custom pagination view code demonstration is mentioned below –

```
<?php

namespace App\Http\Controllers;

use Illuminate\Support\Facades\DB;

use App\Http\Controllers\Controller;

class UserController extends Controller{

    /**
     * Show all of the users for the application.
     *
     * @return Response
     */

    public function index() {

        $users = DB::table('users')->paginate(15);

        return view('user.index', ['users' => $users]);

    }

}
```

The new pagination customization as per Laravel standards is mentioned below –

```
<?php

User::paginate(10)->onEachSide(5);
```

Laravel dump server comes with the version of Laravel 5.7. The previous versions do not include any dump server. Dump server will be a development dependency in laravel/laravel composer file.

With release of version 5.7, you'll get this command which includes a concept out-of-thebox which allows user to dump data to the console or an HTML file instead of to the browser. The command execution is mentioned below –

```
php artisan dump-server
```

```
# Or send the output to an HTML file
```

```
php artisan dump-server --format=html > dump.html
```

Explanation

The command runs a server in the background which helps in collection of data sent from the application, that sends the output through the console. When the command is not running in the foreground, the dump() function is expected to work by default.

Laravel - Action URL

Laravel 5.7 introduces a new feature called callable action URL. This feature is similar to the one in Laravel 5.6 which accepts string in action method. The main purpose of the new syntax introduced in Laravel 5.7 is to directly enable you access the controller.

The syntax used in Laravel 5.6 version is as shown –

```
<?php
```

```
$url = action('UserController@profile', ['id' => 1]);
```

The similar action called in Laravel 5.7 is mentioned below –

```
<?php
```

```
$url = action([PostsController::class, 'index']);
```

One advantage with the new callable array syntax format is the feature of ability to navigate to the controller directly if a developer uses a text editor or IDE that supports code navigation.