ItSolutionStuff.com (https://itsolutionstuff.com)

Laravel 5 Repository Pattern Tutorial from scratch

By Hardik Savani (https://www.linkedin.com/in/savanihd) | April 13, 2016 | | 139202 Viewer | Category : Laravel

In this post i want to share with you how to create Repository Pattern in Laravel 5 application. generally we are getting data directly from model, I mean we are use just MVC. But if you have big application then it is better way if you use Repository Pattern. I will give you few step to create Repository Pattern. normally we use directly from Model but if you make Repository Pattern for every module then it is good way to develop laravel application.

Step 1: Create Interface

In first step we have to create Interface, before create Repositories(**app/Repositories**) directory in app folder. Also we need top create User(**app/Repositories/User**) folder inside Repositories folder.

Ok, now first we will create UserInterface in User directory, so first create UserInterface.php file and put bellow code in that file:

app/Repositories/User/UserInterface.php

```
namespace App\Repositories\User;

interface UserInterface {

    public function getAll();

    public function find($id);

    public function delete($id);
}
```

Step 2: Create Repository

Ok, in this step we will create UserRepository.php for write database login, in this file we will write our database login code. so, first create UserRepository.php file in User directory and put bellow code.

app/Repositories/User/UserRepository.php

```
namespace App\Repositories\User;

use App\Repositories\User\UserInterface as UserInterface;
use App\User;

class UserRepository implements UserInterface
{
    public $user;

    function __construct(User $user) {
        $this->user = $user;
    }

    public function getAll()
    {
        return $this->user->getAll();
    }

    public function find($id)
    {
        return $this->user->findUser($id);
    }

    public function delete($id)
    {
        return $this->user->deleteUser($id);
    }
}
```

Step 3: Create Service Provider

In this step we have to create Service Provider for bind UserInterface and UserRepository class. so let's create UserRepoServiceProvide.php file in User folder and put following code:

app/Repositories/User/UserRepoServiceProvide.php

```php
namespace App\Repositories\User;

use Illuminate\Support\ServiceProvider;

class UserRepoServiceProvide extends ServiceProvider
{
    /**
     * Bootstrap the application services.
     *
     * @return void
     */
    public function boot()
    {

    }

    /**
     * Register the application services.
     *
     * @return void
     */
    public function register()
    {
        $this->app->bind('App\Repositories\User\UserInterface', 'App\Repositories\User\UserRepository');
    }
}
```

Now, we have to add server provide in app.php file, so add UserRepoServiceProvide in **config/app.php** and add bellow line.

config/app.php

```php
return [
        ....
        'providers' => [
                ......,
                App\Repositories\User\UserRepoServiceProvide::class,
        ]
        ....
]
```

Step 4: Create User Model

We have already User Model i think, so you have just need to add getAll(), findUser() and deleteUser() function. But you can also past bellow code too. So, let's put bellow code on your model.

app/User.php

```php
namespace App;

use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    protected $fillable = [
        'name', 'email', 'password',
    ];

    protected $hidden = [
        'password', 'remember_token',
    ];

    public function getAll()
    {
        return static::all();
    }

    public function findUser($id)
    {
        return static::find($id);
    }

    public function deleteUser($id)
    {
        return static::find($id)->delete();
    }
}
```

Step 5: Use In Controller

Now we are ready to use Our Repository Pattern in our Controller. So, let's add bellow code in your UserController.php file.

app/Http/Controllers/UserController.php

```php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Repositories\User\UserInterface as UserInterface;

class UserController extends Controller
{

    public function __construct(UserInterface $user)
    {
        $this->user = $user;
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $users = $this->user->getAll();
        return view('users.index',['users']);
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $user = $this->user->find($id);
        return view('users.show',['user']);
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function delete($id)
    {
        $this->user->delete($id);
        return redirect()->route('users');
    }
}
```

Now at last just fire bellow command because we have to auto load class.

```
composer update
php artisan optimize
```

laravel repository pattern, repository pattern in mvc, repository pattern in laravel 5.2, repository pattern laravel example, repository pattern php laravel, repository pattern laravel 4, repository design pattern laravel, laravel repository pattern relationships, laravel repository pattern tutorial, repository pattern laravel example, laravel repo, laravel repository, laravel repository tutorial laravel 5 interfaces, laravel 5 user interface, laravel 5 repository service provider

Have you any Question ? Then Ask Here.....

(http://nicesnippets.com/forums)

***Do you want me hire for your Project Work? Then Contact US. (/feedback)
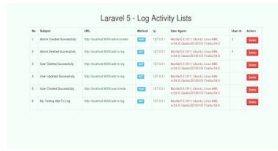
Featured Post

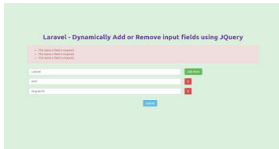**PHP Laravel 5.6 - Rest API with Passport Tutorial**
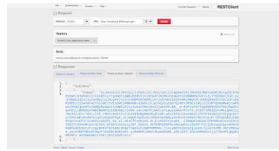
Laravel 5 - How to create API Authentication using...

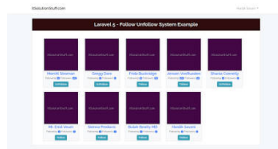**Custom User Log Activity in Laravel 5 App Example**

Laravel 5 - Column sorting with pagination example from scratch

Laravel 5 Tag - It Solution Stuff

PHP Laravel 5 - Like Dislike System Tutorial

Laravel 5 - Follow Unfollow System Example

Laravel 5 - Twitter API using thujohn/twitter composer package...