**GIÁO TRÌNH LARAVEL - BUỔI 15**

**CACHE**

1. **Cache là gì**

*Cache hay còn được gọi là bộ nhớ đệm về cơ bản là một lớp ở giữa website và cơ sở dữ liệu của bạn. Những dữ liệu được lưu trữ trong bộ nhớ cache có thể là kết quả của những lần xử lý trước hoặc là bản sao dữ liệu được lưu trữ ở nơi khác (ví dụ trong cơ sở dữ liệu). Các truy vấn tới cơ sở dữ liệu có thể bị chậm nếu bạn có nhiều người truy cập vào website cùng lúc. Lớp cache sẽ hoạt động như là trung gian để lưu trữ những dữ liệu không thay đổi giữa các request và việc lấy các thông tin từ bộ nhớ cache sẽ nhanh hơn rất nhiều hơn là việc bạn phải tính toán lại kết quả hoặc truy vấn lại tới cơ sở dữ liệu của bạn.*

1. **Yêu cầu cho cache**

a. Database

Khi sử dụng cache driver database, bạn sẽ cần thiết lập một bảng để lưu trữ các cache items.

VD: Khai báo 1 bảng cache

Schema::create('cache', function($table) {

$table->string('key')->unique();

$table->text('value');

$table->integer('expiration');

});

1. **Sử dụng Cache**

a. Tạo đối tượng bộ nhớ cache

Để sử dụng các câu lệnh Cache đầu tiên ta cần khai báo :

use Illuminate\Support\Facades\Cache;

VD: Trong 1 Controller

public function index()

{

$value = Cache::get('key');

}

}

b. Lấy dữ liệu trong bộ nhớ cache

Hàm *get* trong Cache facade được sử dụng để lấy các items trong cache. Nếu như item không tồn tại trong cache, giá trị null sẽ được trả về. Nếu muốn, bạn có thể truyền vào tham số thứ hai để hàm *get* chỉ định giá trị mặc định nếu như item không tồn tại:

$value = Cache::get('key');

$value = Cache::get('key','default');

c. Kiểm tra việc tồn tại của dữ liệu trong bộ nhớ cache

if(Cache::has('key'))

{

//Code nếu có giá trị trong bộ nhớ cache

}

d. Lưu trữ dữ liệu vào bộ nhớ cache

$minutes = 60; //Thời gian hết hạn

Cache::put('key','value',$minutes);

Ngoài ra, thay vì việc xác định số phút thì bạn có thể truyền vào một giá trị PHP Datetime để xác định thời gian hết hiệu lực của dữ liệu

VD:

$expiresAt = Carbon::now()->addDays(7); //Đặt thời gian hết hạn sau 1 tuần

Cache::put('key','value',$expiresAt);

Ngoài ra, ta có thể dùng phương thức add để lưu trữ dự liệu . Phương thức add sẽ trả về giá trị true nếu thêm thành công dữ liệu vào bộ nhớ cache còn không sẽ trả về false:

Cache::add('key','value',$minutes);

Để lưu trữ một giá trị trong bộ nhớ cache vĩnh viễn thì bạn có thể sử dụng phương thức forever:

Cache::forever('key','value');

e. Tăng và giảm các giá trị đối với các dữ liệu kiểu integer

Cache::increment('key'); // Tăng với giá trị =1

Cache::increment('key',$amount); // Tăng với giá trị = $amount

Cache::decrement('key'); // Giảm với giá trị =1

Cache::decrement('key',$amount); // Giảm với giá trị = $amount

f. Xóa dữ liệu trong bộ nhớ cache

* Xóa một dữ liệu

Cache::forget('key');

* Xóa toàn bộ tất cả các dữ liệu

Cache::flush();