



## LARAVEL FRAMEWORK







---

### BÀI 5: SỬ DỤNG ELOQUENT ORM PHẦN 1







- ① Tìm hiểu về thư viện Eloquent trong Laravel
- ① Sử dụng Eloquent để thao tác với database



## Phần I:

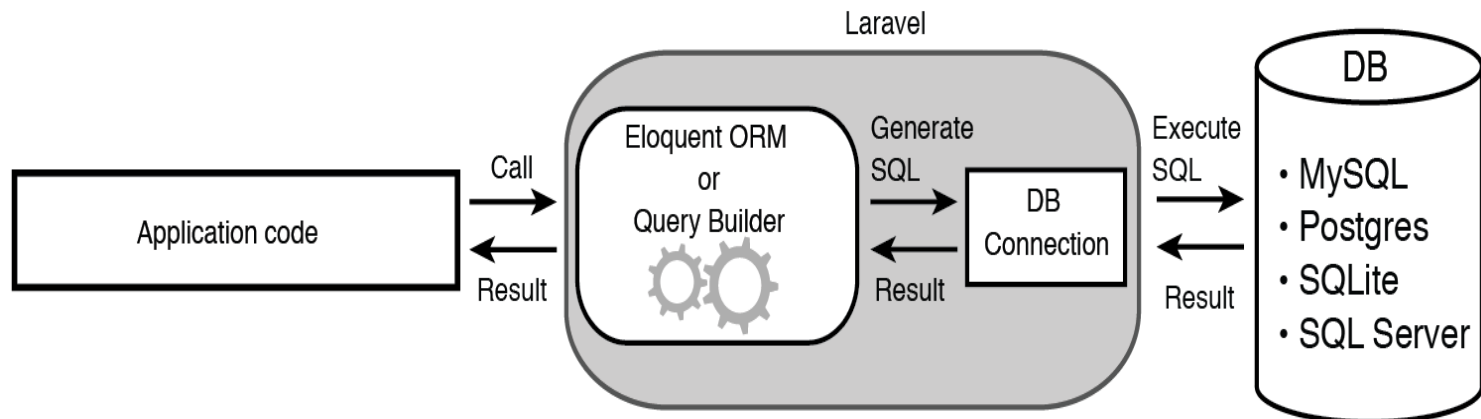
-  Giới thiệu Eloquent
-  Tạo model với Eloquent
-  Các mặc định trong eloquent model
-  Tương tác db với Eloquent model
-  Lấy data với Eloquent model
-  Lấy thông tin tổng kết

## Phần II:

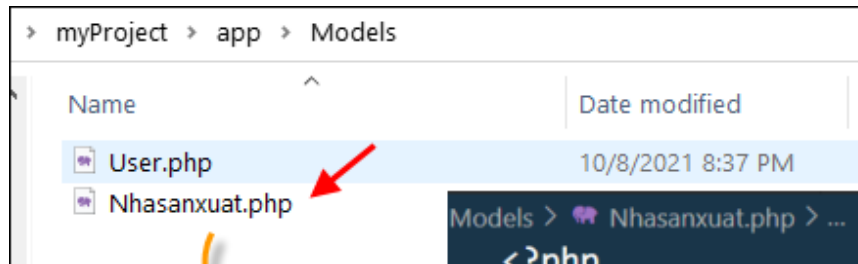
-  Chèn record với Eloquent model
-  Cập nhật record với Eloquent model
-  Xóa record với Eloquent model
-  Xóa mềm record với Eloquent model
-  Phục hồi record đã xóa mềm
-  Events trong Eloquent



- ❑ Eloquent ORM là công cụ trong Laravel cung cấp cách thức tương tác với database rất thuận lợi.
- ❑ Với Eloquent, mỗi table trong database sẽ có một model tương ứng. Model này giúp select, insert, update, delete dữ liệu trong table rất dễ dàng
- ❑ Model để tương tác với table do bạn tạo ra, rất đơn giản bằng công cụ artisan.
- ❑ Mỗi model là 1 file được lưu trong folder app/Models



- ❑ Để tạo model với Eloquent, sử dụng lệnh sau:  
**php artisan make:model Nhasanxuat**
- ❑ Vào folder app/Models → sẽ thấy model mới tạo



```
Models > Nhasanxuat.php > ...  
<?php  
namespace App\Models;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
class Nhasanxuat extends Model {  
    use HasFactory;  
}
```

Trong model, có thể khai báo 1 số biến đặc biệt sau:

- 1. \$table:** Là biến để khai báo tên table liên kết với model hiện tại.  
Mặc định mỗi model map với 1 table có tên giống model nhưng có thêm s hoặc ies.  
Ví dụ model **Product** sẽ map với table **products**, model **tin** map với table **tins**. Để bỏ mặc định thì dùng biến **\$table**

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Nhasanxuat extends Model {
    protected $table = 'nhasanxuat';
    public $primaryKey = 'maNSX';
    public $incrementing = false;
}
```

2. **\$primaryKey**: Là biến để cho biết field nào là khóa chính trong table.

Mặc định, model xem field có tên **id** là khóa chính. Nếu dùng tên khác thì khai báo trong **\$primaryKey**.

Ngoài ra nếu khóa chính không phải là số thì thiết lập **\$incrementing = false**.

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Nhasanxuat extends Model {
    protected $table = 'nhasanxuat';
    public $primaryKey = 'maNSX';
    public $incrementing = false;
}
```

**3. \$timestamps** : Là biến dùng để cho/cấm Eloquent ghi nhận thời điểm tạo/ cập nhật các record.

Nếu cho ghi (**\$timestamps = true**) thì table phải có 2 field **created\_at** , **updated\_at**.

Nếu không muốn ghi thì gán **\$timestamps = false**

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Nhasanxuat extends Model {
    protected $table = 'nhasanxuat';
    public $primaryKey = 'maNSX';
    public $incrementing = false;
    public $timestamps = false;
    protected $attributes = [
        'logo' => '',
        'status' => 0
    ];
}
```



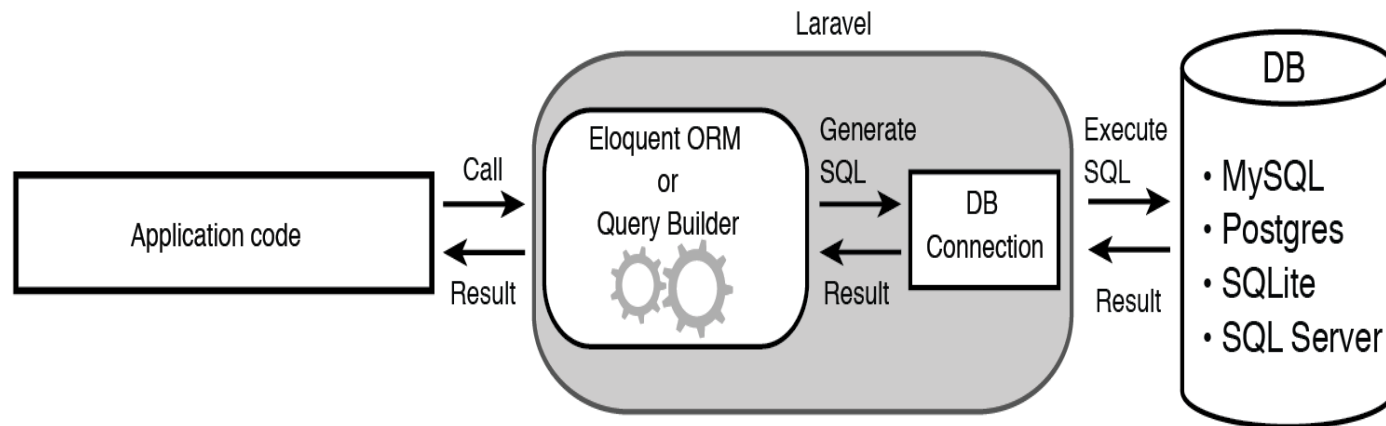
4. **\$attributes:** Là biến dùng để khai báo giá trị mặc định cho các field trong table , dùng khi insert, update

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Nhasanxuat extends Model {
    protected $table = 'nhasanxuat';
    public $primaryKey = 'maNSX';
    public $incrementing = false;
    public $timestamps = false;
    protected $attributes = [
        'logo' => '',
        'status' => 0
    ];
}
```

- 5. **\$fillable** : Là biến dùng để khai báo các field trong table
- 6. **\$dates**: là biến dùng để khai báo các field kiểu ngày trong table.

```
class tin extends Model {  
    protected $table='tin';  
    protected $primaryKey='id';  
    protected $dates = ['ngayDang'];  
    protected $fillable = [  
        'tieuDe',  
        'tomTat',  
        'urlHinh',  
        'ngayDang',  
        'noiDung',  
        'idLT'  
    ];  
}
```

- ❑ Khi đã tạo model và khai báo các thuộc tính cần thiết. Bạn đã có thể xem thêm sửa xóa dữ liệu trong table thông qua model rất dễ dàng.
- ❑ Các slide sau hướng dẫn cách lấy data, insert, update, delete record trong table thông qua Eloquent model



- ❑ Để lấy tất cả record từ table, dùng phương thức **all()**

```
$kq = App\Models\Nhasanxuat::all();  
foreach ($kq as $nsx) {  
    echo $nsx->ten, "<br>";  
}
```

- ❑ Để lấy các record theo điều kiện, dùng phương thức **get()** trong query buider

```
$kq = App\Models\Nhasanxuat::where('status', 1)-  
>orderBy('ten')->get();  
foreach ($kq as $nsx) echo $nsx->ten, "<br>";
```

- ❑ Để lấy 1 record từ table, dùng phương thức **find()**,

```
$kq = App\Models\Nhasanxuat::find(10); //Lấy record  
có id là 10  
echo $kq->ten;
```

- ❑ Để lấy các record đầu tiên thỏa điều kiện, dùng phương thức **first()** hoặc **firstWhere()**

```
$kq=App\Models\Nhasanxuat::where('status',1)->first();  
echo $kq->ten, "<br>";  
$kq = App\Models\Nhasanxuat::firstWhere('status',1);  
echo $kq->ten, "<br>";
```

- ❑ Có thể dùng các phương thức `sum()`, `max()`, `min()`, `count()` để có các thông tin tổng kết

```
$count=Nhasanxuat::where('ten','like','D%')->count();  
$max=Nhasanxuat::where('id','<','10')->max('status');  
echo $count, " ", $max;
```



DEMO

- Demo tạo modle, kết nối và lấy dữ liệu từ table





**FPT POLYTECHNIC**



## **LARAVEL FRAMEWORK**

### **BÀI 5 SỬ DỤNG ELOQUENT ORM PHẦN 2**

[www.poly.edu.vn](http://www.poly.edu.vn)



- ❑ Để tạo record mới trong table, bạn tạo instance cho model, gán giá trị cho các field rồi gọi phương thức save()

```
$nsx = new App\Models\Nhasanxuat;  
$nsx->ten = 'Compact';  
$nsx->logo = 'images/compact.jpg';  
$nsx->save();
```

- ❑ Cũng có thể dùng hàm create() để tạo record mới:

```
$kq = App\Models\Nhasanxuat::create([  
    'ten' => 'Gigabyte',  
    'status' => 1  
]);
```

- ❑ Để dùng hàm create(), bạn cần khai báo các field trong thuộc tính **\$fillable** của model như sau:

```
protected $fillable = [  
    'ten',  
    'logo',  
    'status',  
];
```

- ❑ Để tạo cập nhật 1 record trong table, bạn lấy record trong table với hàm find().
- ❑ Sau đó gán giá trị cho các field rồi gọi phương thức save()

```
$nsx = App\Models\Nhasanxuat::find(1);  
$nsx->ten = 'Realme';  
$nsx->save();
```

- ❑ Cũng có thể dùng hàm update() để cập nhật các record

```
App\Models\Nhasanxuat::where ('id',1)  
->update(['ten'=>'Oppo', 'status'=>2]);
```

- ❑ Để xóa 1 record trong table, gọi phương thức delete()

```
$nsx = App\Models\Nhasanxuat::find(1);  
$nsx ->delete();
```

- ❑ Có thể xóa nhanh record dùng hàm destroy và truyền vào các giá trị trong field khóa chính

```
App\Models\Nhasanxuat::destroy(1);  
App\Models\Nhasanxuat::destroy([3,4,5]);
```

- ❑ Có thể xóa các record thỏa điều kiện bằng phương thức delete()

```
App\Models\Nhasanxuat::where("status",0)->delete();
```

- ❑ Có thể dùng hàm truncate() để xóa tất cả record

```
App\Models\Nhasanxuat::truncate();
```

- ❑ Thay vì thực sự xóa các record khỏi database, Eloquent cũng cung cấp kiểu "soft delete" (xóa mềm) record. Tức là không thực sự xóa, chỉ đánh dấu xóa.
- ❑ Để cho phép xóa mềm, bạn cần:
  - ❖ Thêm vào table 1 field có tên **deleted\_at** kiểu là **timestamp**, cho phép **NULL** . Eloquent sẽ cập nhật field này khi record bị xóa mềm.
  - ❖ Ở đầu model, thêm lệnh:  
**use Illuminate\Database\Eloquent\SoftDeletes;**
  - ❖ Trong model, thêm lệnh **use SoftDeletes;**
- ❑ Lúc này, khi xóa record, cột deleted\_at sẽ được Eloquent cập nhật. Và khi bạn query, record sẽ tự động bị loại khỏi tất cả các kết quả.

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
class Nhasanxuat extends Model {
    protected $table = 'nhasanxuat';
    protected $attributes = ['logo' => '', 'status' => 0,];
    protected $fillable = ['ten', 'logo', 'status'];
    use SoftDeletes;
}
```

```
$nsx = App\Models\Nhasanxuat::find(6);
$nsx ->delete();
```

Máy phục vụ: 127.0.0.1 » Cơ sở dữ liệu: la\_news » Bảng: nhasanxuat

Duyệt	Cấu trúc	SQL	Tìm kiếm	Chèn	Xuất	
id	ten	logo	created_at	updated_at	status	deleted_at
6	Dell 4	NULL	2021-12-11 04:28:51	1	2021-12-11 04:28:51	
7	Dell 8	NULL	NULL	2	NULL	
8	Dell 8	NULL	NULL	1	NULL	

- ❑ Để khôi phục lại một record đã xóa mềm, sử dụng phương thức `restore()`
- ❑ Khi được `restore`, giá trị trong `deleted_at` sẽ mất, còn `updated_at` sẽ được cập nhật.

```
$nsx = App\Models\Nhasanxuat::find(7);  
$nsx ->delete();  
//...  
$nsx ->restore();
```

Máy phục vụ: 127.0.0.1 » Cơ sở dữ liệu: la_news » Bảng: nhasanxuat							
Duyệt   Cấu trúc   SQL   Tìm kiếm   Chèn   Xuất							
id	1	ten	logo	created_at	updated_at	status	deleted_at
6		Dell 4		NULL	2021-12-11 04:28:51	1	2021-12-11 04:28:51
7		Dell 8		NULL	2021-12-11 05:13:58	2	NULL
8		Dell 8		NULL	NULL	1	NULL

- ❑ Có thể restore nhiều record đã xóa :

```
App\Models\Nhasanxuat::withTrashed()  
->where('status',1)  
->restore();
```

- ❑ Để xóa vĩnh viễn một soft delete, sử dụng hàm **forceDelete**.  
Ví dụ xóa vĩnh viễn record có id là 7

```
App\Models\Nhasanxuat::withTrashed()  
->where('id',7)  
->forceDelete();
```

Eloquent model fire ra nhiều sự kiện, cho phép bạn đón và xử lý nếu cần:

- ☐ creating: trước khi create record.
- ☐ created: sau khi created record.
- ☐ updating: trước khi update record.
- ☐ updated: sau khi update record.
- ☐ deleting: trước khi xóa record.
- ☐ deleted: sau khi xóa record.
- ☐ retrieved: khi nhận dữ liệu từ database.



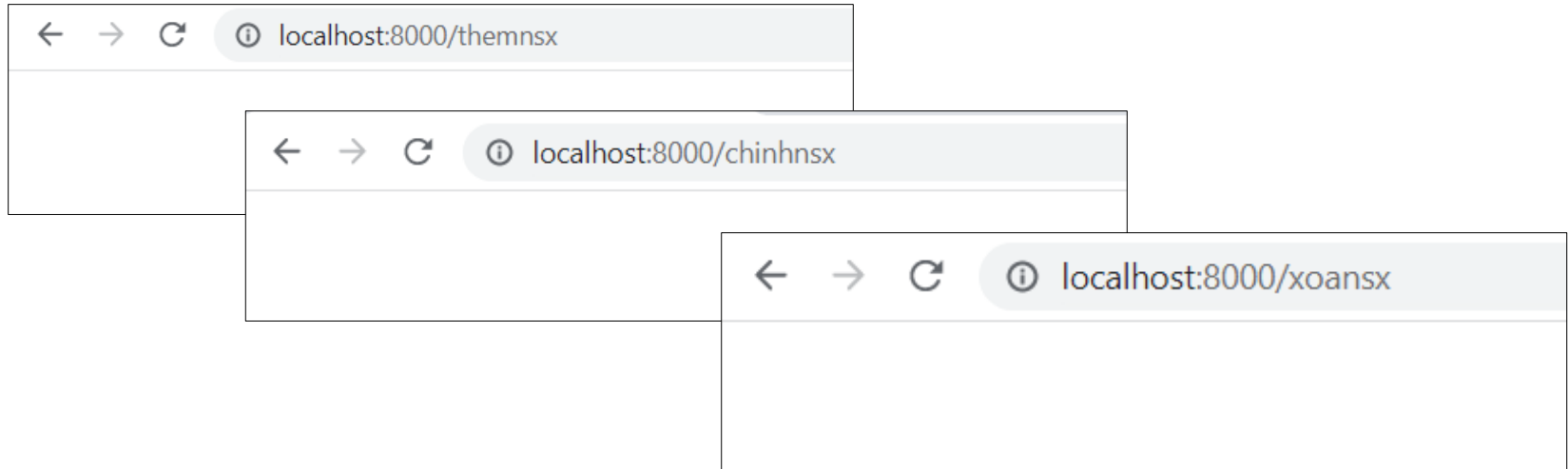
## 1. Trong model, định nghĩa hàm boot và các hàm đón sự kiện

```
<?php // app/Models/Nhaxsanxuat.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Log;
class Nhasanxuat extends Model {
    protected $table = "nhasanxuat";
    protected $attributes = ['logo' => '', 'status' => 0];
    protected $fillable = ['ten', 'logo', 'status'];
    public static function boot() {
        parent::boot();
        static::creating(function($item) {
            if ($item->status > 2) $item->status = 2;
        });
        static::created(function($item) {
            Log::info('Đã chèn nsx: '.$item);
        });
        static::updating(function($item) {
            if ($item->status < 0) $item->status = 0;
        });
        static::updated(function($item) {
            Log::info('Đã cập nhật nsx: '.$item);
        });
        static::deleted(function($item){
            Log::info('Đã xóa nsx: '.$item);
        });
    }
}
```

## 2. Gọi các hàm chèn, chỉnh, xóa record

```
<?php //routes/web.php
use Illuminate\Support\Facades\Route;
use App\Models\Nhasanxuat;
Route::get('/themnsx', function () {
    Nhasanxuat::create(['ten'=>'Gigabyte', 'status'=>10]);
});
Route::get('/chinhnsx', function () {
    Nhasanxuat::find(1)->update(['ten'=>'Intel', 'status'=>-5]);
});
Route::get('/xoansx', function () {
    Nhasanxuat::find(1)->delete();
});
```

### 3. Request và xem kết quả db + file storage/logs/Laravel.log



file storage/logs/Laravel.log

```
[2022-03-06 08:43:04] local.INFO: Đã chèn nsx:
{"logo":"","status":2,"ten":"Gigabyte","updated_at":"2022-03-06T08:43:04.000000Z","created_at":"2022-03-06T08:43:04.000000Z","id":63}
[2022-03-06 08:43:17] local.INFO: Đã cập nhật nsx:
{"id":10,"ten":"Intel","logo":"","status":0,"created_at":null,"updated_at":"2022-03-06T08:43:17.000000Z"}
[2022-03-06 08:43:45] local.INFO: Đã xóa nsx:
{"id":10,"ten":"Intel","logo":"","status":0,"created_at":null,"updated_at":"2022-03-06T08:43:17.000000Z"}
```









DEMO







-Demo CRUD



## Phần I:

-  Giới thiệu Eloquent
-  Tạo model với Eloquent
-  Các mặc định trong eloquent model
-  Tương tác db với Eloquent model
-  Lấy data với Eloquent model
-  Lấy thông tin tổng kết

## Phần II:

-  Chèn record với Eloquent model
-  Cập nhật record với Eloquent model
-  Xóa record với Eloquent model
-  Xóa mềm record với Eloquent model
-  Phục hồi record đã xóa mềm
-  Events trong Eloquent



 <https://laravel.com/docs/eloquent>

 <https://eloquentbyexample.com/>

 <https://stackify.com/laravel-eloquent-tutorial/>



**Cảm ơn**