**Bài 10: Query Builder trong Laravel**

  -Trước đây khi làm việc với PHP thuần, để tương tác với database các bạn sẽ cần phải kết nối với database và viết những câu truy vấn hết sức là phức tạp để gọi được dữ liệu ra. Nhưng khi sử dụng các famework đặc biệt là Laravel thì các bạn sẽ không phải lo về điều đó nữa, bởi Laravel có cung cấp cho chúng ta một chức năng rất hay là **Query Builder.**

**1, Cấu hình database.**

-Trước tiên để sử dụng được Query builder các bạn cần phải cấu hình database kết nối đến cơ sở dữ liệu.

-Để cấu hình Database các bạn cần mở file **.env**lên và cấu hình như sau.

APP\_ENV=local

APP\_KEY=base64:w6dZccXCvdehAax2NYUt6f/igtsyw7QTQCx2eGEQBgI=

APP\_DEBUG=true

APP\_LOG\_LEVEL=debug

APP\_URL=http://localhost

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_USERNAME=root

DB\_PASSWORD=

BROADCAST\_DRIVER=log

CACHE\_DRIVER=file

SESSION\_DRIVER=file

QUEUE\_DRIVER=sync

REDIS\_HOST=127.0.0.1

REDIS\_PASSWORD=null

REDIS\_PORT=6379

MAIL\_DRIVER=smtp

MAIL\_HOST=mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=null

MAIL\_PASSWORD=null

MAIL\_ENCRYPTION=null

PUSHER\_APP\_ID=

PUSHER\_KEY=

PUSHER\_SECRET=

-Chú thích:

* DB\_HOST : Là server chứa mysql của bạn.
* DB\_PORT: Là cổng post của server Mysql của bạn (thường để mặc định là 3306).
* DB\_DATABASE: Là tên database của bạn.
* DB\_USERNAME: Là username để truy cập vào database của bạn.
* DB\_PASSWORD: Là password để truy cập vào database của bạn.

**2, Sử dụng Query Builder.**

-Sau khi đã cấu hình được database (ở phần trên). Thì giờ mọi người bắt đầu thực hành truy vấn Sql với Query Builder nhé.

**Lấy tất cả dữ liệu trong bảng.**

-Cú Pháp:

DB::table('tablename')->get()

**VD:** Lấy tất cả dữ liệu trong bảng tbl\_product.

Route::get('/', function () {

$data=DB::table('tb\_product')->get();

print\_r($data);

});

**Kết quả:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | Illuminate\Support\Collection Object | |  | ( | |  | [items:protected] => Array | |  | ( | |  | [0] => stdClass Object | |  | ( | |  | [id] => 5 | |  | [title] => Váy đầm xòe Asose | |  | ) | |  |  | |  | [1] => stdClass Object | |  | ( | |  | [id] => 6 | |  | [title] => Váy đầm xòe Asose | |  | ) | |  |  | |  | ) | |  |  | |  | ) | |  |  | |

**Select cột trong bảng.**

-Cú pháp:

DB::table('tablename')->select('columnfirst','columnsecond')->get();

**Select có điều kiện.**

-Cú pháp:

+Điều kiện bằng:

DB::table('tablename')->where('column','filter')->get();

+Điều kiện lớn hơn:

DB::table('tablename')->where('column','>','filter')->get();

+Điều kiện nhỏ hơn:

DB::table('tablename')->where('column','<','filter')->get();

+Điều kiện khác:

DB::table('tablename')->where('column','<>','filter')->get();

+Điều kiện lồng:

VD:

$users = DB::table('users')->where('votes', '>', 100)->orWhere('name', 'John')->get();

Truy vấn tìm kiếm.

DB::table('tablename')->where('column','like','filter')->get();

// vd : DB::table('product')->where('name','like','%Sách%')->get();

**Join bảng.**

$users = DB::table('users')->join('contacts', 'users.id', '=', 'contacts.user\_id')->get();

+Left join:

$users = DB::table('users')->leftjoin('contacts', 'users.id', '=', 'contacts.user\_id')->get();

**Unions.**

$first = DB::table('users')

->whereNull('first\_name');

$users = DB::table('users')

->whereNull('last\_name')

->union($first)

->get();

**OrderBy.**

$users = DB::table('users')

->orderBy('name', 'desc')

->get();

**Random.**

$randomUser = DB::table('users')

->inRandomOrder()

->first();

**GroupBy/having.**

$users = DB::table('users')

->groupBy('account\_id')

->having('account\_id', '>', 100)

->get();

**Thêm (insert).**

DB::table('users')->insert(

['email' => 'john@example.com', 'votes' => 0]

);

**Sửa (update).**

DB::table('users')

->where('id', 1)

->update(['votes' => 1]);

**Xóa (delete).**

DB::table('users')->delete();

//hoặc

DB::table('users')->where('votes', '>', 100)->delete();

**3, Chú ý.**

-Để sử dụng được các câu truy vấn trên bắt buộc các bạn phải:

* Kết nối với cơ sở dữ liệu.
* Nếu truy vấn trong controllers thì các bạn cần phải khai báo use Illuminate\Support\Facades\DB; còn trong Route thì không cần.