**Bài 15: Schema Buider trong Laravel**

Ở các bài trước mình đã giới thiệu với mọi người cách truy xuất dữ liệu trong Laravel Với [Query Buider](http://toidicode.com/query-builder-trong-laravel-11.html) và [Eloquent ORM](http://toidicode.com/eloquent-orm-trong-laravel-13.html) rồi. Hôm nay tiếp tục với series [Học Laravel](http://toidicode.com/Laravel-14) của Toidicode.com mình xin được giới thiệu với mọi người một thuật ngữ mới nữa trong Laravel đó là **Schema Builder.**

**1, Schema Builder trong Laravel là gì?**

-Schema Builder Là một class trong bộ Facades của Laravel nó giúp chúng ta làm việc với tất cả các cơ sở dữ liệu mà Laravel hỗ trợ với các hàm được định nghĩa sẵn.

-Thường thì nó sẽ được kết hợp với Migrations để xây dự cấu trúc database.

-Điều kiện để sử dụng Schema Builder:

* Phải có kết nối với database .
* Muốn sử dụng được Schema Builder thì phải gọi namespace của nó **Illuminate\Support\Facades\Schema**

**2, Các lệnh thường dùng trong Schema Builder.**

**Tạo bảng.**

Cú pháp:

Schema::table('tablename', function ($table) {

//code

});

Vd: Tạo bảng users.

Schema::table('users', function ($table) {

//code

});

**Đổi tên bảng.**

Cú Pháp:

Schema::rename($from, $to);

VD: Đổi bảng users thành admin.

Schema::rename('users','admin');

**Xóa bảng.**

Cú Pháp:

Schema::drop('tableName');

//hoac

Schema::dropIfExists('tableName');

VD: Xóa bảng users.

Schema::drop('users');

//hoặc

Schema::dropIfExists('users');

**Thêm cột.**

Cú Pháp:

Schema::table('tableName', function ($table) {

$table->typeData('columnName');

});

VD: Thêm cột email với kiểu dữ liệu vachar.

Schema::table('users', function ($table) {

$table->string('email');

});

->Chú ý: Các kiểu dữ liệu mình sẽ nói rõ hơn ở [bài sau](http://toidicode.com/bai-16-migrations-trong-laravel/).

**Đổi tên cột.**

Schema: table('tableName', function ($table) {

$table->renameColumn('from', 'to');

});

VD: Đổi cột email trong bảng users thành usermail.

Schema::table('users', function ($table) {

$table->renameColumn('email', 'usermail');

});

**Xóa cột.**

Cú Pháp:

Schema::table('tableName', function ($table) {

$table->dropColumn('columnName');

//hoặc xóa nhiều cột cùng lúc.

$table->dropColumn(['columnName1','columnName2','columnName3']);

});

VD: xóa cột email trong bảng users.

Schema::table('users', function ($table) {

$table->dropColumn('email');

});

**Tạo index.**

Cú Pháp:

$table->index('coumnName');

Các kiểu Index.

| **Cú pháp** | **Mô tả** |
| --- | --- |
| table->primary('id'); | Add a primary key. |
| table->primary(['first', 'last']); | Add composite keys. |
| $table->unique('email'); | Add a unique index. |
| $table->unique('state', 'my\_index\_name'); | Add a custom index name. |
| $table->unique(['first', 'last']); | Add a composite unique index. |
| $table->index('state'); | Add a basic index. |

-Mấy cái này dịch sang tiếng việt rất khó biểu diễn nên mình để tiếng anh cho dễ :D.

-VD: mình sẽ đánh unique index cho cột email.

$table->unique('email');

**Xóa Index.**

Cú Pháp:

| **Cú Pháp** | **Mô Tả** |
| --- | --- |
| $table->dropPrimary('users\_id\_primary'); | Drop a primary key from the "users" table. |
| $table->dropUnique('users\_email\_unique'); | Drop a unique index from the "users" table. |
| $table->dropIndex('geo\_state\_index'); | Drop a basic index from the "geo" table. |

**Thêm khóa ngoại.**

Cú Pháp:

Schema::table('tableName', function ($table) {

$table->integer('conmName')->unsigned();

$table->foreign('Conmumname')->references('columnName')->on('tableName2');

});

Vd: Mình sẽ thêm cột users\_id của bảng posts làm khóa ngoại trên bảng users.

Schema::table('posts', function ($table) {

$table->integer('user\_id')->unsigned();

$table->foreign('user\_id')->references('id')->on('users');

});

Hoặc mình muốn khi xóa mội users thì đồng thời sẽ xóa hết các bài đăng của users đó đi thì .

$table->foreign('user\_id')

->references('id')->on('users')

->onDelete('cascade');

**Xóa khóa ngoại.**

Cú Pháp

$table->dropForeign('columnName');

VD: Mình xóa khóa ngoại ở ví dụ trên đi.

$table->dropForeign('user\_id');