لفرض أن لدينا الجداول التالية

Articles

Videos

Books

ولكل جدول من هذه الجداول يكون لها مجموعة من التعليقات.

لتخزين التعليقات في قاعدة البيانات يلجئ البعض لإنشاء جدول للتعليقات لكل نوع مثلا ينشئ الجداول التالية

Article\_comments

Id

comment

Article\_id

Timestamps

video\_comments

Id

comment

video\_id

Timestamps

book\_comments

Id

comment

book\_id

Timestamps

كما نلاحظ أن جميع الحقول متشابهه بإستثناء حقل book\_id, video\_id, article\_id.

إلا أن لارافيل وفرت لنا علاقة وهي Polymorphic Relationship حيث يتم إنشاء جدول واحد لجميع الجداول ويتم به تخزين الصور لجميع الحقول الأخرى (articles, projects, users).

يكون شكل الجدول على الشكل التالي

comments

Id

Filename

Photoable\_id

Photoable\_type

Timestamps

كيفية إنشاء جدول photos

كما نعلم جميعنا يمكن ذلك من خلال إستخدام الأمر

php artisan make:model Comment -m

ويجب أن يحتوي على الحقول التالي

public function up()

{

Schema::create(comments, function (Blueprint $table) {

$table->id();

$table->string('filename');

// $table->unsignedBigInteger('photoable\_id');

// $table->string('photoable\_type');

$table->morphs('photoable');

$table->timestamps();

});

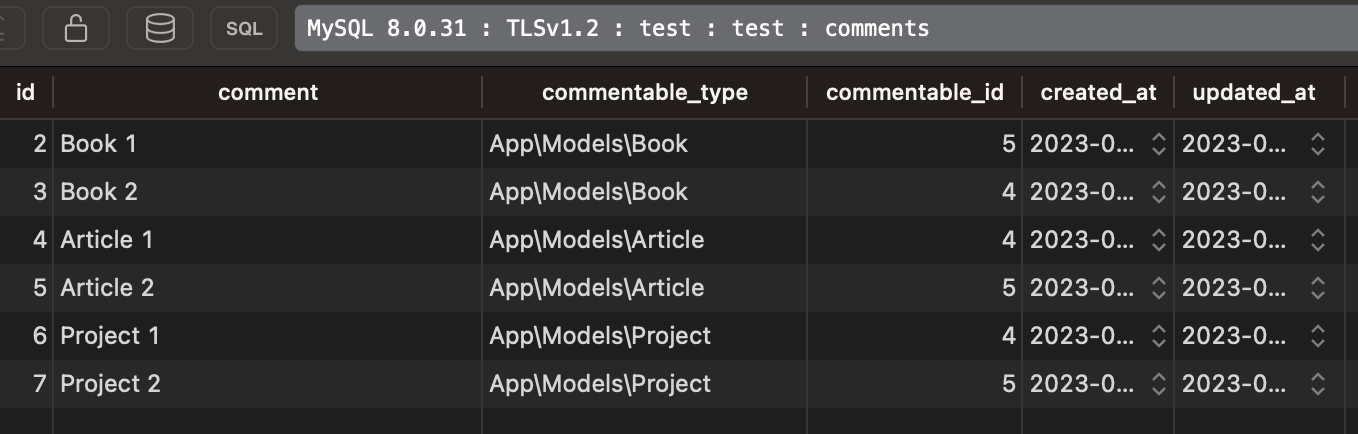
}

كما نلاحظ أنه يوجد نوع جديد من البيانات وهي morphs، حيث تقوم بإنشاء حقلين وهما commentable\_id وكذلك commentable\_type في جدول comments.

حقل commentable\_type يجب تسميته هكذا اذا كان إسم الجدول comments، أما إذا كان إسم الجدول photos فإنه يجب أن يكون photoable\_type وهكذا، بمعنى أنه يجب وضع إسم الجدول singular ومن ثم إضافة able.

في حقل commentable\_id يتم تخزين id الجدول التابع له التعليق.

في حقل commentable\_type يتم تخزين مسار الـ model التابعة له التعليق، كما في الصوره التالية



توضيح

نلاحظ أن التعليق Book 1, Book 2 أنهما يتبعان للموديل Book أي الحقل books وعرفنا ذلك من خلال commentable\_type.

نلاحظ أيضا أن commentable\_id قد يتكرر، لأن التعليق على id الكتاب رقم 5 ، وكذلك تكرر commentable\_id يحمل رقم 5 في التعليق Article 1 وكذلك Project 2 ، إلا إنه كما قلنا يتم التمييز لأي Model تتبع من خلال Commentable\_type.

الأن بعد أن فهمنا عملية migration وهيكلية الجدول comments نأتي لإنشاء العلاقات.

إنشاء العلاقات في Polymorphic relations

في الموديل Comment نضيف الدالة commentable وقد شرحنا بالأعلى لماذا يجب تسميتها بهذا الشكل

public function commentable()

{

return $this->morphTo();

}

وكما نلاحظ تأخذ نوع العلاقة morphTo.

في الموديل الأخرى وهي Book, Article, Project يجب أن نضيف الدالة comments وبداخلها نحدد العلاقة morphMany ومن ثم تحديد إسم الموديل وفي هذه الحالة هو Comment وكذلك تحديد إسم العلاقة في الموديل Comment وفي هذه الحالة هي commentable

public function comments(): MorphMany

{

return $this->morphMany(Comments::class, 'commentable');

}

كما قلنا يجب أن نضع العلاقة في جميل الموديل (Book, Article, Projects)

إضافة التعليقات

في إضافة وتخزين التعليقات يجب إستخدام العلاقة

مثلا إذا كنا نريد إضافة تعليق على كتاب معين

public function addComment($book\_id, Request $request)

{

$book=Book::findOrFail($book\_id);

$book->comments()->create([

'comment'=>$request->comment

]);

}

مع الإنتبها إنه يجب إضافة comment كـ fileable في comment model

protected $fillable = ['name'];

وكذلك إن أردنا إضافة تعليق على Article

public function addComment($article\_id, Request $request)

{

$article=Article::findOrFail($article\_id);

$article->comments()->create([

'comment'=>$request->comment

]);

}

عرض التعليقات

لعرض التعليقات لمقال معين، نستخدم with ومن ثم إسم العلاقة وهي comments في حالتنا هذه

$article=Article::with('comments')->where('id',$article->id)->first();