

Laravel

Laravel, ese framework que nos tiene enamorado.

- [Specified key was too long](#)
- [Consejos y trucos de Laravel. Sólo enlaces](#)
- [Cómo recuperar registros entre dos marcas de tiempo en Laravel con Eloquent](#)
- [Crear nuevo índice compuesto único, en una tabla con duplicados MySQL y Laravel Eloquent](#)
- [Comprobar la configuración de correo en Laravel](#)
- [Test de excepciones en Laravel con PestPHP](#)

Specified key was too long

Escenario

Despliegue de una app de laravel en un servidor. Ejecutamos el primer migrate y zas.

```
php artisan migrate
```

```
...
```

```
Illuminate\Database\QueryException : SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key  
was too long; max key length is 1000 bytes
```

```
...
```

No nos asustemos; el tema está en una diferencia de versión MySQL

“ ATENCIÓN a programadores cada vez las diferencias entre versiones de un mismo motor son fuertes (MySQL 5.X vs MySQL 8, por ejemplo), MySQL vs MariaDB vs Percona. Algunas pueden producir serios problemas en caso de tener que recuperar un backup de otro motor o versión, o una configuración.

Posibles soluciones

Actualización del motor de Base de Datos

Si es posible lo adecuado, es que uses el mismo motor tanto en desarrollo, testing y producción. Y la misma configuración.

“ Tener apps profesionales, en un entorno compartido, es como ser un feriante (con respeto a su profesión). Sufriras, tu tienda o negocio no tendrá el aspecto que tendrías en un [servidor dedicado o VPS](#), optimizado para tu negocio

Modificación del engine en tu app

Si te es imposible o no quieres actualizar tu monotr de bases de datos, intenta modificar el engine de mysql en el archivo `config/database.php`

```
'mysql' => [  
    'driver' => 'mysql',  
    ...  
    'engine' => 'InnoDB',  
    ...  
],
```

Modificación del AppServiceProvider

Si no te funciona la anterior, puede agregar una linea en tu fichero

`app/Providers/AppServiceProvider.php`

```
...  
use Illuminate\Support\ServiceProvider;  
use Illuminate\Support\Facades\Schema;  
...  
  
public function boot()  
{  
    Schema::defaultStringLength(191);  
}
```

Explicación

La naturaleza del problema es debido a que por defecto Laravel usa como codificación **utf8mb4**, con soporte para los emojis.

Enlaces y cosas sobre el tema

- [mysql 8 ERROR 1231 \(42000\) at line Variable 'time_zone'](#)
- [Mysql restore con error 1217 y error 1292](#)

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Consejos y trucos de Laravel. Sólo enlaces

Laravel

Muchas partes del ecosistema Laravel no se ven a simple vista en su documentación, y otras tantas sus avances son tan rápidos que uno no tiene tiempo de asimilar todo o conocer todo.

Así que dejo aquí enlaces que me parecen muy interesantes

| Enlace | Keys |
|---|------------------------------|
| 30 Laravel Eloquent Tips & Tricks | eloquent, attributes, push() |
| Laravel Lang Publisher | laravel, lang, publisher |

Cómo recuperar registros entre dos marcas de tiempo en Laravel con Eloquent

Tip

Use `whereBetween` con una fecha desde y hasta. Si necesita proporcionar un respaldo, puede hacerlo con el operador coalescente nulo `??`

```
$invoice = Invoice::whereBetween('invoiced_at', [  
    $request->since ?? '2020-01-01',  
    $request->until ?? today()->toDateTimeString()  
)
```

Agradecimientos

Gracias a [LaravelEloquent](#)

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Crear nuevo índice compuesto único, en una tabla con duplicados MySQL y Laravel Eloquent

Introducción

Una de las cosas que mas enamorado me tiene, son las [migraciones de Laravel](#) ya que te permite trabajar con software en producción, siendo un poco cuidadoso, por supuesto.

En el caso de este artículo, se trataba de añadir un índice compuesto, sobre una o varias tablas, que contenían cientos de miles de filas, y que por necesidad y eficacia, deberían haber tenido este índice compuesto único.

Que decir tiene, que la razón del mismo era, que al no tenerlo se habían producido duplicados no deseados, que fueron detectados en un bug abierto.

Procedimiento

- Crear un branch del proyecto
- **Crear un backup de mysql** del proyecto para trabajar con los datos en local.
- Generar la migración de actualización.
- Ejecutarla para ver el resultado.

Migración fallida

En el escenario normal escribimos una migración para añadir un índice compuesto en nuestra app Laravel

```
$ php artisan make:migration add_index_modem_plus_job_modems_table --table="logger_modems"
```

Donde escribiremos el índice compuesto que queremos añadir.

```
Schema::table('logger_modems', function (Blueprint $table) {  
    $table->unique(['command_center_id', 'job']);  
});
```

Pero al ejecutarlo ya sabemos lo que ocurrirá.

```
Migrating: 2022_04_23_124738_add_index_modem_plus_job_modems_table
```

```
Illuminate\Database\QueryException
```

```
SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry '8-2022-04-21 02:00:01' for key  
'logger_modems.logger_modems_command_center_id_job_unique' (SQL: alter table `logger_modems` add  
unique `logger_modems_command_center_id_job_unique` (`command_center_id`, `job`))
```

Comprobación del problema

En Mysql (shell) o en nuestra app preferida (MySQLWorkbench, TablePlus,...) comprobamos el problema

```
SELECT command_center_id, COUNT(command_center_id) job, COUNT(job)  
FROM sitelight.logger_analyzers  
GROUP BY command_center_id, job HAVING COUNT(*) > 1;
```

MySQLWorkbench Duplicados

Modificación de la migración

La secuencia en formato SQL sería la de abajo, pero hemos dicho que queremos trabajar con eloquent, por muchos motivos.

```
CREATE TABLE tmp_data SELECT * FROM mytable;  
TRUNCATE TABLE mytable;  
ALTER TABLE mytable ADD UNIQUE INDEX myindex (A, B, C, D);  
INSERT IGNORE INTO mytable SELECT * from tmp_data;  
DROP TABLE tmp_data;
```


Así que aquí abajo lo adaptamos

```
public function up()
{
    DB::statement('CREATE TABLE sitelight.tmp_data SELECT * FROM sitelight.logger_modems');
    LoggerModem::query()->delete(); // TRUNCATE via eloquent
    Schema::table('logger_modems', function (Blueprint $table) {
        $table->unique(['command_center_id', 'job']);
    });
    DB::statement('INSERT IGNORE INTO sitelight.logger_modems SELECT * FROM sitelight.tmp_data');
    DB::statement('DROP TABLE sitelight.tmp_data');
}
```

Agradecimientos

- [Removing duplicates with unique index](#)
- [Find and remove duplicate rows by two columns](#)

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Comprobar la configuración de correo en Laravel

Configuración del correo en Laravel

A veces, algo falla, y es necesario probar que el problema no esta en la configuración del correo en nuestra aplicación Laravel.

Un metodo sencillo y eficaz es probarlo en el **tinker**

```
> php artisan tinker
```

```
Psy Shell v0.12.3 (PHP 8.3.6 — cli) by Justin Hileman
```

```
> Mail::raw('Hello World!', function($msg) {$msg->to('emails@mail.com')->subject('Test Email'); });
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Test de excepciones en Laravel con PestPHP

Introducción

El cambio de PHPUnit a [PestPHP](#) es algo progresivo.

Muchas veces se nos pasa que hay otros caminos de hacer las cosas. Este es uno de ellos, los **test de excepciones**

Test de Excepción: PHPUnit vs PestPHP

Con Pest es mas limpio hacer:

```
expect(fn () => CommandCenterProblem::factory()->create([
    'command_center_id' => $commandCenter->id,
    'trigger_id' => $trigger->id,
    'created_at' => $now,
]))->toThrow(UniqueConstraintViolationException::class);
```

aunque también podemos hacerlo a la vieja usanza.

```
$this->expectException(UniqueConstraintViolationException::class);

CommandCenterProblem::factory()->create([
    'command_center_id' => $commandCenter->id,
    'trigger_id' => $trigger->id,
    'created_at' => $now,
]);
```

Como siempre, en programación hay más de un camino, perfectamente validos. Son gustos o preferencias personales.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).