

### Artisan

```
php artisan serv //lance un serveur de test
php artisan down //Met l'application en maintenance
php artisan up //enleve l'application de la maintenance
php artisan env //affiche les variables d'environnement
php artisan help //affiche l'aide de la commande
php artisan list //list les commandes
php artisan optimize //optimise l'application
php artisan clear-compiled //efface du cache les fichiers compilés
php artisan tinker // lance l'application en mode console
```

### Make

```
php artisan make:controller MonController
php artisan make:command MaCommande
php artisan make:event Monevenement
php artisan make:job MonJob
php artisan make:listener MonListener
php artisan make:middleware MonMiddleWare
php artisan make:migration MaMigration
php artisan make:model MonModel
php artisan make:provider MonProvider
php artisan make:request MaRequest
php artisan make:seeder MonSeeder
```

### Blade Layout

```
//if condition
@if(condition)
@else
@endif
//loop
@foreach($foo as $bar)
@endforeach
//inclusion
// Ressources /views /folder/page.blade.php
@include('folder.page')
//Template
```

### Blade Layout (cont)

```
> @yield('section')
@extends('template')
@section('section')
@endsection
```

### Eloquent

```
//Model en base de donnees
class Post extend Model{
    protected $primaryKey = " IDP ost "
    protected $table = " pos t"
}
//Requerection par cle primaire
Post::find(1);
//Requerection par where
Post::where('titre', '=', 'valeur') ->get();
//Faire un join
Post::join('table', 'post.idTable',
'table.IDTable' )-> get();
//Insertion en base
$post = new Post();
$post->titre = " hel lo";
$post->save();
//Update
$post = Post::find(1);
$post->titre = " mod if"
$post->save();
//suppression
$post->delete();
```

### Route

```
php artisan route:cache //Met les routes en cache
php artisan route:clear //Supprime le cache des routes
php artisan route:list //affiche la liste des routes
```



By **nargil**  
[cheatography.com/nargil/](http://cheatography.com/nargil/)

Not published yet.  
 Last updated 13th November, 2017.  
 Page 1 of 3.

Sponsored by **CrosswordCheats.com**  
 Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

### Route

```
//Route Simple
Route::get('/', function () {
    return view('hello');
});

//Route avec parametre
Route::get('/post/{id}', function ($id) {
    return view('post', ["id" => $id]);
});

//Route avec controller et methode
Route::get('/add', 'MusiqueController@formAdd');

//Route nommé
Route::get('/add', 'MusiqueController@formAdd' -> name('showAddForm'));

//Route validation regex
Route::get('/post/{id}', "MusiqueController@oller@detail " -> where('id', '[0-9]+'));

//Methode http
Route::get('/', "controller@example");
Route::post('/', "controller@example");
Route::put('/', "controller@example");
Route::patch('/', "controller@example");
Route::delete('/', "controller@example");

//groupe
Route::prefix('api', function () {
    //http :fo.o.c om/api
    Route::get('/', function () {
        return view('api');
    });

    //http :fo.o.c om/ api /test
    Route::get('/test', function () {
        return view('api');
    });
});

//utilisation d'un middleware
//sur plusieurs routes
Route::middleware(['first', 'second']) -> -
group(function () {
```

### Route (cont)

```
> Route::get('/', function () {
    // utilise first & second Middleware
});
Route::get('user/profile', function () {
    // utilise first & second Middleware
});
});
```

### Seeder

```
php artisan migrate --seed
```

### Seeder

```
class DatabaseSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        //seed
    }
}
```

### Migration

```
php artisan migrate
```

```
php artisan migrate:install
```

```
php artisan migrate:rollback
```

```
php artisan migrate:refresh
```

```
php artisan migrate:reset
```

```
php artisan migrate:status
```



### Migration

```
class CreateMusiquesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('musiques',
function (Blueprint $table) {
            $table->increments('id');
            $table->integer('order');
            $table->string('titre',64);
            $table->string('description',256)->nullable();
            $table->string('url',256);
            $table->date('publication');
            $table->integer('categorie_id')->unsigned();
            $table->foreign('categorie_id')->references('id')->on('categories');
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('musiques');
    }
}
```

