Artisan

```
php artisan routes
php artisan controller:make UserController
// Migrations
php artisan migrate:make create_users_table
php artisan migrate:make create_users_table --create=users
php artisan migrate
php artisan migrate:rollback
php artisan migrate:refresh
// Seed
php artisan generate:seed posts
php artisan db:seed
php artisan migrate:refresh --seed
php artisan db:seed --class=PostsTableSeeder
// Generators
php artisan generate:resource post --fields="title:string, body:text"
php artisan generate:pivot categories users
```

Migrations

```
...
public function up(){
Schema::create('users', function(Blueprint $table){
$table->increments('id');
$table->integer('role');
$table->string('email')->unique();
$table->string('password', 60);
$table->rememberToken();
$table->timestamps;
});
}
public function down(){
Schema::drop('users'):
}
...
```

Seeds (faker)

```
...
User::create([
'email' => $faker->email(),
'password' => $faker-> md5()
]);
...
```

Routes

Ruta simple

```
Route::get('/',function(){
return View::make('hello');
});
```

Ruta amb paràmetres

```
Route::get('posts/{id}',function($id){
return View::make('post.single')->with('id', $id);
});
```

Ruta Controlador + mètode

```
Route::get('post', 'PostController@show');
```

Ruta nominal

```
Route::get('post/all', array('uses' => 'PostController@all', 'as' =>
'post.all'));
```

Ruta + validació RegEX

```
Route:get('post/{id}', array('uses' => 'PostController@single', 'as' =>
'get.post.single'))->where('id', '[1-9][0-9]*');
```

Ruta POST

```
Route::post('post', array('uses' => 'PostController@create', 'as' =>
'post.post.create'));
```

Ruta Resource

```
Route::resource('post', 'PostController');
Route::resource('post', 'PostController', array('except' => 'show'));
Route::resource('post', 'PostController', array('only' => 'show'));
```

Filtres

```
Route::get('post/create', array('uses' => 'PostController@create', 'as'
=> 'post.create', 'before' => 'auth'));
```

Grups

```
Route::group(array('before' => 'auth'), function(){
// Route:: ...
// Route:: ...
```


Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

Routes (cont)

```
});
```

Prefixs

```
Route::group(array('prefix' => 'admin'), function(){
// Route:: ...
// Route:: ...
});
```

Blade functions

```
@if(count($posts))
@foreach($posts as $post)
<p>{{{ $post->title }}} </p>
@endforeach
@endif
```

Blade Layout

```
<!-- HTML -->
@include('partials.menu');
[...]
@yield('content');
[...]
@section('sidebar');
[...]
@show
```

Blade Template

```
@extends('layouts.default');
@section('content');
[...]
@stop
@section('sidebar')
@parent
[...]
@stop
```

Query Builder

```
// SELECT
$users = DB::table('users')->get();
$users = DB::table('users')->find(2);
$users = DB::table('users')->where('id',2)->get();
$users = DB::table('users')->where(array('id' => 2, 'email' => 'test@-
test.com'))->get();
$users = DB::table('users')->where('id',2)->orWhere('id', 3)->get();
$users = DB::table('users')->where(array('id' => 2, 'email' => 'test@-
test.com'))->get();
$users = DB::table('users')->where('id', '>', 1)->orderBy('id', 'asc')->t-
ake(2)->skip(2)->get();
$users = DB::table('users')->join('posts', 'users.id', '=', 'posts.user_-
id')->get();
// Log
dd(DB::getQueryLog());
// INSERT
$data = array(
'email' => 'bernat.torras@uvic.cat',
'password' => '123456'
);
DB::table('users')->insert($data);
// UPDATE
$data = array(
'email' => 'bernat.torras@uvic.cat',
'password' => 'abc'
);
DB::table('users')->where('email', $data['email'])->update($data);
// DELETE
DB::table('users')->where('email', 'bernat.torras@uvic.cat')->delete();
```

Eloquent ORM

```
// SELECT
$posts = Post::all();
$posts = Post::find(2);
$posts = Post::where('title', 'LIKE', '%et%')->get();
$posts = Post::where('title', 'LIKE', '%et%')->take(1)->skip(1)->get();
// INSERT
$post = new Post;
$post->title = 'post1 title';
$post->body = 'post1 body';
```


Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

Eloquent ORM (cont)

```
$post->save();
// Insert amb vector de dades
$data = array(
'title' => 'post2 title',
'body' => 'post2 body'
);
Post::create($data);
// UPDATE
$post = Post::find(1);
$post->title('updated title');
$post->save();
// DELETE
$post = Post::find(1);
$post->delete();
```

Relacions BDD (Model)

```
class Post extends \Eloquent {
...
public function user(){
return $this->belongsTo('User');
// hasMany
// hasOne
// belongsToMany
}
...
}
```


Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com