Migration :操作数据库

1） 目录：database/migrate/

2) $ php artisan migrate

// $ php artisan migrate:rollback

3)自己建表：

$ php artisan make:migration crate\_...\_table

表单验证

$ php artisan make:request CreateUserRequest

Seeding

Eloquent ORM

1)~4) 在Model文件中声明

1) create A model (App\)

$ php artisan make:model <ModelName:User> [-m]//-m用来生存database migration

2) 一般情况下以类名的复数作为表名（如User类，表为Users）//定义的一种规范

自己声明表名 ： protected $table = '<tableName>';

3) 默认每个表都有主键,名为id

自定义： protected $primaryKey = '<keyName>';

4) create\_at updated\_at都是自动生成，如果不希望存在

public $timesatmps = false;

5) 创建Model后，可如下使用：(在controller中使用)

User::all();

User::where()->...()->get();

User::find($id);

User::where()->first();

User::findOrFail($id);

User::where()->firstOrFail();

User::where()->count();

User::where()->max/min/...('<fieldName>');

Update:

$obj = User::find($id);

$obj->name = …;

$obj->save();

//

User::where()->update(['' => '']);

6) Mass Assignment //批量赋值 \*\*\*\*\*Model\*\*\*\*\*中使用

$fillable / $guarded ： mass assignable update / insert

protected $fillable = ['','',''...];// 允许赋值 --- 白名单

protected $guarded = ['',''...]; //不允许 ['\*']:表示全部不允许赋值 ---黑名单

firstOrCreate (找不到则添加并返回实例) //User::firstOrCreate([''=>''])

firstOrNew （找不到则返回当前数据新实例） //User::firstOrNew([''=>'']) 需要手动 save

//如果数据存在，则firstOrCreate 和 firstOrNew 返回该条数据的一个实例

7) 删除

$obj = User::find($id);

$obj->delete();

//

User::where('','')->delete();

软删除 Soft Deleting \*\*\*\*Model\*\*\*\*

需要delete\_at字段

use SoftDelete

8) Query Scope

9) 事件 事件

事件绑定：

creating created updating updated saving saved deleting deleted

可在AppServiceProvider中注册事件绑定

User::creating(function ($user)) {

if( ! $user->isValid()) { return false; }

})

RelationShips

1) One To One

Base Eloquent model : hasOne

return $this->hasOne(':model',[':foreign\_key',':local\_key']);

:model : 关联的表的模型类

The Inverse Of a hasOne Relationship：

return $this->belongsTo(':baseModel”,

［:foreign\_key/local\_key','parent\_key］

2) One To Many

基本与One To One相似

hasOne → hasMany

3） Many To Many

users / roles / role\_user(枢纽表命名 , user\_id role\_id)

User:

public function roles() { return $this->belongsToMany('Role'); }

belingsToMany('Role','<自定义枢纽表名>');

$roles = User::find($id)->roles;

Role:

public function users() { return $this->belongsToMany('User'); }

$users = Role::find($id)->users;