1. 查看每次执行的sql

use Illuminate\Support\Facades\DB; // 使用DB

DB::enableQueryLog(); // 开启日志

print\_r(DB::getQueryLog());die; // 打印日志

1. 显示模板以及分配变量

return view('api.mall.show',compact('data','is\_share'));

1. 模板文件所在路径

E:\work\downton\resources\views

4、 Eloquent ORM  对象关系映射(Object Relational Mapper)工具

简介：laravel所自带的Eloquent ORM 是一个ActiveRecord实现，用于数据库操作。每个数据表都有一个与之对应的模型，用于数据表交互。

建立模型，在app目录下建立一个Student模型，即Student.php，不需要带任何后缀。

1. <?php
2. namespace App;
3. **use** Illuminate\Database\Eloquent\Model;
4. **class** Student **extends** Model{
5. //指定表名
6. **protected** $table= 'vipinfo';
7. //指定主键
8. **protected** $primaryKey= 'vip\_ID';
9. }

在Student控制器里增加一个test3方法，配置路由Route::get('test3',['uses'=>'StudentController@test3']);

1. **public** **function** test3(){
2. // all()方法查询所有数据
3. $studnets=Student::all();
4. dd($studnets);
5. //find()查询一条，依据主键查询。findOrFail()查找不存在的记录时会抛出异常
6. $student=Student::find(5);  //主键为5的记录
7. var\_dump($student['attributes']);
8. //查询构造器的使用,省略了指定表名
9. $student=Student::get();
10. var\_dump($student);
11. }

2 . 新增数据、自定义时间戳、批量赋值

（1）使用save方法新增

laravel会默认维护created\_at,updated\_at 两个字段，这两个字段都是存储时间戳，整型11位的，因此使用时需要在数据库添加这两个字段。如果不需要这个功能，只需要在模型里加一个属性：public $timestamps=false; 以及一个方法，可以将当前时间戳存到数据库

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. **protected** **function** getDateFormat(){
2. **return** time();
3. }

这样就不需要那两个字段了。

控制器里写：

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. $student=**new** Student();
2. //设定数据
3. $student->vip\_name='xiaoming';
4. $student->vip\_type='出行';
5. $student->vip\_fenshu=900;
6. $bool=$student->save();  //保存
7. echo $bool;

从数据库里取得某条记录的时间戳时，默认取得的是按日期格式化好的时间戳，如果想取得原本的时间戳，则在模型里增加asDateTime方法。

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. **protected** **function** asDateTime($val){
2. **return** $val;
3. }

（2）使用create方法新增时，需要在模型里增加：

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. **protected** $fillable=['vip\_name','vip\_fenshu','vip\_type'];   //允许批量赋值的字段

 控制器里写：

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. Student::create(['vip\_name'=>'mmm','vip\_fenshu'=>999,'vip\_type'=>'出行']);

这样即可新增成功！

（3）firstOrCreate()以属性查找记录，若没有则新增

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. $student=Student::firstOrCreate(['vip\_name'=>'mmm']);
2. echo $student;

（4）firstOrNew()以属性查找记录，若没有则会创建新的实例。若需要保存，则自己调用save方法()

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. $student=Student::firstOrNew(['vip\_name'=>'mmm']);
2. $student->save();
3. echo $student;

3.  修改数据

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. //通过模型更新数据
2. $student=Student::find(2);
3. $student->vip\_fenshu=10000;
4. $student->save(); //返回bool值
5. //通过查询构造器更新
6. $num=Student::where('vip\_ID','>',2)->update(['vip\_fenshu'=>2000]);
7. echo $num;  //返回更新的行数

4.  删除数据

**[php]** [view plain](http://blog.csdn.net/zls986992484/article/details/52824962) [copy](http://blog.csdn.net/zls986992484/article/details/52824962)

1. //(1)通过模型删除数据
2. $student=Student::find(11);
3. $student->delete(); //返回bool值
4. //(2)通过主键删除
5. $num=Student::destroy(10); //删除主键为10的一条记录
6. echo $num; //返回删除的行数
7. $num=Student::destroy(10,5); //删除多条  或者$num=Student::destroy([10,5]);
8. echo $num; //返回删除的行数

设置关联模型的表

protected $table = ‘my\_flights’;

 设置关联模型的表

protected $table = ‘my\_flights’;

设置是否表中有create\_at 和update\_at

public $timestamps = fasle;

设置日期存储格式

protected $dateFormat = 'U';

model里声明可以批量操作的字段。fillable就像是白名单

protected $fillable ＝ ［’title’］;

对应的黑名单就是 guarded 。

可以 protected $guraded = []; 来快速设置所有字段都可以自由插入。

//指定主键

protected $primaryKey= 'vip\_ID';

#### 5、自增或自减一个字段的值

DB::table('users')->increment('votes');

DB::table('users')->increment('votes', 5);

DB::table('users')->decrement('votes');

DB::table('users')->decrement('votes', 5);