17．模型修改和查询

学习要点：

1.数据修改

2.数据查询

本节课我们来学习模型中的修改和查询操作。

一．数据修改

1. 使用 get()方法通过主键获取数据，然后通过 save()方法保存修改，返回布尔值；

```
$user = UserModel::get(118);
$user->username     =   '李黑';
$user->email        =   'lihei@163.com';
$user->save();
```

2. 通过 where()方法结合 find()方法的查询条件获取的数据，进行修改；

```
$user = UserModel::where('username', '李黑')->find();
$user->username     =   '李白';
$user->email        =   'libai@163.com';
$user->save();
```

3. save()方法只会更新变化的数据，如果提交的修改数据没有变化，则不更新；
4. 但如果你想强制更新数据，即使数据一样，那么可以使用 force()方法；

```
$user->force()->save();
```

5. Db::raw()执行 SQL 函数的方式，同样在这里有效；

```
$user->price        =   Db::raw('price+1');
```

6. 如果只是单纯的增减数据修改，可以使用 inc/dec；

```
$user->price        =   ['inc', 1];
```

7. 直接通过 save([],[])两个数组参数的方式更新数据；

```
$user->save([
    'username'      =>  '李黑',
    'email'         =>  'lihei@163.com'
],['id'=>118]);
```

8. 通过 saveAll()方法，可以批量修改数据，返回被修改的数据集合；

```
$list = [
    ['id'=>118, 'username'=>'李白', 'email'=>'libai@163.com'],
    ['id'=>128, 'username'=>'李白', 'email'=>'libai@163.com'],
    ['id'=>129, 'username'=>'李白', 'email'=>'libai@163.com']
];
```

```
$user->saveAll($list);
```

9. 批量更新 saveAll()只能通过主键 id 进行更新；
10. 使用静态方法结合 update()方法来更新数据，这里返回的是影响行数；

```
UserModel::where('id', 118)->update([
    'username'      =>  '李黑',
    'email'         =>  'lihei@163.com'
]);
```

11. 另外一种静态方法 update()，返回的是对象实例；

```
UserModel::update([
    'id'            =>  118,
    'username'      =>  '李黑',
    'email'         =>  'lihei@163.com'
]);
```

12. 模型的新增和修改都是 save()进行执行的，它采用了自动识别体系来完成；
13. 实例化模型后调用 save()方法表示新增，查询数据后调用 save()表示修改；
14. 当然，如果在 save()传入更新修改条件后也表示修改；
15. 再当然，如果编写的代码比较复杂的话，可以用 isUpdate()方法显示操作；

```
//显示更新
$user->isUpdate(true)->save();
//显示新增
$user->isUpdate(false)->save();
```

二. 数据查询

1. 使用 get()方法，通过主键(id)查询到想要的数据；

```
$user = UserModel::get(129);
return json($user);
```

2. 也可以使用 where()方法进行条件筛选查询数据；

```
$user = UserModel::where('username', '辉夜')->find();
return json($user);
```

3. 不管是 get()方法还是 find()方法，如果数据不存在则返回 Null；
4. 和数据库查询一样，模型也有 getOrFail()方法，数据不存在抛出异常；
5. 同上，还有 findOrEmpty()方法，数据不存在返回空模型；
6. 通过模型->符号，可以得到单独的字段数据；

```
return $user->username;
```

7. 如果在模型内部获取数据，请不要用$this->username，而用如下方法；

```
public function getUserName()
{
```

```
        return self::where('username', '辉夜')->find()->getAttr('username');
    }
```

8. 通过 all()方法，实现 IN 模式的多数据获取；

```
$user = UserModel::all('79, 118, 128');
$user = UserModel::all([79, 118, 128]);
```

9. 使用链式查询得到想要的数据；

```
UserModel::where('gender', '男')->order('id', 'asc')
                                        ->limit(2)->select();
```

10. 获取某个字段或者某个列的值；

```
UserModel::where('id', 79)->value('username');
UserModel::whereIn('id',[79,118,128])->column('username','id');
```

11. 模型支持动态查询：getBy*，*表示字段名；

```
UserModel::getByUsername('辉夜');
UserModel::getByEmail('huiye@163.com');
```

12. 模型支持聚合查询；

```
UserModel::max('price');
```