

23. 模型的数据更新

学习要点：

1. 数据更新

本节课我们来学习模型中的数据更新的功能。

一. 数据更新

1. 使用 `find()` 方法获取数据，然后通过 `save()` 方法保存修改，返回布尔值；

```
$user = UserModel::find(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. 使用 `allowField()` 方法，允许要更新的字段，其它字段就无法写入了；

```
$user->allowField(['username', 'email'])->save(...)
```

7. 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);
```

8. 批量更新 `saveAll()` 只能通过主键 `id` 进行更新；
9. 使用静态方法 `::update()` 更新，返回的是对象实例；

```
UserModel::update([  
    'id' => 118,  
    'username' => '李黑',
```

```
        'email'          => 'lihei@163.com'
    });

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

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

10. 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；
11. 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；
12. 当然，如果在 **save()** 传入更新修改条件后也表示修改；