

## 21. 模型类型转换和数据完成

学习要点：

1. 模型类型转换
2. 模型数据完成

本节课我们来学习模型中字段的类型转换和数据操作的自动完成。

### 一. 模型类型转换

1. 系统可以通过模型端设置写入或读取时对字段类型进行转换；
2. 我们这里，通过读取的方式来演示部分效果；
3. 在模型端设置你想要类型转换的字段属性，属性值为数组；

```
protected $type = [  
    'price'      =>    'integer',  
    'status'     =>    'boolean',  
    'create_time' =>    'datetime:Y-m-d'  
];
```

4. 数据库查询读取的字段很多都是字符串类型，我们可以转换成如下类型：  
integer(整型)、float(浮点型)、boolean(布尔型)、array(数组)  
object(对象)、serialize(序列化)、json(json)、timestamp(时间戳)  
datetime(日期)

5. 由于数据库没有那么多类型演示，常用度不显著，我们提供几个方便演示的；

```
public function typeConversion()  
{  
    $user = UserModel::get(21);  
    var_dump($user->price);  
    var_dump($user->status);  
    var_dump($user->create_time);  
}
```

6. 类型转换还是会调用属性里的获取器等操作，编码时要注意这方面的问题；

### 二. 模型数据完成

1. 模型中数据完成通过 auto、insert 和 update 三种形式完成；
2. auto 表示新增和修改操作，insert 只表示新增，update 只表示修改；

```
protected $auto    = ['email'];  
protected $insert  = ['uid'=>1];  
protected $update  = [];
```

3. 先理解 insert，当我们新增一条数据时会触发新增数据完成；
4. 此时，并不需要自己去新增 uid，它会自动给 uid 赋值为 1；

```
$user = new UserModel();  
$user->username    = '李白';  
$user->password     = '123';  
$user->gender       = '男';  
$user->email        = 'libai@163.com';  
$user->price        = 100;  
$user->details      = '123';  
$user->save();
```

5. `auto` 表示新增和修改均要自动完成，而不给默认值的字段需要修改器提供；

```
public function setEmailAttr($value)  
{  
    return strtoupper($value);  
}
```

6. 新增时，邮箱字符串会被修改器自动完成大写，那数据完成的意义何在？  
7. 修改时，如果你不去修改邮箱，在数据自动完成强制完成，会自动完成大写；  
8. 也就是说，邮箱的大写，设置 `update` 更加合适，因为新增必填必然触发修改器；  
9. 对于 `update` 自动完成，和 `auto`、`insert` 雷同，自行演示；