

# Mongoose 预定义模式修饰符 Getters 与 Setters 自定义修饰符

主讲教师：（大地）

合作网站：[www.itying.com](http://www.itying.com) （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、 mongoose 预定义模式修饰符.....	1
二、 mongoose 的安装以及使用.....	2

## 一、mongoose 预定义模式修饰符

lowercase、uppercase 、trim

mongoose 提供的预定义模式修饰符，可以对我们增加的数据进行一些格式化。

```
var UserSchema=mongoose.Schema({
  name:{
    type:String,
    trim:true
  },
  age:Number,
  status:{
    type:Number,
    default:1
  }
})
```

## 二、Mongoose Getters 与 Setters 自定义修饰符

除了 mongoose 内置的修饰符以外，我们还可以通过 set（建议使用） 修饰符在增加数据的时候对数据进行格式化。也可以通过 get（不建议使用）在[实例获取数据](#)的时候对数据进行

格式化。

```
var NewsSchema=mongoose.Schema({
  title:"string",
  author:String,
  pic:String,
  redirect:{
    type:String,
    set(url){

      if(!url) return url;
      if(url.indexOf('http://')!=0  && url.indexOf('https://')!=0){
        url = 'http://' + url;
      }
      return url;
    }
  },
  content:String,
  status:{

    type:Number,
    default:1

  }
})
```

```
var NewsSchema=mongoose.Schema({
  title:"string",
  author:String,
  pic:String,
  redirect:{
    type:String,
    set(url){

      if(!url) return url;
      if(url.indexOf('http://')!=0  && url.indexOf('https://')!=0){
        url = 'http://' + url;
      }
    }
  },
  content:String,
  status:{

    type:Number,
    default:1

  }
})
```

```
    }  
    return url;  
  },  
  get: function(url){  
    if(!url) return url;  
  
    if(url.indexOf('http://')!=0 && url.indexOf('https://')!=0){  
      url = 'http://' + url;  
    }  
    return url;  
  }  
},  
content:String,  
status:{  
  
  type:Number,  
  default:1  
  
}  
})
```