

MVVM 框架

MVVM 框架是在 MVC 框架的基础上升级（改良）而来

MVC (Model, View, Controller)

MVVM (model, view, viewModel) 。

View 和 Model 之间的交互是通过 ViewModel 进行的。View 绑定到 ViewModel 中的数据，当数据变化时，View 也会自动更新。同样，当用户在 View 中执行操作时，这些操作也会自动映射到 ViewModel 中的命令，进而更新 Model。这种架构模式使得开发人员可以将 View 和逻辑分离出来，使得程序更易于维护和扩展。

1. 双向绑定

value <-> data

双向是指：数据驱动页面的显示，页面的操作改变数据。

绑定是指：自动化处理，不需要人为干预，框架帮我们做好了。

1.1. 原理

通过 `Object.defineProperty()` 来监听数据的变化，从而刷新视图。

这个方法接收三个参数，要操作的对象，要定义或修改的属性的名称，以及将被定义或修改的属性描述符。存取描述符的 `get` 和 `set` 就是属于存取描述符对象的属性。在双向数据绑定中，主要用到了这个方法的 `get` 和 `set` 操作。

`Object.defineProperty()` 和 `reflect.defineProperty()` 区别

静态方法 `Object.defineProperty()` 基本等同于 `Reflect.defineProperty()` 方法，唯一不同是前者返回一个新的对象，后者返回一个 Boolean 值。

```
let obj1 = {}
Reflect.defineProperty(obj1, 'x', {value: 7}) // true Boolean 值
obj1.x // 7

let obj2 = {}
let obj3 = Object.defineProperty(obj2, 'x', {value: 7}) // {x: 7} 新对象
obj3.x // 7
```