



链滴

Android mvvm 框架 databinding

作者: [ganxj](#)

原文链接: <https://ld246.com/article/1573213236382>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

本文参照[Data Binding Library - Android Developers 谷歌官方][Data Binding Library - Android developers]

从今天开始将为大家由浅入深（PS:希望不是从入门到放弃）的讲一下谷歌在安卓平台上的数据绑定架：**Databinding**。

前言：

2015年谷歌I/O大会上就介绍了一个框架DataBinding，堪称解决界面逻辑的黑科技，但是使用率非少，研究的人更少，那这是不是一个优秀的框架呢，是不是对项目有用呢，接下来的几天我将带大家入研究一下。

这篇文章不说技术，只了解一下DataBinding。

一、数据绑定是什么？

几年前，数据绑定便已在前端世界中风生水起，Angular.js、React.js、vue.js等热门前端框架都具备种能力。

数据绑定简单来说，就是通过某种机制，把代码中的数据和xml(UI)绑定起来，双方都能对数据进行作，并且在数据发生变化的时候，自动刷新数据。

二、数据绑定怎么用？

数据绑定分为**单向绑定**和**双向绑定**两种方式。

单向绑定就是说数据的流向是单方面的，只能从代码流向UI；

双向绑定的数据流向是双向的，当业务代码中的数据改变时，UI上的数据能够得到刷新；当用户通过U交互编辑了数据时，数据的变化也能自动的更新到业务代码中的数据上。

![[Image 1]]

单向数据绑定和双向数据绑定示例

三、DataBinding这门技术有什么优点呢？

- 1.**性能很好**，因为它0反射，而且性能比直接findViewById要高。后面的文章将会从源码角度介绍。
- 2.**谷歌原生支持**（相比与其他第三方的数据绑定框架比如[Android][]-atabinding和RoboBinding），可以说是亲儿子，有自家IDE支持。
- 3.将MVP框架变成**MVVM框架**，代码更简洁，逻辑更清晰。
- 4.**减少代码线程切换更新UI**（列表除外），除了要在列表中考考虑线程问题。
- 5.**自动检查空指针**，也不用担心空指针。

四、DataBinding这门技术一直不温不火，为什么呢？

根据我的使用感受是这么几条：

1. **不太成熟**，当然并不是说技术不成熟，只是Android Studio的IDE对xml编写时还是不太友好；代码自动补全功能做得还是有点差；另外如果 XML 中有编辑错误，在 Build 的时的提示会很莫名其妙，难排查到问题。这几个小问题有待Google日后的改善。

2. **使用局限**，在使用过程中发现它与某些技术冲突，比如插件化技术、热修复技术，如果想用DataBinding还需要另外增加开发成本来配合它。

3. **学习成本**，其实我认为这个并不算一条，因为做我们这行学什么没有成本呢，是吧。

回答完上面几个问题，那我们看DataBinding解决了哪几个痛点会被称为解决界面逻辑的黑科技的呢？(咳咳，划重点了)

- 需要多次使用findViewById，损害了应用性能且令人厌烦
- 更新UI数据需切换至UI线程，将数据分解映射到各个view比较麻烦
- 自动检查空指针，例如 `android:text="@{user.lastName}`，如果user为空，程序不会崩溃，只是显示空值。

Data Binding 解决了 Android UI 编程中的这几个痛点，官方原生支持 MVVM 模型可以让我们在不变既有代码框架的前提下，非常容易地使用这些新特性。其实在此之前，已经有些第三方的框架 ([RoAndroid][1]) 可以支持 MVVM 模型，无奈由于框架的侵入性太强，导致一直没有流行起来。

说了这么多，有没有跃跃欲试呢？

现在Android Studio 2.0已经发布了，对data binding的支持又更进了一步！