


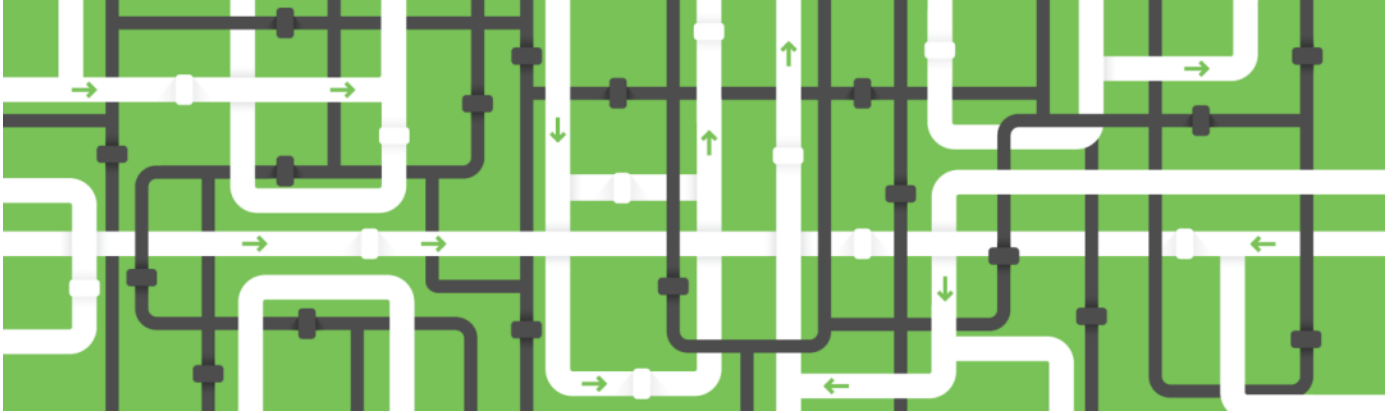
两步使用 LiveData 替换 Observable Field



谷歌开发者 
已认证的官方帐号

已关注

2 人赞同了该文章



可观察性指的是一个对象会在其数据发生变更时向其他类发出通知。**可观察性**是数据绑定库 (Data Binding) 的重要特性之一，它可以将数据和 UI 元素绑定在一起——当数据发生变化时，屏幕上的相关元素也会随之更新。

默认情况下，普通函数和字符串是**不可观察**的，这就意味着，当您在数据绑定布局中需要使用它们时，只能在新建的时候获取它们的值，但在后续的操作中，却不能得到相应的数据。

为了使对象可观察，数据绑定库中包含了一系列可观察的类，如: `ObservableBoolean`、`ObservableInt`、`ObservableDouble...` 和一些通用类、`ObservableField<T>`。这里我们将其统称为“Observable Fields”。

再后来，在我们发布Android 架构组件时首批就包含了 LiveData，这是另一个“可观察”类，并且与数据绑定库兼容。

LiveData 可以感知生命周期，这一点与 Observable Fields 相比并没有多大优势，因为 Data Binding 原本就可以检查视图活跃情况。因此对于 LiveData 来说，**它的优势在于不仅支持 Transformations，而且可以与许多架构组件 (如Room、WorkManager) 相互配合使用。**

综上，我们推荐您使用 **LiveData**。方法也非常简单，只需要两个步骤。

第一步: 用 LiveData 替换 Observable Fields

如果您是直接在数据绑定中使用 Observable Fields，只需将 `Live ObservableSomething` (或 `ObservableField <Something>`) 替换为 `LiveData <Something>` 即可。

替换前:

```

<data>
    <import type="android.databinding.ObservableField"/>
    <variable
        name="name"
        type="ObservableField<String>" />
</data>

...
<TextView
    android:text="@{name}"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

```

注意: "<" 不是错误, 而是您必须对此处的 "<" 符号进行转义。

替换后:

```

<data>
    <import type="android.arch.lifecycle.LiveData" />
    <variable
        name="name"
        type="LiveData<String>" />
</data>

...
<TextView
    android:text="@{name}"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

```

或者, 如果您是通过 ViewModel、Presenter 或 Controller 暴露可观察对象的话, 则无需更改布局, 只要用 ViewModel 中的 LiveData 替换这些 ObservableFields 即可。

替换前:

```

class MyViewModel : ViewModel() {
    val name = ObservableField<String>("Ada")
}

```

替换后:

```

class MyViewModel : ViewModel() {
    private val _name = MutableLiveData<String>().apply { value = "Ada" }

    val name: LiveData<String> = _name // Expose the immutable version of the l
}

```

第二步: 设置 LiveData 的生命周期所有者

视图的绑定类中包含一个 `setLifecycleOwner` 方法，想要从数据绑定布局观察 LiveData，必须使用该方法。

替换前:

```
val binding = DataBindingUtil.setContentView<TheGeneratedBinding>(
    this,
    R.layout.activity_data_binding
)

binding.name = myLiveData // or myViewModel
```

替换后:

```
val binding = DataBindingUtil.setContentView<TheGeneratedBinding>(
    this,
    R.layout.activity_data_binding
)

binding.lifecycleOwner = this // Use viewLifecycleOwner for fragments

binding.name = myLiveData // or myViewModel
```

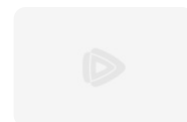
小提示: 如果要设置 *fragment* 的内容，建议使用 *fragment.viewLifecycleOwner* (而不是 *fragment* 的生命周期) 来处理潜在的分离 *fragment*。

现在，LiveData对象可以与Transformations或MediatorLiveData配合使用，完成数据转换。

我们也在 2019 年的 Android Dev Summit 上发布了一个与 LiveData 相关的视频，如下:

- 腾讯视频链接

在协程和Flow使用LiveData | ADS中
文字幕视频_腾讯视频
v.qq.com



- Bilibili 视频链接

<https://www.bilibili.com/video/BV1PJ411m7ke/>
www.bilibili.com



[点击这里](#)了解更多 LiveData 使用指南

发布于 04-21

Android