

引子

首先来让我们回顾一下 Android 目前已有的布局有哪些。

Simple	Complex
LinearLayout	RelativeLayout
FrameLayout	GridLayout
TableLayout	

我想大家使用比较多的应该是 `LinearLayout` 和 `RelativeLayout` 这两种布局文件。

实际在使用过程中，这两种布局都会有各自的不足之处。比如你要用 `LinearLayout` 来实现一个复杂的布局，那么就会产生一个很深的层级嵌套，而这显然会带来一定的性能问题。但是如果你要使用 `RelativeLayout` 来实现的话，你会发现无论在使用上还是布局编辑器上都非常难用。

有的小伙伴说，那 `LinearLayout` 和 `RelativeLayout` 混合使用不就可以解决问题了？

额...这...（兄弟，你不按套路出牌啊）

开个玩笑，因为作为Android官方来说，他们肯定有更优雅的方式嘛。这就是今天的主角-**ConstraintLayout**

ConstraintLayout 简介

`ConstraintLayout` 直译的话，就是约束性布局.我们来看官方是怎么来介绍它的：

A small , unbundled library compatible with API level 9 .
Designed to reduce nesting .
Designed to be used in Studio .

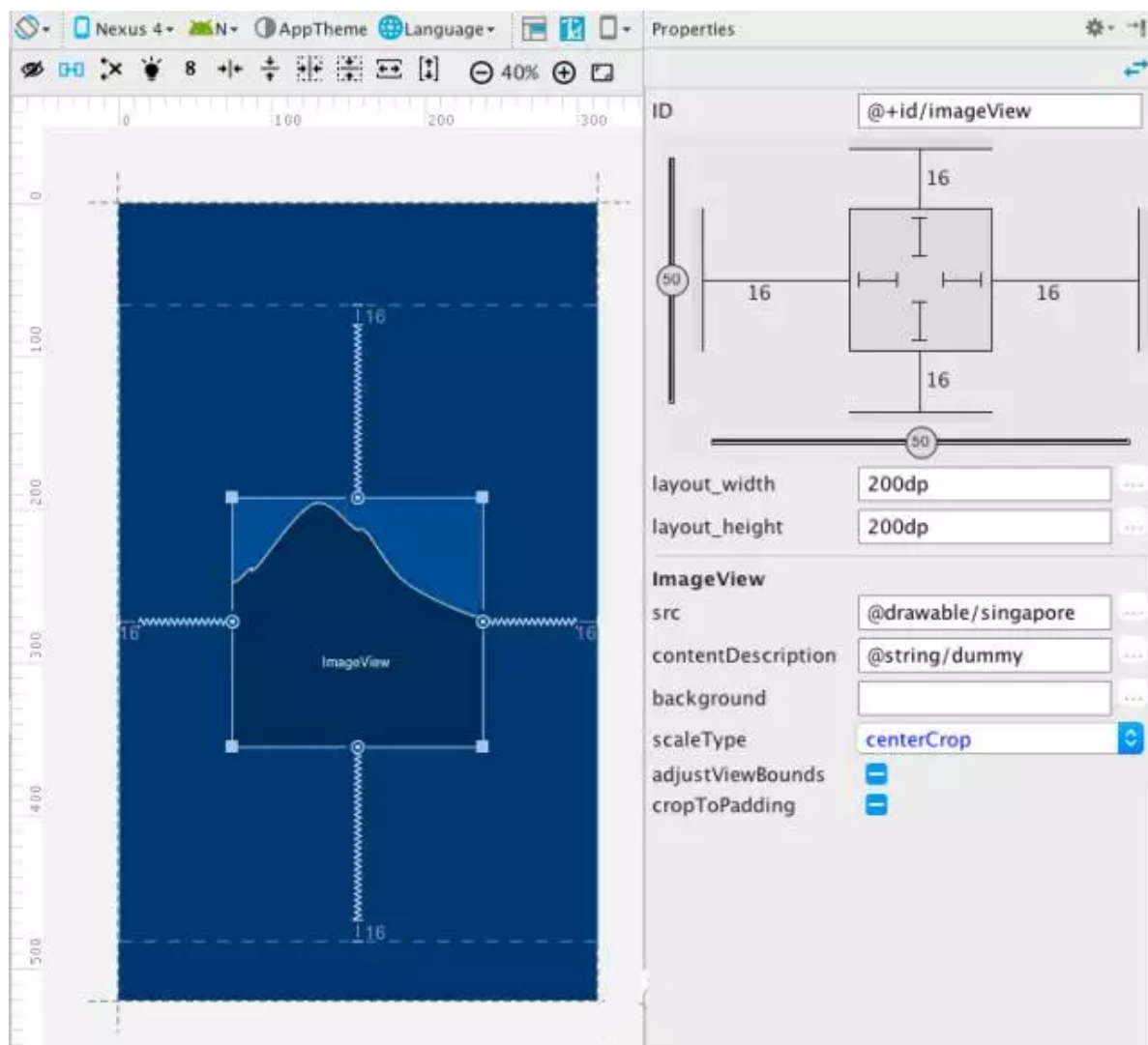
一句话，它很棒，你用不用！

可是我想知道 `ConstraintLayout` 到底是个什么呢？

`ConstraintLayout` does what `RelativeLayout` does , but more .

对，你可以理解为一个增强版的 `RelativeLayout` .

而相对于 `RelativeLayout` , `ConstraintLayout` 则更加灵活, 性能出众, 更主要的是, 布局编辑器终于可以不那么鸡肋了.



看起来很Cool! 是不是。

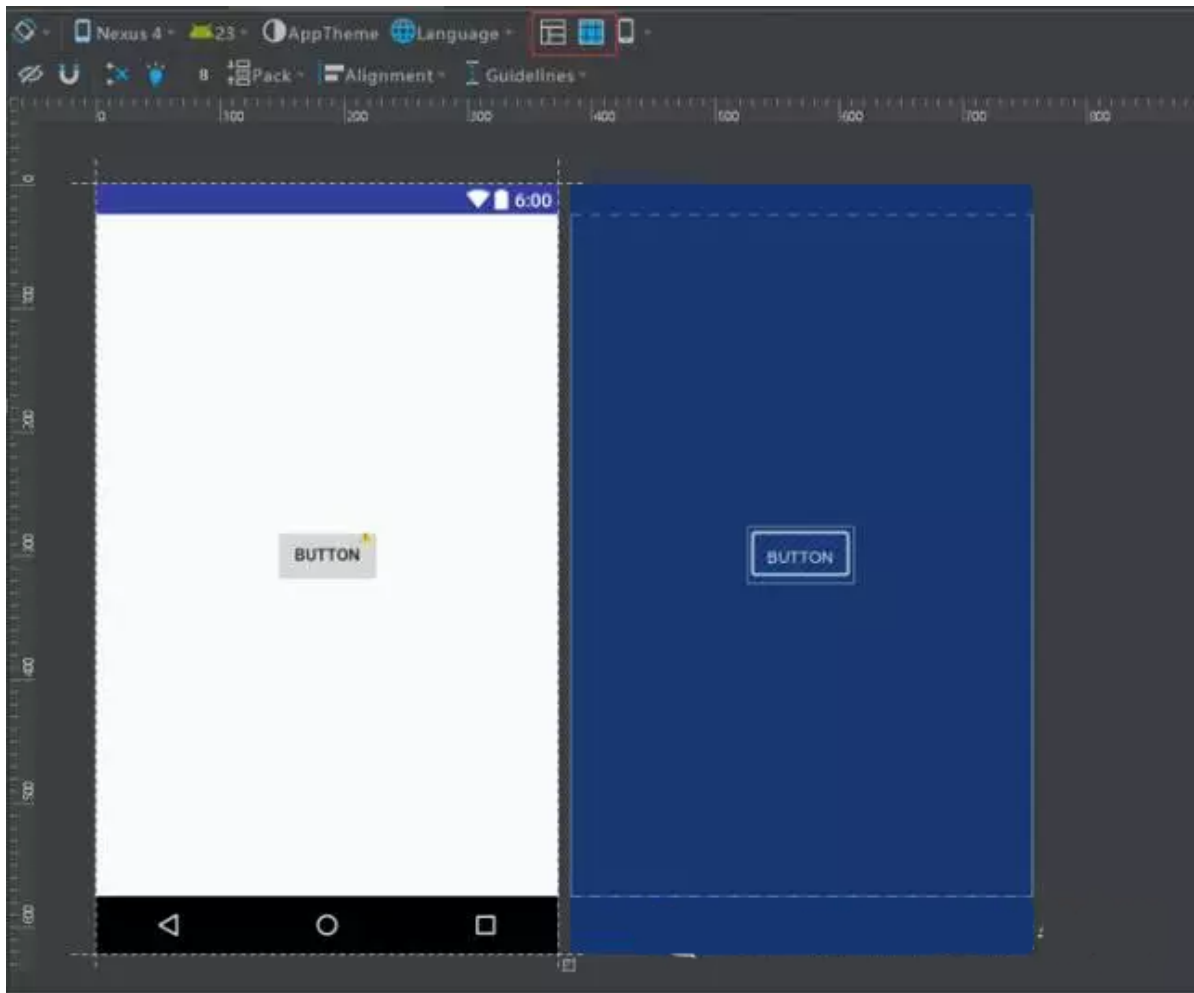
ConstraintLayout 使用

准备工作

- Android Studio 升级到2.2 preview
- 添加 `compile 'com.android.support.constraint:constraint-layout:1.0.0-alpha3'` 到 `dependencies` 中

显示

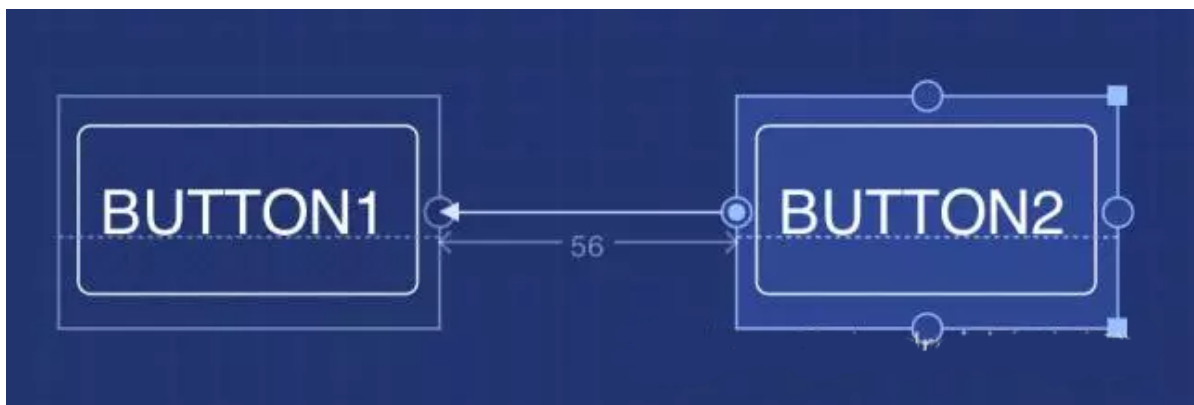
在布局编辑器中有两种显示方式: Design 和 Blueprint.



Constraints

这部分内容参考(偷个懒)了官方指导教程。

`ConstraintLayout` 的重点在于Constraints，可以直译为约束，或者关联也可以。Constraints可以帮助你保持控件对齐。你可以使用锚点(比如下图展示的Button2的四个边上圆点)来确定与之关联控件之间的对齐规则。例如，设置一个从 button 2 的左边到 button 1 的右边的约束(见下图)，意味着控件 button 2 将会位于 button 1 右边 56dp 处。



操作类型的说明



上图中显示了该控件所能操作的类型

调整控件大小的操作:



建立控件之间边关联的操作:



控件基准线约束的操作:



示例

需要进一步了解或者感兴趣的小伙伴可以直接去看看
constraint-layout 这是Android官方提供的代码示例.

关于 `ConstraintLayout` 的用法这里只简单做个介绍, 目前 `ConstraintLayout` 还处在一个快速迭代更新的阶段, 所以并不推荐在实际项目中去使用, 就我个人的实际体验上, 确实还是存在着一些问题, 当然这并不影响我们去了解它。

感想

正如 `ConstraintLayout` 官方介绍视频中多次强调的那一句话:

Our job is to make your life easier.

我相信很快 `ConstraintLayout` 会给Android 开发者带来一个不一样的体验.

虽然我感觉它来的有点晚 ...