

Flutter面试题

1.简单了解下Flutter框架，以及它的优缺点

Flutter是Google推出的一套开源跨平台UI框架，可以快速地在安卓iOS和Web平台上构建高质量的原生用户界面。同时，Flutter还是Google新研发的Fuchsia操作系统的默认开发套件。在全世界，Flutter正被越来越多的开发者和组织使用，并且Flutter是完全免费、开源的。Flutter采用现代响应式框架构建，其中心思想是使用组件来构建应用的UI。当组件的状态发生改变时，组件会重构它的描述，Flutter会对比之前的描述，以确定底层渲染树从当前状态转换到下一个状态所需要的最小更改。

优点

热重载，利用Android Studio直接一个ctrl+s就可以保存并重载，模拟器立马就可以看见效果，相比原生冗长的编译过程强很多；

一切皆为Widget的理念，对于Flutter来说，手机应用里的所有东西都是Widget，通过可组合的空间集合、丰富的动画库以及分层扩展的架构实现了富有感染力的灵活界面设计；

借助可移植的GPU加速的渲染引擎以及高性能本地代码运行时以达到跨平台设备的高质量用户体验。简单来说就是：最终结果就是利用Flutter构建的应用在运行效率上会和原生应用差不多。

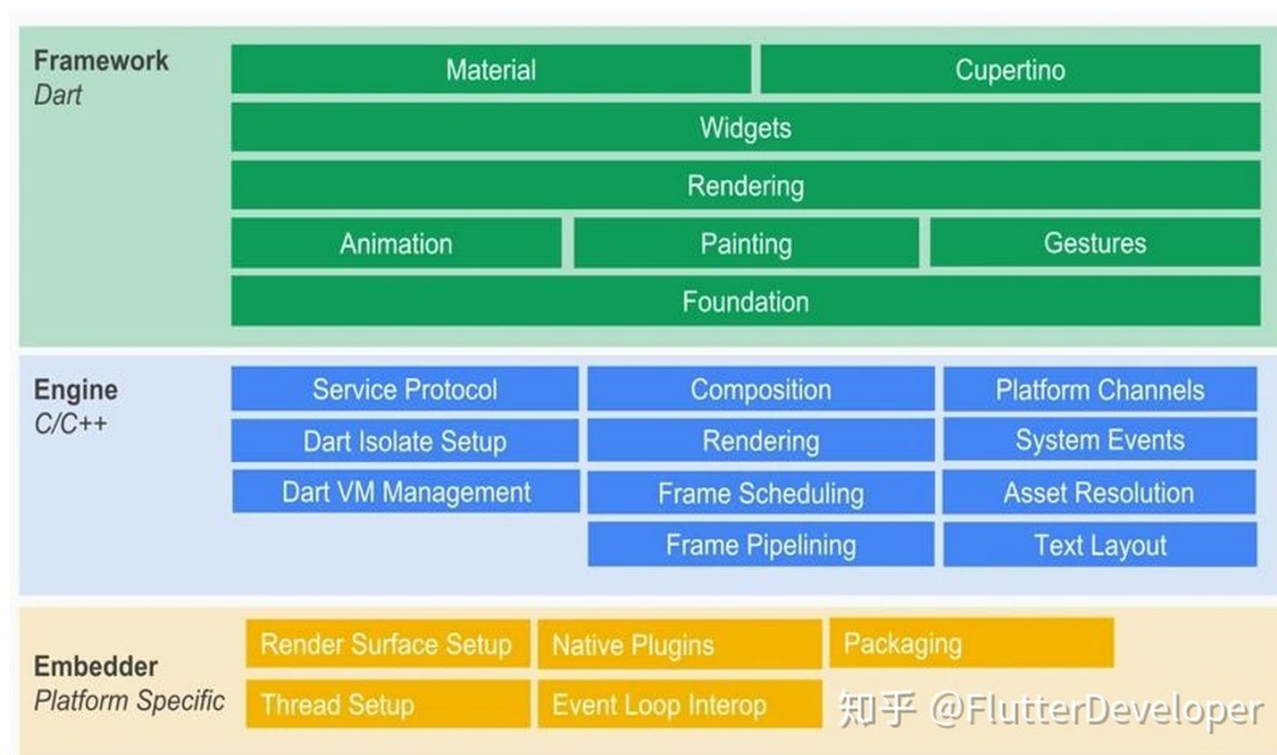
缺点

不支持热更新；

三方库有限，需要自己造轮子；

Dart语言编写，有学习成本；

2.Flutter理念架构



Flutter框架自下而上分为Embedder、Engine和Framework三层。其中，Embedder是操作系统适配层，实现了渲染Surface设置，线程设置，以及平台插件等平台相关特性的适配；Engine层负责图形绘制、文字排版和提供Dart运行时，Engine层具有独立的虚拟机，正是由于它的存在，Flutter程序才能运行在不同的平台上，实现跨平台运行；Framework层则是使用Dart编写的一套基础视图库，包含了动画、图形绘制和手势识别等功能，是使用频率最高的一层。

3.介绍下Flutter的Framework层和Engine层，以及他们的作用

Flutter的Framework层是用dart编写的框架，它实现了一套基础库，包含Material和Cupertino的UI界面，下面是通用的Widgets，之后是一些动画、绘制、渲染、手势库等。这个纯Dart实现的SDK被封装为了一个叫做dart:ui的Dart库。我们在使用Flutter写App的时候，直接导入这个库即可使用组件等功能。

Flutter的Engine层是Skia 2D的绘图引擎库，其前身是一个向量绘图软件，Chrome和Android均采用Skia作为绘图引擎。Skia提供了非常友好的API，并且在图形转换、文字渲染、位图渲染方面提供了友好、高效的表现。Skia是跨平台的，所以可以被嵌入到Flutter的iOS SDK中，而不用去研究iOS闭源的Core Graphics / Core Animation。Android自带了Skia，所以Flutter Android SDK要比iOS SDK小很多。

4.介绍下Widget、State、Context概念

Widget：在Flutter中，几乎所有的东西都是Widget。将一个Widget想象为一个可视化的组件，当你需要构建与布局直接或间接相关的任何内容时，你正在使用Widget.

Widget树：Widget以树结构进行组织。包含其他Widget的widget被称为父Widget。包含在父widget中的widget被称为子Widget

Context：仅仅是已创建的所有Widget树结构中的某个Widget的位置引用。简而言之，将context作为widget树的一部分，其中context所对应的widget被添加到此树中。一个context只从属于一个widget，它和widget一样是链接在一起的，并且会形成一个context树。

State：定义了StatefulWidget实例的行为，它包含了用于“交互/干预”Widget信息的行为和布局。应用于State的任何更改都会强制重建Widget.

5.简述Widget的StatelessWidget和StatefulWidget两种状态组件类

StatelessWidget:一旦创建就不关心任何变化，在下次构建之前都不会改变。它们除了依赖于自身的配置信息外不再依赖于任何其他信息。比如典型的Text、Row、Column、Container等，都是StatelessWidget。它的生命周期相对简单：初始化、通过build()渲染。

StatefulWidget：在生命周期内，该类Widget所持有的数据可能会发生变化，这样的数据被称为State，这些拥有动态内部数据的Widget被称为StatefulWidget。比如复选框、Button等。State会与Context相关联，并且此关联是永久性的，State对象将永远不会改变其

Context，即使可以在树结构周围移动，也仍将与该context相关联。当state与context关联时，state被视为已挂载。StatefulWidget由两部分组成，在初始化时必须要在createState()时初始化一个与之相关的State对象。

6.StatefulWidget的生命周期

initState()：Widget初始化当前State，在当前方法中是不能获取到Context的，如想获取，可以试试Future.delayed()

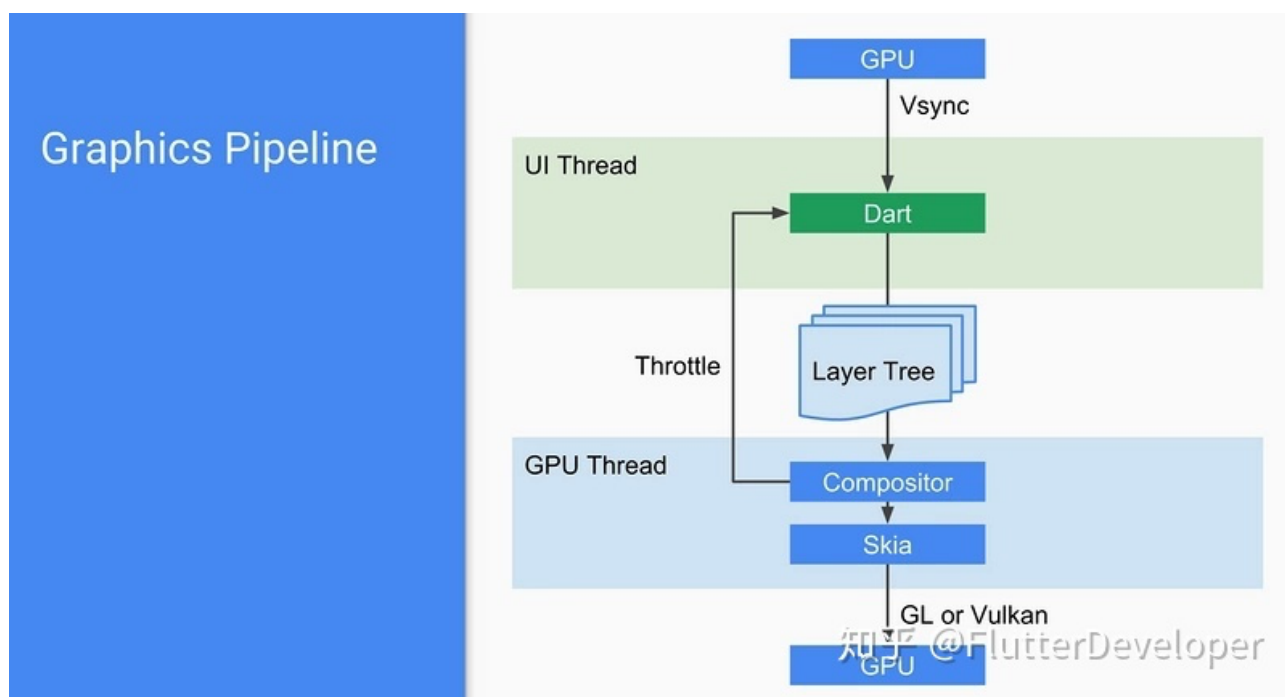
didChangeDependencies()：在initState()后调用，State对象依赖关系发生变化的时候也会调用。

deactivate()：当State被暂时从视图树中移除时会调用这个方法，页面切换时也会调用该方法，和安卓里的onPause差不多。

dispose()：Widget销毁时调用。

didUpdateWidget：Widget状态发生变化时调用。

7.简述Flutter的绘制流程



Flutter只关心向GPU提供视图数据，GPU的VSync信号同步到UI线程，UI线程使用Dart来构建抽象的视图结构，这份数据结构在GPU线程进行图层合成，视图数据提供给Skia引擎渲染为GPU数据，这些数据通过OpenGL或者Vulkan提供给GPU

8.Flutter是如何与原生进行通信的

Flutter通过PlatformChannel与原生进行交互，其中PlatformChannel分为三种：

BasicMessageChannel：用于传递字符串和半结构化的信息。

MethodChannel：用于传递方法调用。

EventChannel：用于数据流的通信。

同时PlatformChannel并非是线程安全的。