

Mobil Programlama

Flutter Framework Artchitecture



Dr.Öğr.Üyesi İsmail İŞERİ

Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Ekim 2020



ismail.iseri@omu.edu.tr



<https://www.linkedin.com/in/dr-ismail-iseri/>

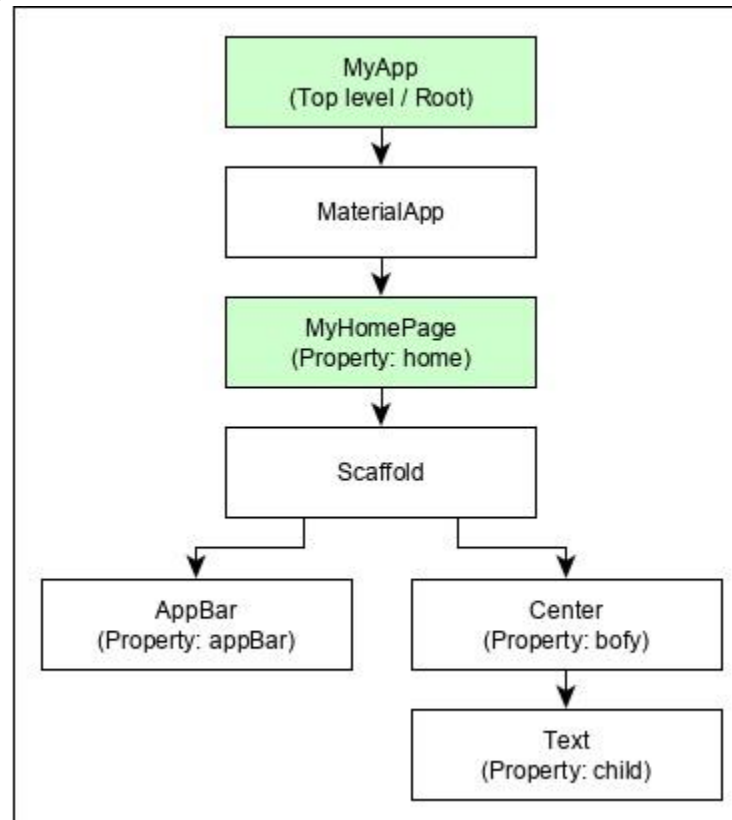


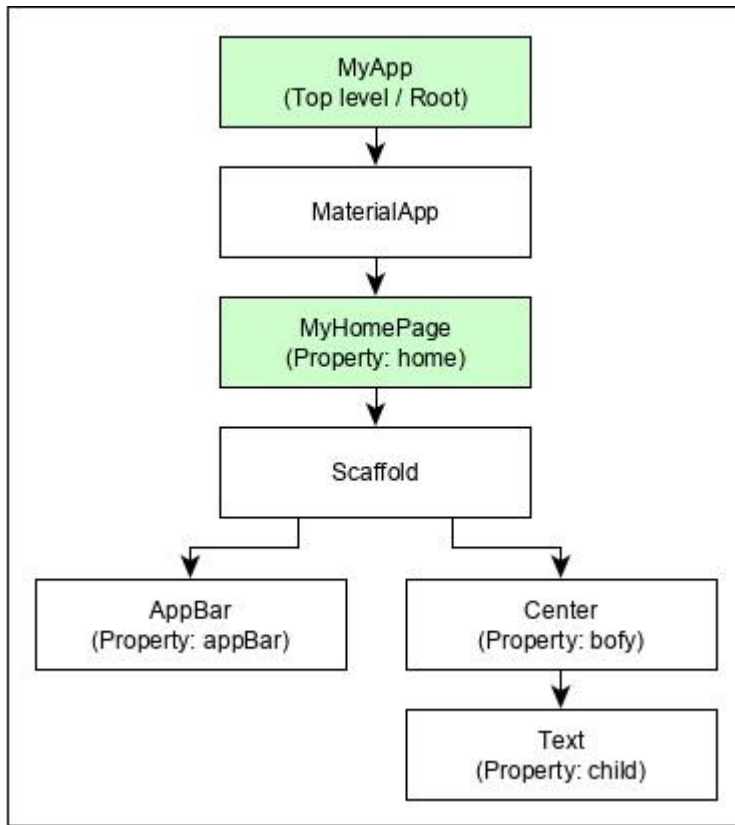
<https://personel.omu.edu.tr/tr/ismail.iseri>

Widgets

- Flutter frameworksinin temel konsepti «**Everything is a widget**» yani her şey bir widgettır.
- Widget'lar temelde uygulamanın kullanıcı arayüzünü oluşturmak için kullanılan kullanıcı arayüzü **bileşenleridir**.
- Flutter'da uygulamanın kendisi bir **widget'tır**.
- Uygulama, en üst düzey widget'tır ve kullanıcı arayüzü, yine alt widget'ları kullanarak oluşturulan bir veya daha fazla çocuk (widget) kullanılarak oluşturulur.
- Bu birleştirilebilirlik (**composability**) özelliği, herhangi bir karmaşıklıkta bir kullanıcı arayüzü oluşturmamıza yardımcı olur.

- Örneğin, merhaba dünya uygulamasının (önceki bölümde oluşturulmuş) widget hiyerarşisi aşağıdaki şemada belirtildiği gibidir -





MyApp, kullanıcı tarafından oluşturulan widget aracıdır ve Flutter native widget ögesi olan MaterialApp kullanılarak oluşturuldu.

MaterialApp, giriş sayfasının kullanıcı arayüzünü belirtmek için bir *home* özelliğine sahiptir. Bu özellik MyHomePage isimli kullanıcı tarafından oluşturulan bir widget ile temsil edilir.

MyHomePage, başka bir flutter yerel widget olan Scaffold kullanılarak oluşturulmuştur

body, ana kullanıcı arayüzünü belirtmek için kullanılır ve *appBar*, başlık kullanıcı arayüzünü belirtmek için kullanılır

Header bölümü (veya UI), flutter native widget olan AppBar kullanılarak oluşturulur ve Body arayüzü/bölümü/UI, Center widget kullanılarak oluşturulur.

Center widget'ının gerçek içeriğe atıfta bulunan child özelliği vardır ve Text widget'ı kullanılarak oluşturulur

Gestures (Hareketler)

- Flutter widget'ları, özel bir widget olan GestureDetector aracılığıyla etkileşimi destekler
- GestureDetector, alt widget'ına dokunma, sürükleme vb. Gibi kullanıcı etkileşimlerini yakalayabilen görünmez bir widgettır.
- Flutter'ın birçok native widget'ı, GestureDetector'ın kullanımıyla etkileşimi destekler.
- Ayrıca bir widget'ı GestureDetector ile oluşturarak ona bir interaktiflik özelliği ekleyebiliriz
- Sonraki bölümlerde Hareketler konusuna ayrıca değineceğiz

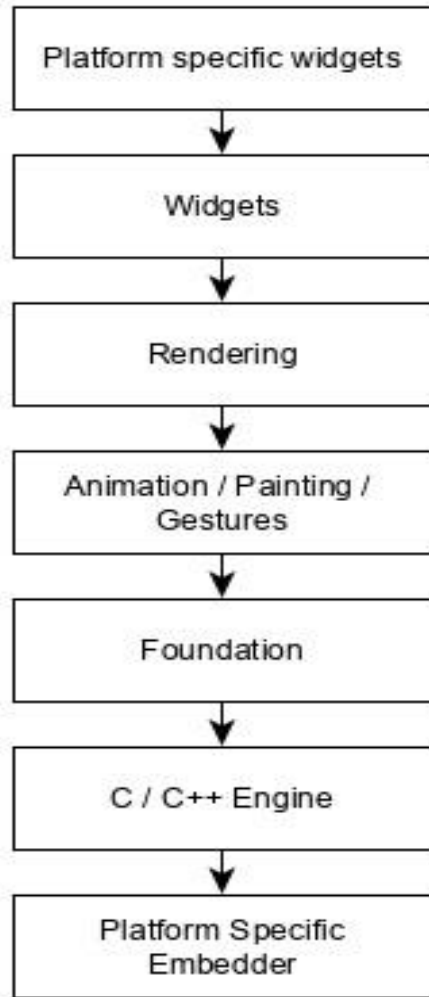
State Kavramı

- Flutter widget'leri, StatefulWidget adlı özel bir widget ögesi sağlayarak Durum devamını (State maintenance) destekler. Durum devamını desteklemek için Widget'ın StatefulWidget widget'ından türetilmesi gerekir ve diğer tüm widget'lar StatefulWidget'ten türetilmelidir.
- Flutter widget'ları yerel olarak reaktiftir (tepki veren). Bu durum, reactjs'ye benzer ve StatefulWidget, dahili durumu her değiştiğinde otomatik olarak yeniden oluşturulur. Yeniden oluşturma, eski ve yeni widget ögesi arayüz arasındaki farkı bularak ve yalnızca gerekli değişiklikleri oluşturarak (render) optimize edilir

Layers (Katmanlar)

- Flutter framework en önemli kavramı, frameworknin karmaşıklık açısından birden çok kategoriye ayrılması ve karmaşıklığı katmanlarda azaltarak açık şekilde düzenlenmiş olmasıdır.
- Bir katman, hemen sonraki seviye katmanı kullanılarak oluşturulur.
- En üstteki katman, Android ve iOS'a özel widget'tır. Bir sonraki layer flutter'a özgü native widgettır. Bir sonraki katman, düşük seviyeli oluşturucu bileşeni olan ve flutter uygulamasında her şeyi oluşturan **Rendering** layer'dır.
- Katmanlar temel platforma özgü kodlara iner

Flutter'daki bir katmanın genel görünümü aşağıdaki diyagramda belirtilmiştir.



Aşağıdaki noktalar Flutter'ın mimarisini özetlemektedir:

Flutter'da her şey bir widget'tır ve karmaşık bir widget, zaten var olan widget'lardan oluşur.

Etkileşimli özellikler, gerektiğinde GestureDetector widget'ı kullanılarak dahil edilebilir.

Bir pencere aracının durumu korunmak gerektiğinde, StatefulWidget aracı kullanılarak korunabilir.

Widget Tree / Ağaç Yapısı

