

KONU: Fluttera Giriş.

Repository: "<https://github.com/BerkeCanGoktas/Staj>"

Widgetları Extract Etmek

Birden fazla kullanılan widgetları (ve özelliklerini) tekrar tekrar yazmak yerine ayrı bir widgetmış gibi tanımlayıp kullanmak daha avantajlıdır. Bunun için yazdığımız widget yapısından extract etmek istediğimiz katmandaki widgeta gelip ampul simgesine tıklamalıyız (VS Code). Ardından burada çıkan extract widget seçeneğini seçip isim verdiğimizde widgetımız verdiğimiz isimle ayrıştırılmış olacaktır.

Not: Ampule tıklandığında gelen kullanışlı diğer seçenekler de vardır. Örneğin bu sayede widgetın üstüne kolayca başka bir widget eklenebilir.

Sayfalar Arası Geçiş Yapmak

Sayfalar arası geçiş için Navigator.push(context, MaterialPageRoute(builder: (context) => Sayfa_Adı)) komutu kullanılabilir. Bu şekilde istendiği kadar sayfa arasında geçiş yapılabilir, geri tuşuyla önceki sayfaya dönüş sağlanabilir.

Daha Temel Widgetları Kullanarak Daha Customisable Widgetlar Türetmek

Flutterın en büyük avantajlarından biri open source olmasıdır. Bu sayede isteğimize tam uymayan widgetları inceleyip daha temel widgetlardan isteğimize uygun şekilde kendimiz türetebiliriz. Örneğin butonlar RawMaterialButton widgetı flutter ile kullandığımız hazır butonları türetmek için kullanılmış daha temel bir widgettır ve butonların kaynağına gidildiğinde görülebilir ayrıca kullanımına da ulaşılabilir. Buradaki kullanımdan yola çıkar daha kişisel ve isteğimize, amaçlarımıza uygun bir buton türetilip projelerimizde kullanabiliriz.

GestureDetector Widget

Bir widgetın üstüne konduğunda o widgetı etkileşimlere duyarlı hale getirir. Bu widget sayesinde basılı tutma, tıklama, çift tıklama gibi durumlarda ne yapılacağına karar verilebilir.

Uygulama İzinleri

Uygulama izinleri android için AndroidManifest.xml ios için Info.plist dosyası içerisinde belirtilir. Kullandığımız gps gibi paketlerin dökümanlarında doğru kullanım şekli incelenmeli ve projeye uygulanmalıdır (Örneğin geolocator paketinde bu işlemler fonksiyonlarla yapılmaktadır). Uygulama izinlerinin istenmesi hem güvenlik hem de kullanıcı dostu olma açısından büyük önem taşımaktadır.

Not: Emulator kullanıldığında gps kullanımı vb. olaylarda sorun çıkabilir. Bu sorunun emulator kaynaklı olup olmadığını test etmek için gerçek bir cihazla deneme yapmak uygun olacaktır.

build, initState, deactivate Metotları

Uygulamamızdaki stateful bir sayfanın her deęişime uğradığında o sayfanın build metodu tekrar çağrılarak güncellemeler sağlanır. Ancak bazen sayfaların deęişimden ziyade, başlangıç ve sonlanma durumlarında (sayfanın ilk açılışı ve kapanması) işlem yapılması istenir. Bu durumlardan başlangıç işlemleri için initState, bitiş işlemleri için deactivate methodları override edilerek istenen işlemlerin yapılması sağlanabilir. Burada dikkat edilmesi gereken bu fonksiyonların async olamayacağıdır. Ancak bu fonksiyonların içerisinde async olan başka fonksiyonlar çağrılabilir.

İnternette Veri Çekme

Bu işlem için http paketinin kullanılması gereklidir. Bu modül içerisindeki get metoduna verilen Uri'den bir Response objesi döndürülür. Bu obje kendi oluşturduğumuz Response objesine atanıp üstünde çeşitli işlemler gerçekleştirilebilir. Örneğin erişimin durumunu kontrol etmek için statusCode, eriştiğimiz sitenin içeriğine ulaşmak için body propertyleri kullanılabilir.

Platform Spesifik Kullanıcı Arayüzü

Flutter ile aynı anda hem android hem ios projeleri yazmak mümkündür. Ancak genellikle bu iki platformdaki uygulama arayüzlerinin farklı olması istenir. Bu durumu sağlamak için dart:io kütüphanesindeki Platform sınıfı kullanılabilir. Platform sınıfının sağladığı isAndroid ve isIOS gibi metotlar ile uygulamanın çalıştırıldığı platform tespit edilip buna göre widgetlar gönderilebilir ve platforma göre arayüz ayarlaması yapılabilir.

Berke Can GÖKTAŞ

12.12.2021