

# Görsel Programlama

## Yönlendirme işlemleri

Emir ÖZTÜRK

### Navigasyon

- Sayfalar arası geçiş
- Hangi sayfadan geldiğinin bilinmesi
- Önceki sayfa sırasının saklanması
- Önceki sayfa durumunun saklanması
- Stack yapısı

Sayfalar arası geçiş yapabilmek için navigasyon kullanılmaktadır. Navigasyon işlemleri bir sayfanın diğerinin üstüne çizilmesi veya sayfalar arası değişiklik yapılması şeklinde gerçekleştirilebilmektedir. Sayfalar üst üste eklendiğinde bir önceki sayfanın verileri saklanabilmektedir. Sayfalar kapandıkça bir önceki sayfaya dönüleceğinden stack yapısında bir ilerleme gerçekleştirilmiş olur.

### Navigator

- Route
  - Push()
  - Pop()
- NamedRoute
  - PushNamed()
  - Pop()

Navigator sınıfının push ve pop isimli iki metodu bulunmaktadır. Eğer yönlendirme belirli bir string isim ile gerçekleştirilecekse push yerine pushNamed kullanılır.

### Navigator Route - Push / Pop

- Navigator.push(ctx,MaterialPageRoute);
  - MaterialPageRoute(builder: (ctx) => SayfaAdi())
- Navigator.pop(ctx);
- Navigator.pop(ctx,"return value");

Navigator.push metodu context ve bir route alır. Route türünde bir nesne tanımlamak için MaterialPageRoute sınıfı kullanılır. Bu sınıf parametre olarak bir builder alır ve builder içerisinde context alan ve sayfayı çağıran bir metot tanımlanır. Sayfadan geri dönme işlemi için ise Navigator.pop kullanılmaktadır. Geri dönüşte sayfadan bir değer döndürülmesi isteniyorsa ikinci parametre olarak bu dönüş değeri verilebilir.

### Navigator Argüman Aktarımı

- Aktarılabacak argümanların yapıcıda alınması
- MaterialPageRoute içerisindeki builder'ın kullanılması
- Oluşturulacak sayfaya parametre aktarımı
  - Yapıcı yolu ile
  - ModalRoute ile

Navigator sınıfında push sırasında bilgilerin yeni sayfaya iletilmesi gerekebilir. Bu bilgiler sayfa çağırımı esnasında parametre olarak yapıcıda verilebilir. Bir diğer parametre aktarım yöntemi ise ModalRoute'dur.

### Navigator ModalRoute Kullanımı

- Gönderilecek parametrenin MaterialPageRoute'a RouteSettings verilerek tanımlanması
- settings: RouteSettings(arguments: Gönderilecekparametreler))
- Parametre alacak sayfanın parametreyi alacağı kaynak
  - final sonuc = ModalRoute.of(context)!.settings.arguments as GönderilenTür

Push işleminde sayfanın yapıcısı çağırıldığı için parametreler bu yapıcıda verilir ve açılan sayfada da yapıcı üzerinden elde edilebilmektedir. NamedRoute kullanılmak istendiğinde ise sayfanın yolu string olarak verildiği için bu parametreler yapıcıda verilememektedir. Bu sebeple NamedRoute verildiğinde settings parametresinde RouteSettings verilerek arguments değişkeni üzerinden değişkenler iletilir. Bu değişkenler daha sonra veriyi alacak sayfada ModalRoute.of(context)!.settings.arguments üzerinden elde edilebilir. Gönderilen değişkenin türü bilindiğinde bu türe dönüşüm için as kullanılır.

### Navigator

#### Dönen değerin eldesi

- Gidilen ekranda veri girdisi
- Girilen verinin ekranı kapatmasından sonra geriye döndürülmesi
- Navigator.pop(dondurulecek\_deger)
- Navigator.push(ctx, MaterialPageRoute()).then((donen\_deger){});

Geri dönen değerin kullanılması için sayfayı çağıran push metodunun sonucu kullanılmaktadır. Dönecek olan sonuç asenkron döneceği için await ile alınabileceği gibi, then metodu da kullanılabilir. Then metodu asenkron bir işlem tamamlandığında tetiklenir ve içerisinde verilen metot çalıştırılır.

### Navigator

#### NamedRoute Tanımlama

- Anasayfada MaterialApp
- initialRoute
- routes
  - {}
  - `"/Yol":(ctx) => sayfa()`

İsimlendirilmiş yol tanımlamak için uygulamanın başında MaterialApp içerisine route'lar tanımlanır. Routes değişkenine sözlük olarak yollar verilir. Bu sözlükte anahtar yol stringi, değer ise açılacak sayfa olarak belirlenir. İstenirse ilk açılacak sayfanın belirtilmesi için initialRoute değeri verilebilir.

### Navigator

#### NamedRoute PushNamed / Pop

- Navigator.pushNamed(ctx,"Yol");
- Navigator.pop(ctx)
- Navigator.pop(ctx,"return value");

İsimlendirilmiş yollarda Navigator'ın push metodu yerine pushNamed metodu kullanılır ve ikinci parametre olarak string bir yol verilir.

## Navigator

### NamedRoute saklanması

- Yolun metninin bütünlüğü
  - /Kullanıcılar
  - /Urunler
  - /Alerts
  - /Kullanıcılar/Kullanıcı/1
  - /Urunler/1
- Yolun stringinin sınıf içerisinde saklanması
- Statik

Yol metinleri belirli bir mantık dahilinde belirlenmelidir. İsimlendirme standartlaştırılmalıdır. Ayrıca yol stringleri statik bir sınıf içerisinde saklanıp sınıflara haritalanarak olası hatalı girdilerin engellenmesi sağlanabilir.

## Navigator

### Route saklanan sınıf

- Routing mekanizmasının ayrı bir sınıfa alınması
- Kod takibini kolaylaştırmak
- Statik route isimlendirilmesi ile kullanılabilir

Routing ayrı bir sınıfa alındığında takibi kolaylaşır ve bağılılığın azaltılması sağlanır.

## Navigator

### NamedRouter'da Argüman Aktarımı

- Aktarılabilecek argümanların pushNamed() fonksiyonunda arguments ile verilmesi
- Navigator.pushNamed(ctx,"yol",arguments:ÖzelSınıf()),
- ÖzelSınıf()
- Argümanların sözlüğü

NamedRoute'a argüman aktarırken arguments özelliği kullanılabilir. Bu durumda birden fazla argüman verilmesi isteniyorsa bu argüman grubuna bir sınıf açılarak bu sınıf iletilir. Ayrıca argümanlar bir sözlük biçiminde oluşturulup gönderilebilir.

## TabBar

- Birden fazla sayfa göstermek için
- DefaultTabController
- Scaffold AppBar'ı içerisinde bottom: TabBar(tabs:)
- Tab(icon,text)
- Icon(Icons.simgAdi)

Uygulamada appBar'ın altında sayfa değişim menüsü göstermek için TabBar kullanılabilir. TabBar kullanabilmek için DefaultTabController tanımlanması gerekmektedir. AppBar içerisinde bottom özelliğine TabBar verilebilir. TabBar tabs özelliği üzerinden bir Tab listesi alır. Tab listesi ise icon almaktadır. Icon'lar Icons statik sınıfı içerisinde kullanılabilir fakat bunun için bu statik sınıfın Icon sınıfına parametre olarak verilmesi gerekmektedir.

## TabBar

Sayfalar arası geçiş

- Scaffold body içerisinde TabBarView()
- TabBarView(children:)
- Her çocuk düğüm bir sayfa
- TabBarView -> DefaultTabController -> TabBar
- Otomatik geçiş

TabBar'ın içeriğini göstermek için TabBarView tanımlanır. TabBarView Scaffold body'si içerisine eklenebilir. TabBarView'lar index sırasına göre eklenir ve tıklama esnasında belirlenen indeksteki sayfa gösterilir.

## BottomNavigationBar

- Scaffold
  - bottomNavigationBar
- BottomNavigationBar
  - backgroundColor
  - SelectedItemColor
  - UnselectedItemColor
  - currentIndex

Bir diğer yönlendirme widget'ı ise BottomNavigationBar olarak adlandırılır. BottomNavigationBar için eleman olarak BottomNavigationBarItem eklenmeli ve hangi sayfada olduğunun bilinmesi amacı ile currentIndex atanmalıdır.

## BottomNavigationBar

### Geçiş

- Items
  - BottomNavigationBarItem(icon, title)
  - onTap
- Geçiş elle yapılmalı
  - Sayfalar birer widget listesine eklenir
  - selectedIndex tanımlanmalı (isim farklı olabilir)
  - setState() ile onTap yapıldığında index güncellenmeli

Item'lar birer simge ve text olarak birer title alırlar. Bu item'lara tıklandığında bir olayın gerçekleşmesi için onTap kullanılır.

Sayfalar arası geçiş mantığının yazılması gerekmektedir. Scaffold body içerisine o anki sayfanın verilmesi gerekmektedir. onTap esnasında index değiştirilmeli ve yeniden çizdirme işlemi için setState yapılmalıdır.

## Drawer

- Scaffold
- drawer:
- Drawer(child)
- Menu başka bir widget olarak oluşturulabilir
- Menü elemanlarının onTap metoduna dışarıdan fonksiyon parametresi verilebilir
- Navigator kullanılarak yönlendirme yapılabilir

## Drawer

- Navigator.push ile stack dolabilir
- Navigator.push yerine pushReplacement
- Navigator.pushNamed yerine pushReplacementNamed
  - Geri tuşu olmaz
  - Stack'e atılmaz