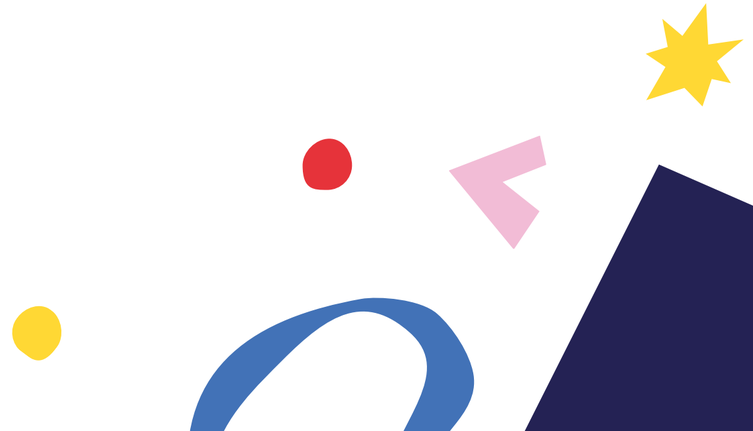
Flutter
Ankara

FLUTTER

PageView Kullanımı

SEDANUR SAVAŞ

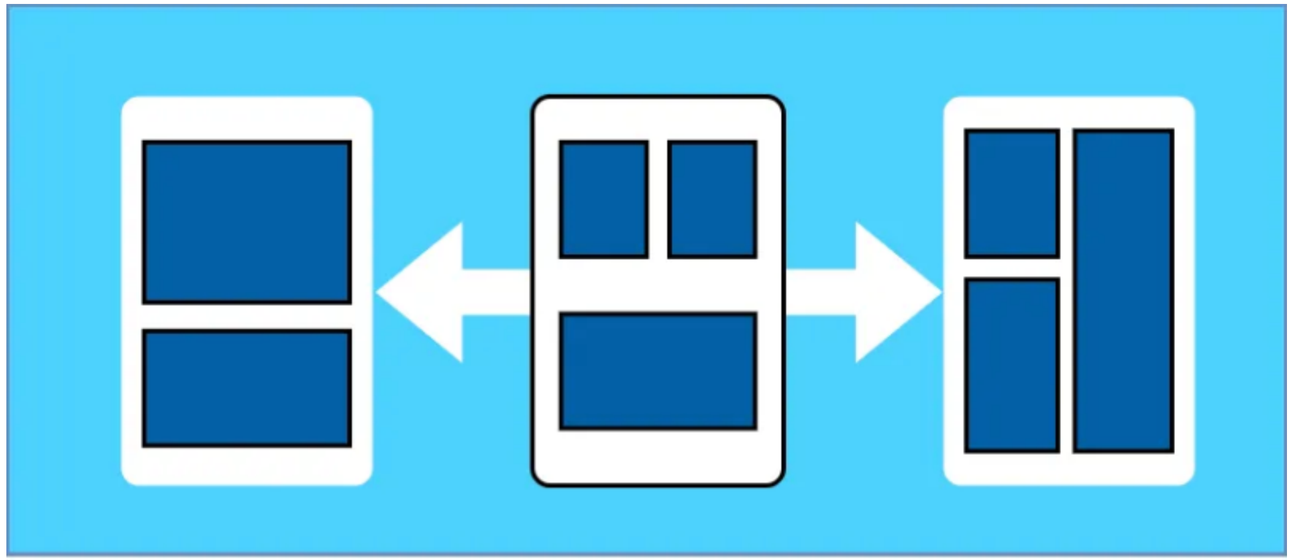
Junior Flutter Developer



01

PageView sınıfı, Flutter'da yatay veya dikey olarak kaydırılabilen sayfa tabanlı bir liste oluşturmanızı sağlar.

Ekranlar arasında geçiş yapmanın basit bir yoludur.



PageView

PageView Nerelerde Kullanılır:

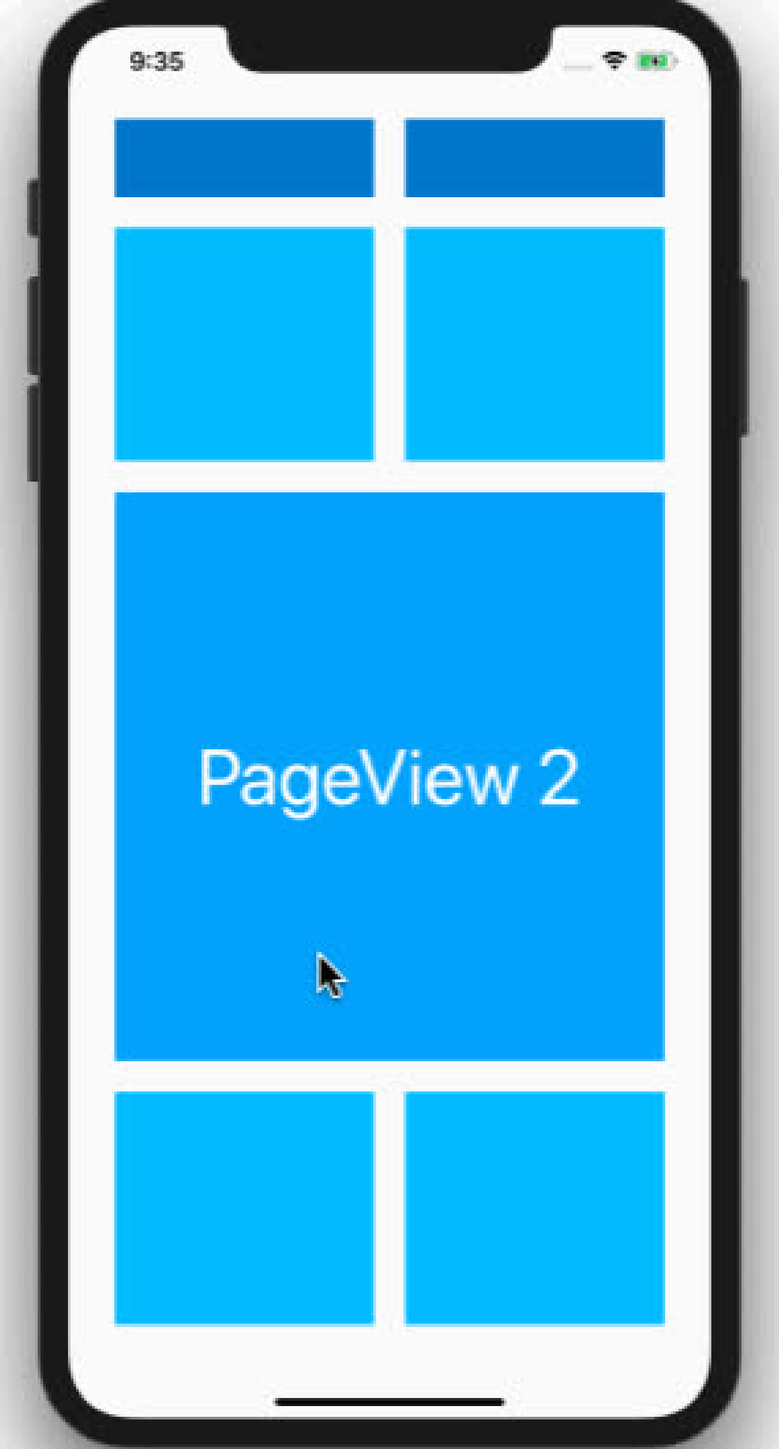
1. Onboarding Ekranları
2. Resim Galerisi
3. Ürün Kataloğu
4. Haber Akışı
5. Çok Sayfalı Formlar

.

.

.

03



04

Nasıl Kullanılır?

İlk olarak bir PageController oluşturun.

```
final controller = PageController(
);
```

Başlangıç sayfanızı atayın.

```
final controller = PageController(
  initialPage: 1,
);
```

05

Nasıl Kullanılır?

Sonrasında PageView ı oluşturun.

```
Final pageView = PageView(
  controller: controller,
  children: [
    MyPage1Widget(),
    MyPage2Widget(),
    MyPage3Widget(),
  ],
);
```

Nasıl Kullanılır?

Sayfalarınızın scroll yapısını isteğinize göre yatayda(horizontal) ya da dikeyde(vertical) ayarlayın.

```
Final pageView = PageView(
      controller: controller,
      scrollDirection: Axis.vertical,
      children: [
        Page1(),
        Page2(),
        Page3(),
      ],
);
```

```
PageView({
  Key? key,
  Axis scrollDirection = Axis.horizontal,
  bool reverse = false,
  PageController? controller,
  ScrollPhysics? physics,
  bool pageSnapping = true,
  void Function(int)? onPageChanged,
  List<Widget> children = const <Widget>[],
  DragStartBehavior dragStartBehavior = DragStartBehavior.start,
  bool allowImplicitScrolling = false,
  String? restorationId,
  Clip clipBehavior = Clip.hardEdge,
  ScrollBehavior? scrollBehavior,
  bool padEnds = true,
});
```

PageView.builder

İsteğe bağlı olarak oluşturulan widget'ları kullanarak sayfa sayfa çalışan kaydırılabilir bir liste oluşturur.

- Büyük veri kümesi ile çalışırken idealdir. Örneğin, binlerce veya daha fazla sayfaya sahip bir liste.
- Sayfa içeriği genellikle veri kaynağına bağlı olarak dinamik olarak oluşturulur.
- Bellek kullanımını optimize etmek için kullanılabilir çünkü yalnızca görünümde olan sayfaları oluşturur.

dart

```
PageView.builder(
  itemCount: myPageCount,
  itemBuilder: (context, index) {
    return myPageWidget(index);
  },
)
```

PageView.custom

Özel bir alt modelle sayfa sayfa çalışan kaydırılabilir bir liste oluşturur.

- Daha fazla kontrol gerektiren karmaşık sayfa görünümleri için uygundur.
- Özel davranışlar eklemek veya belirli kaydırma özelliklerini uygulamak için gereksinim olduğunda kullanışlıdır.
- Sayfa koleksiyonunun oluşturulması ve yönetilmesi için özelleştirilmiş bir yaklaşım gerektiğinde kullanılabilir.

dart

```dart
PageView.custom(
  controller: myCustomController,
  physics: myCustomPhysics,
  childrenDelegate: MyChildrenDelegate(
    children: myPageWidgets,
  ),
)
```

Hangi yöntemin kullanılacağına karar verirken özellikle;

- sayfa sayısı veya içeriğin dinamik olup olmadığına,
- özel davranışların gerekip gerekmediğine ve performansın ne kadar önemli olduğuna dikkat edilmelidir.

Genel olarak, **PageView.builder** büyük veri kümesi ile çalışırken ve basit bir kullanım sağlamak istendiğinde tercih edilirken, **PageView.custom** daha karmaşık ve özelleştirilmiş bir davranış gerektiğinde kullanılır.

Teşekkürler

Sedanur Savaş

https://www.linkedin.com/in/sedsax3/

Flutter Ankara