

LISTVIEW UYGULAMALARI

Örnek 1:

◆ ListView Nedir?

Flutter'da ListView, kaydırılabilir (scrollable) listeler oluşturmak için kullanılan temel bileşenlerden biridir. İçerisinde çok sayıda öge bulunduran içerikleri verimli şekilde göstermek için kullanılır.

◆ Adım Adım ListView Örneği

Örneğimizde, üniversite öğrencileri için basit bir "**Derslerim**" uygulaması yapacağız. Bu uygulama, öğrencilerin aldığı derslerin listesini görüntüleyen ve her derse tıklanıldığında basit bir mesaj gösteren yapıda olacak.

▶ Adım 1: Flutter Projesini Oluştur

Öncelikle Android Studio veya VS Code'da yeni bir Flutter projesi oluşturun:

```
flutter create derslerim
```

Proje klasörüne geçin:

```
cd derslerim
```

▶ Adım 2: main. Düzeni

Varsayılan main.dart dosyasının içeriğini temizleyip aşağıdaki gibi güncelleyin:

```
import 'package:flutter/material';
```

```
void main() {  
  runApp(MyApp());  
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Derslerim Uygulaması',  
      home: DersListesiSayfasi(),  
    );  
  }  
}
```

```
}
```

Burada, temel Material uygulamamızı oluşturduk.

► Adım 3: Ders Modelini Oluşturma

Dersleri daha düzenli tutmak için basit bir Ders modeli oluşturalım:

```
class Ders {  
    final String ad;  
    final String hoca;  
  
    Ders({required this.ad, required this.hoca});  
}
```

► Adım 4: Ders Verilerini Hazırlama

Örnek verileri içeren ders listesini oluşturalım:

```
final List<Ders> dersler = [  
    Ders(ad: 'Algoritma ve Programlama', hoca: 'Dr. Ahmet Yılmaz'),  
    Ders(ad: 'Veri Yapıları', hoca: 'Prof. Dr. Ayşe Demir'),  
    Ders(ad: 'Yapay Zeka', hoca: 'Dr. Mehmet Çetin'),  
    Ders(ad: 'Mobil Programlama', hoca: 'Doç. Dr. Selin Korkmaz'),  
    Ders(ad: 'Veritabanı Sistemleri', hoca: 'Dr. Emre Aksoy'),  
];
```

► Adım 5: ListView ile Dersleri Listeleme

Şimdi bu verileri ekranda gösterecek olan DersListesiSayfasi widget'ını oluşturalım:

```
class DersListesiSayfasi extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(  
                title: Text('Derslerim'),  
            ),  
            body: ListView.builder(  

```

```
itemCount: dersler.length, // Liste uzunluğu

itemBuilder: (context, index) {

  final ders = dersler[index]; // listedeki ders

  return ListTile(

    leading: Icon(Icons.book, color: Colors.blue),

    title: Text(ders.ad),

    subtitle: Text(ders.hoca),

    trailing: Icon(Icons.arrow_forward_ios),

    onTap: () {

      // Ders üzerine tıklanıldığında ekranda mesaj gösterme

      ScaffoldMessenger.of(context).showSnackBar(

        SnackBar(content: Text('${ders.ad} seçildi!')),

      );

    },

  );

},

);

},

),

);

}

}
```

► Adım 6: Çalıştır ve Test Et

Projenizi terminalde aşağıdaki komutla çalıştırın:

shell

flutter run

Artık, derslerinizi görebileceğiniz basit ve güzel bir liste uygulamanız var:

- Liste scrollable yani uzun olduğunda aşağı yukarı kaydırılabilir.
- Her bir öğeye dokunduğunuzda aşağıda küçük bir mesaj (SnackBar) göreceksiniz.

◆ Kod Açıklamaları (Detaylı)

- **ListView.builder:**

- Çok sayıda öğeyi verimli şekilde oluşturur ve ekranda yalnızca görüntülenebilecek öğeleri belleğe yükler.
- itemCount: listenin kaç eleman içerdiğini belirtir.
- itemBuilder: her bir öğe için görünümü oluşturmak üzere çağırılır.
- **ListTile:**
 - Liste öğelerinin standart Flutter widget'ıdır.
 - Özellikleri:
 - leading: Sol tarafta görünen simge veya widget.
 - title: Ana başlık metni.
 - subtitle: Alt başlık metni.
 - trailing: Sağ taraftaki simge veya widget.
 - onTap: Tıklama durumunda çalışan fonksiyon.
- **SnackBar:**
 - Kullanıcıya kısa mesaj göstermek için kullanılır. Özellikle durum bildirimi için idealdir.

◆ **ListView Kullanım Alanları ve Performans İpuçları:**

- Çok sayıda öğe içeren listeler (yüzlerce, binlerce öğe) için **ListView.builder** kullanmak en iyi yöntemdir.
- Kısa listeler (10-20 öğe) için `ListView(children: [])` doğrudan kullanılabilir.
- Performans için `ListView.builder` tercih edilmelidir, çünkü sadece görünür öğeleri belleğe yükler.

.....

Örnek 2:

Dersleri **JSON** formatında tutup bu veriyi Flutter uygulamasında okuyup listelemek için detaylı bir örnek yapalım.

🔴 **Flutter'da JSON ile ListView Kullanımı (Ayrıntılı Örnek)**

Örneğimizin amacı:

- JSON dosyasındaki ders verilerini almak.
- Bu verileri Dart nesnesine dönüştürmek.
- Oluşturulan nesne listesini ekranda göstermek.

▶ Adım 1: Projeyi oluşturun

Yeni bir Flutter projesi oluşturun veya mevcut projenizi kullanın:

```
flutter create json_derslerim
```

```
cd json_derslerim
```

▶ Adım 2: JSON Verisi Oluşturun

Proje klasörünüzde assets adında yeni bir klasör oluşturun ve içerisine dersler.json adında bir dosya ekleyin:

 **assets/dersler.json**

```
[  
  {  
    "ad": "Algoritma ve Programlama",  
    "hoca": "Dr. Ahmet Yılmaz"  
  },  
  {  
    "ad": "Veri Yapıları",  
    "hoca": "Prof. Dr. Ayşe Demir"  
  },  
  {  
    "ad": "Yapay Zeka",  
    "hoca": "Dr. Mehmet Çetin"  
  },  
  {  
    "ad": "Mobil Programlama",  
    "hoca": "Doç. Dr. Selin Korkmaz"  
  },  
  {
```

```
"ad": "Veritabanı Sistemleri",  
"hoca": "Dr. Emre Aksoy"  
}  
]
```

▶ Adım 3: pubspec.yaml Düzenlemesi

pubspec.yaml dosyasına aşağıdaki satırları ekleyin:

flutter:

assets:

- assets/dersler.json

Daha sonra pubspec.yaml dosyasını kaydedin ve konsoldan şu komutu çalıştırın:

flutter pub get

▶ Adım 4: Ders Modelini Tanımlayın (ders.dart)

Projenizde lib/models/ders.dart dosyasını oluşturun ve aşağıdaki gibi düzenleyin:

 **lib/models/ders.dart**

```
class Ders {  
  final String ad;  
  final String hoca;  
  
  Ders({required this.ad, required this.hoca});  
  
  factory Ders.fromJson(Map<String, dynamic> json) {  
    return Ders(  
      ad: json['ad'],  
      hoca: json['hoca'],  
    );  
  }  
}
```

- fromJson fonksiyonu JSON'dan veri okurken kullanılır.
-

▶ Adım 5: JSON Verisini Okuyan Fonksiyonu Yazın

Yeni dosya oluşturun: lib/services/json_service.dart

 **lib/services/json_service.dart**

```
import 'dart:convert';

import 'package:flutter/services.dart';

import '../models/ders.dart';

class JsonService {

  static Future<List<Ders>> dersleriOku() async {

    // JSON dosyasını okuyoruz

    final String response = await rootBundle.loadString('assets/dersler.json');

    // JSON'u decode edip listeye dönüştürüyoruz

    final List<dynamic> data = jsonDecode(response);

    // JSON nesnesini Ders sınıfına dönüştürme

    return data.map((ders) => Ders.fromJson(ders)).toList();

  }

}
```

▶ Adım 6: Ana Sayfada JSON Verilerini Listeleyin (main.dart)

 **lib/main.dart**

Ana ekranınız şöyle olacak:

```
import 'package:flutter/material.dart';

import 'models/ders.dart';

import 'services/json_service.dart';

void main() {

  runApp(MyApp());

}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'JSON Ders Listesi',  
      home: DersListesiSayfasi(),  
    );  
  }  
}
```

```
class DersListesiSayfasi extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text('JSON ile Dersler')),  
      body: FutureBuilder<List<Ders>>(  
        future: JsonService.dersleriOku(), // JSON'dan verileri alıyoruz  
        builder: (context, snapshot) {  
          if (snapshot.connectionState == ConnectionState.waiting) {  
            return Center(child: CircularProgressIndicator());  
          } else if (snapshot.hasError) {  
            return Center(child: Text('Hata oluştu: ${snapshot.error}'));  
          } else {  
            final dersler = snapshot.data!;  
            return ListView.builder(  
              itemCount: dersler.length,  
              itemBuilder: (context, index) {  
                final ders = dersler[index];  
                return ListTile(  
                  leading: Icon(Icons.book, color: Colors.blue),  
                  title: Text(ders.ad),
```



```
        subtitle: Text(ders.hoca),
        trailing: Icon(Icons.arrow_forward_ios),
        onTap: () {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text('${ders.ad} seçildi!')),
          );
        },
      );
    },
  );
}
},
),
);
}
```

- FutureBuilder:
 - Asenkron verileri bekler ve alındığında otomatik olarak ekrana çizer.
 - snapshot.connectionState: Veri yükleniyor mu, geldi mi kontrolü yapar.
 - Veri yüklenirken yüklenme animasyonu (CircularProgressIndicator) gösterir.
 - Hata olursa hata mesajını gösterir.

Adım 7: Uygulamayı Çalıştırma

Terminalden aşağıdaki komutla uygulamayı çalıştırın:

```
flutter run
```

Uygulamanız JSON'dan dersleri çekip ekranda liste halinde gösterecek. Derslere tıklayınca SnackBar'da basit mesaj görünecek.

Sonuç ve Ek Geliştirme Önerileri

Bu örnekle Flutter'da JSON verisini okuyup ListView kullanarak ekrana yazdırdık.

- Ders detay sayfası oluşturabilir, JSON içerisine derslerin açıklaması, dönem bilgisi gibi alanlar ekleyebilirsiniz.

- JSON dosyasını web API'den alarak gerçek zamanlı veri gösterimi sağlayabilirsiniz.

Böylece bilgisayar mühendisliği öğrencileri JSON veri kullanarak Flutter uygulaması geliştirmeyi kolaylıkla kavrayabilirler.