

FLUTTER

API Verisini Listeleme

27. hafta • İnternet ve Uzak Veri

36 hafta, 72 bağımsız çalışan Flutter projesi, gerçek uygulama ekranları, uygulamalı dersler ve öğrenci geri bildiriminden sonra incelenebilen çözüm kaynakları.

Bu kitap güncel ders kaynaklarından otomatik üretilmiştir; özet ya da jenerik şablon değildir.

Öğrenme Haritası

Hafta	Ders	Projeler
27	API Verisini Listeleme	Gönderi Akışı • Toplu Taşıma Duyurusu

27. Hafta — API Verisini Listeleme

Bu hafta ne yapacağız?

Tek bir HTTP nesnesini göstermekten daha gerçekçi bir adıma geçeceğiz. Uzak API'den gelen listeyi modele dönüştürecek, `ListView.builder` ile verimli biçimde çizecek, yerel arama ve filtre uygulayacak, yenileme ve ayrıntı görünümü ekleyeceğiz.

Kazanımlar

- JSON dizisini model listesine dönüştürme
- Uzak veriyi `ListView.builder` ile göstermek
- Asıl listeyi koruyarak türetilmiş filtre sonucu üretme
- Arama, kaydetme ve çoklu filtre koşullarını birleştirme
- `RefreshIndicator` ile uzak veriyi yeniden isteme
- Alt sayfa ve genişleyen kartla ayrıntı gösterme
- Boş, yükleniyor, hata ve dolu liste durumlarını yönetme

Çözümlü uygulama — API Gönderi Akışı

Model

```
class Gonderi {  
  Gonderi({  
    required this.id,  
    required this.baslik,  
    required this.icerik,  
    this.kaydedildi = false,  
  });  
  
  final int id;  
  final String baslik;  
  final String icerik;  
  bool kaydedildi;  
}
```

`id`, `baslik` ve `icerik` API'den gelir. `kaydedildi` yalnız bu uygulamanın kullanıcı durumudur; uzak JSON içinde bulunması gerekmez.

Uzak listeyi alma

```

final uri = Uri.parse(
  'https://jsonplaceholder.typicode.com/posts?_limit=12',
);

final yanit = await client.get(uri).timeout(
  const Duration(seconds: 8),
);

final liste = jsonDecode(yanit.body) as List;
return liste
  .map((e) => Gonderi.fromJson(
    Map<String, dynamic>.from(e as Map),
  ))
  .toList();

```

Arama ve kaydedilenler filtresi

```

List<Gonderi> get gorunen => (tum ?? []).where((gonderi) {
  final metin = sorgu.toLowerCase();
  final aramayaUygun =
    gonderi.baslik.toLowerCase().contains(metin) ||
    gonderi.icerik.toLowerCase().contains(metin);
  final kaydaUygun = !sadeceKayitli || gonderi.kaydedildi;
  return aramayaUygun && kaydaUygun;
}).toList();

```

Asıl tum listesi silinmez veya daraltılmaz. Arama metni değiştiğinde görünür liste yeniden hesaplanır; böylece filtre kapatılınca bütün uzak veriler geri gelir.

Verimli liste üretme

```

ListView.builder(
  itemCount: liste.length,
  itemBuilder: (context, index) {
    final gonderi = liste[index];
    return GonderiKarti(gonderi: gonderi);
  },
)

```

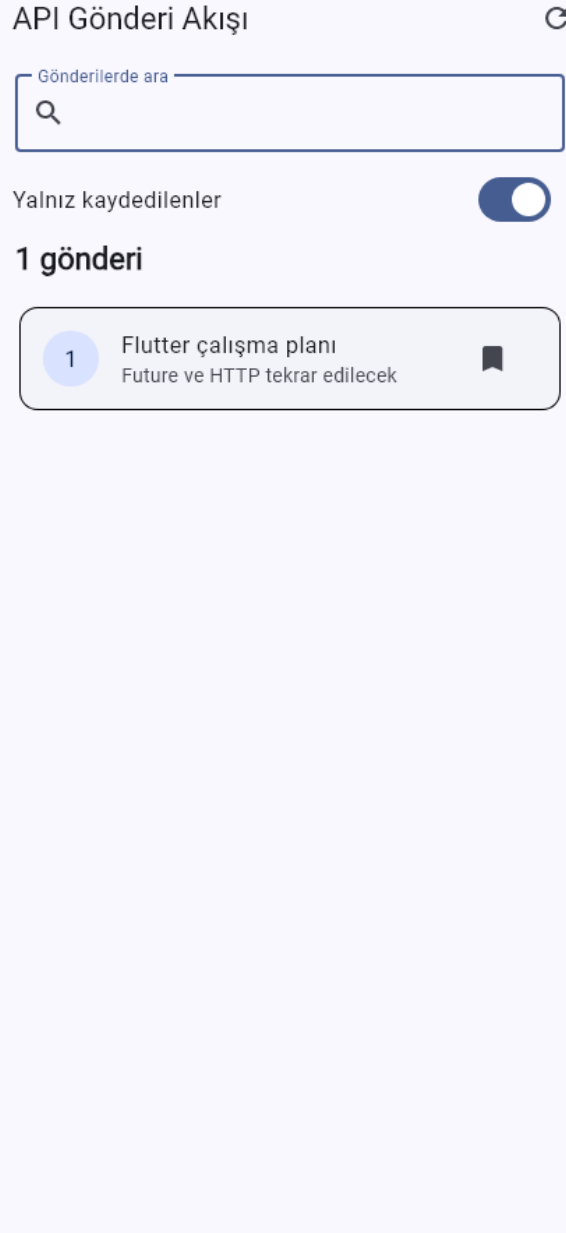
builder, yalnız ekranda ihtiyaç duyulan satırları oluşturduğu için uzun API listelerinde doğrudan Column kullanmaktan daha uygundur.

Ayrıntı alt sayfası

Kart seçildiğinde showModalBottomSheet ile başlık, tam içerik ve gönderi kimliği gösterilir. Liste kartında ise uzun metin maxLines ve TextOverflow.ellipsis ile sınırlandırılır.

Gerçek uygulama ekranı

Test üç uzak gönderiyi çözümlemiş, “Flutter” aramasıyla sonucu bire indirmiş, gönderiyi kaydetmiş ve yalnız kaydedilenler filtresinde doğru kartın kaldığını doğrulamıştır.



API Gönderi Akışı gerçek Flutter ekranı

Özgün uygulama — Canlı Ulaşım Duyuruları

Bu uygulama uzak gönderileri eğitim amacıyla ulaşım duyurusu modeline dönüştürür. Gerçek kurum API'sine geçerken yalnız servis ve fromJson eşlemesi değişir; filtreleme arayüzü aynı kalır.

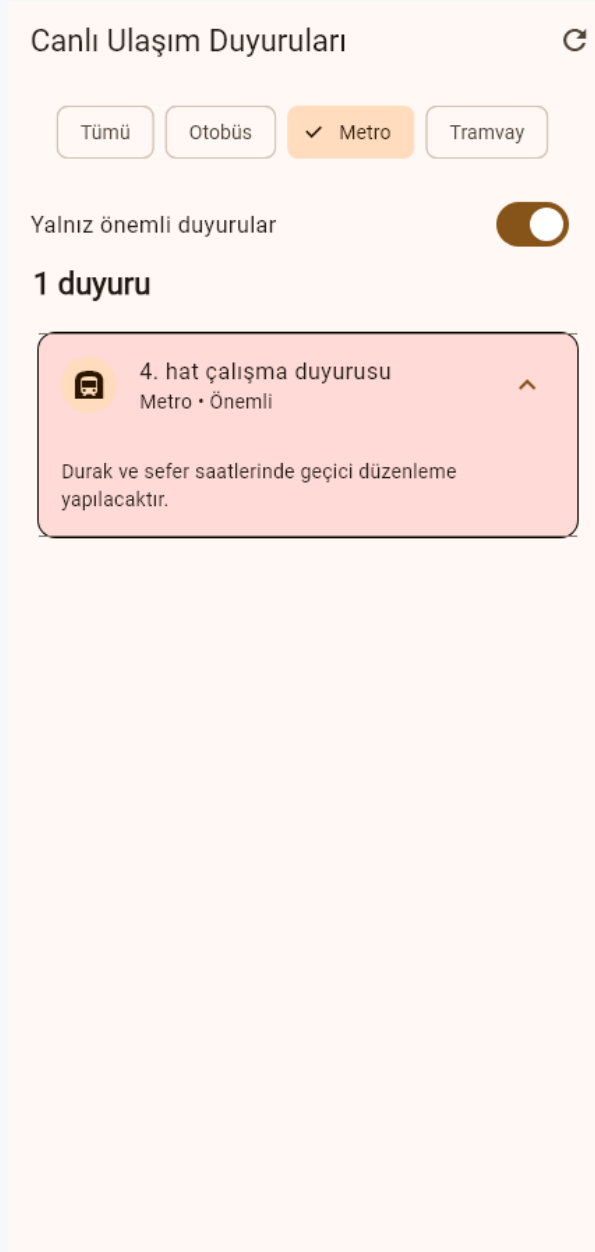
```
class Duyuru {  
  const Duyuru({  
    required this.id,  
    required this.hat,  
    required this.baslik,  
    required this.detay,  
    required this.acil,  
  });  
  
  final int id;  
  final String hat;  
  final String baslik;  
  final String detay;  
  final bool acil;  
}
```

İki filtreyi birlikte uygulama

```
List<Duyuru> get gorunen => (tum ?? []).where((duyuru) {  
  final hatUygun = hat == 'Tümü' || duyuru.hat == hat;  
  final onemUygun = !sadeceAcil || duyuru.acil;  
  return hatUygun && onemUygun;  
}).toList();
```

Ayrıntıyı aynı kartta açma

ExpansionTile, başlık ve hat bilgisini özet hâlinde gösterir. Kullanıcı karta dokunduğunda uzak JSON'dan gelen tam açıklama açılır.



Toplu Taşıma Duyurusu gerçek Flutter ekranı

Liste ekranı durumları

- `tum == null`: yükleniyor göstergesi

- hata != null: hata ve yeniden deneme
- tum boş: API başarılı fakat içerik yok
- tum dolu, gorunen boş: arama/filtre sonucu yok
- gorunen dolu: liste kartları

API'nin boş liste döndürmesi ile bağlantı hatası aynı durum değildir; kullanıcı mesajları da farklı olmalıdır.

Uygulama görevleri

Kolay — Fotoğraf başlıkları

Bir fotoğraf API'sinden başlık listesini alın. Arama ve yeniden yükleme ekleyin.

Orta — Okul etkinlikleri

Etkinlikleri kategoriye ve tarihe göre filtreleyin. Kart seçildiğinde tam açıklamayı alt sayfada gösterin.

İleri — Çok sayfalı ürün kataloğu

API'den sayfa sayfa veri alın. Listenin sonuna yaklaşıncaya yeni sayfayı yükleyin; ilk yükleme, sayfa sonu yüklemesi, hata, yeniden deneme ve yinelenen kimlik durumlarını yönetin.

Test ve kontrol

- JSON dizisindeki bütün öğeler modele dönüşmeli.
- Arama büyük/küçük harften etkilenmemeli.
- Birden fazla filtre AND mantığıyla birlikte çalışmalı.
- Filtre kapatılınca asıl liste geri gelmeli.
- Boş filtre sonucu açıklayıcı mesaj göstermeli.
- Yenileme yeni HTTP isteği göndermeli.
- Kart ayrıntısı kesilmiş değil, tam metni göstermeli.

Geri bildirim ve çözüm erişimi

Öğrenci uzak liste ekranını gönderdikten sonra İpucunu göster modelleme, türetilmiş liste ve durum akışını açıklar. Örnek çözümü göster gerçek HTTP servis ve widget kodlarını; indirme bağlantısı ise çalışan projeyi çevrim dışı inceleme için sunar.

Kaynak projeler

- Konu uygulaması: 07 - ■nternet ve Uzak Veri/Uygulamalar/Uygulama-27-Gönderi Ak■■■
- Özgün uygulama: 07 - ■nternet ve Uzak Veri/Uygulamalar/Ozgun-27-Toplu Ta■■■ma Duyurusu