

FLUTTER

Uzak Veride CRUD Mantığı

28. 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
28	Uzak Veride CRUD Mantığı	Görev Servisi • Okul Talep Sistemi

28. Hafta — Uzak Veride CRUD Mantiğı

Bu hafta ne yapacağız?

Yerel SQLite CRUD işlemlerini REST servislerine taşıyacağız. Listeleme için GET, oluşturma için POST, kısmi güncelleme için PATCH ve silme için DELETE kullanacağız. Her istekte URI, JSON gövdesi, durum kodu, kullanıcı geri bildirimi ve başarısızlıkta arayüzü geri alma davranışını ele alacağız.

Kazanımlar

- CRUD işlemleriyle HTTP yöntemlerini eşleştirme
- JSON istek gövdesi ve content - type başlığı gönderme
- Yeni kaydın sunucudan dönen kimliğini kullanma
- Kaynağa özel /resource/{id} URI'si oluşturma
- İyimser güncelleme ve hata durumunda geri alma
- 200, 201 ve 204 gibi başarılı durum kodlarını işleme
- Testte HTTP yöntemi, URI ve JSON gövdesini doğrulama

REST eşlemesi

İşlem	HTTP	Örnek URI
Listele	GET	/todos
Oluştur	POST	/todos
Kısmi güncelle	PATCH	/todos/201
Tam değiştir	PUT	/todos/201
Sil	DELETE	/todos/201

Çözümlü uygulama — Uzak Görev Servisi

GET — Görevleri listeleme

```
final yanıt = await client.get(
  temel.replace(queryParameters: {'_limit': '4'}),
);

kontrol(yanıt);
final data = jsonDecode(yanıt.body) as List;
return data
  .map((e) => Gorev.fromJson(Map<String, dynamic>.from(e as Map)))
  .toList();
```

POST — Yeni görev oluşturma

```
final yanıt = await client.post(
  temel,
  headers: {'content-type': 'application/json; charset=utf-8'},
  body: jsonEncode({
    'title': baslik,
    'completed': false,
    'userId': 1,
  })),
);

kontrol(yanıt, beklenen: {200, 201});
return Gorev.fromJson(jsonDecode(yanıt.body));
```

Yeni nesnenin kimliği istemci tarafından uydurulmaz; sunucunun POST yanıtından alınır.

PATCH — Yalnız tamamlanma alanını değiştirme

```
final yanıt = await client.patch(
  temel.resolve('${temel.path}/${gorev.id}'),
  headers: {'content-type': 'application/json; charset=utf-8'},
  body: jsonEncode({'completed': gorev.tamam}),
);
```

İyimser arayüz ve geri alma

```
final eski = gorev.tamam;
setState(() => gorev.tamam = yeniDeğer);

try {
  await servis.guncelle(gorev);
} catch (e) {
  setState(() => gorev.tamam = eski);
}
```

Arayüz hemen tepki verir. Sunucu isteği başarısız olursa eski değer geri yüklenir.

DELETE — Kimlikle silme

```
final yanıt = await client.delete(
  temel.resolve('${temel.path}/${id}'),
);

kontrol(yanıt, beklenen: {200, 204});
```

Yerel listeden silme yalnız sunucu başarılı yanıt verdikten sonra yapılır.

Gerçek uygulama ekranı

Test sırasıyla GET, POST, PATCH ve DELETE isteklerini çalıştırmış; her yöntemi ve POST/PATCH JSON gövdelerini doğrulamıştır.

Uzak Görev Servisi

POST

DELETE başarılı: #1 silindi

GET • POST • PATCH • DELETE

 Flutter ödevini tamamla
Uzak kimlik: 201 

Uzak Görev Servisi gerçek Flutter ekranı

Özgün uygulama — Uzak Okul Talep Sistemi

Talep modeli başlık, açıklama ve iş akışı durumu taşır:

```
class Talep {  
  Talep({  
    required this.id,  
    required this.baslik,  
    required this.detay,  
    required this.durum,  
  });  
  
  final int id;  
  final String baslik;  
  final String detay;  
  String durum;  
}
```

Talebi sunucuya gönderme

```
body: jsonEncode({
  'title': baslik,
  'body': detay,
  'status': 'Açık',
  'userId': 1,
})
```

İş akışını ilerletme

```
talep.durum = talep.durum == 'Açık'
  ? 'İnceleniyor'
  : 'Çözüldü';

await client.patch(
  talepUri,
  body: jsonEncode({'status': talep.durum}),
);
```

İstek başarısız olursa önceki durum geri yüklenir. Bu, ekrandaki bilginin sunucuyla çelişmesini önler.

Uzak Okul Talep Sistemi

Talep başlığı

Açıklama

Talebi POST ile gönder

Talep silindi

Laboratuvar isteği

İnceleniyor

Bilgisayar faresinin yenilenmesi

Durumu ilerlet

Uzak Okul Talep Sistemi gerçek Flutter ekranı

Durum kodu stratejisi

- GET: genellikle 200
- POST: çoğunlukla 201, bazı servislerde 200
- PATCH/PUT: 200 veya gövdesiz 204
- DELETE: 200 veya 204
- 400: doğrulama hatası
- 401/403: kimlik doğrulama/yetki sorunu
- 404: kaynak bulunamadı
- 409: çakışma
- 500+: sunucu hatası

Başarılı kabul edilen kodlar servis sözleşmesine göre açıkça tanımlanmalıdır.

Üretim ortamında dikkat

- Yetkilendirme anahtarı kod içine yazılmaz; güvenli yapılandırmadan alınır.
- Hassas veri yalnız HTTPS üzerinden gönderilir.
- Tekrarlanan POST isteklerinde çift kayıt riskine karşı idempotency anahtarı düşünülebilir.
- Silme öncesi kullanıcı onayı ve gerekirse geri alma sunulur.
- Ağ kesintisinde çevrim dışı kuyruk kullanılacaksa çakışma politikası belirlenir.
- JSON alanları ve sunucu hata gövdesi güvenli biçimde doğrulanır.

Uygulama görevleri

Kolay — Uzak not servisi

Notları GET ile listeleyin, POST ile yeni not gönderin ve DELETE ile kaldırın.

Orta — Kulüp başvuru sistemi

Başvuru oluşturun; durumu PATCH ile “Bekliyor”, “Onaylandı” ve “Reddedildi” arasında değiştirin. Hata durumunda eski durumu geri alın.

İleri — Çevrim dışı destekli talep sistemi

CRUD isteklerini servis katmanına ayırın. İnternet yokken işlemleri yerel kuyruğa yazın; bağlantı geldiğinde sırayla gönderin. Sunucu çakışmalarını, çift POST'u ve başarısız DELETE'i kullanıcıya açıklayın.

Test ve kontrol

- GET doğru koleksiyon URI'sine gitmeli.
- POST gövdesi zorunlu alanları içermeli.
- Sunucunun döndürdüğü kimlik modele aktarılmalı.
- PATCH yalnız değişen alanı göndermeli.

- DELETE doğru /id URI'sine gitmeli.
- Başarısız güncellemede arayüz eski değere dönmeli.
- 200/201/204 sözleşmeye uygun biçimde kabul edilmeli.
- Test HTTP yöntemlerinin sırasını doğrulamalı.

Geri bildirim ve çözüm erişimi

Öğrenci uzak CRUD servisini gönderdikten sonra İpucunu göster yöntem–URI–gövde ilişkisini açıklar. Örnek çözümü göster gerçek servis, model ve ekran kodlarını; indirme bağlantısı ise çalışan Flutter projesini sunar.

Kaynak projeler

- Konu uygulaması: 07 - ■nternet ve Uzak Veri/Uygulamalar/Uygulama-28-Görev Servisi
- Özgün uygulama: 07 - ■nternet ve Uzak Veri/Uygulamalar/Ozgun-28-Okul Talep Sistemi