

Flutter ile Uygulama Geliştirme Kursu | Android & IOS

Flutter Depolama İşlemleri

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Android - IOS Developer and Trainer

Eğitim İçeriği

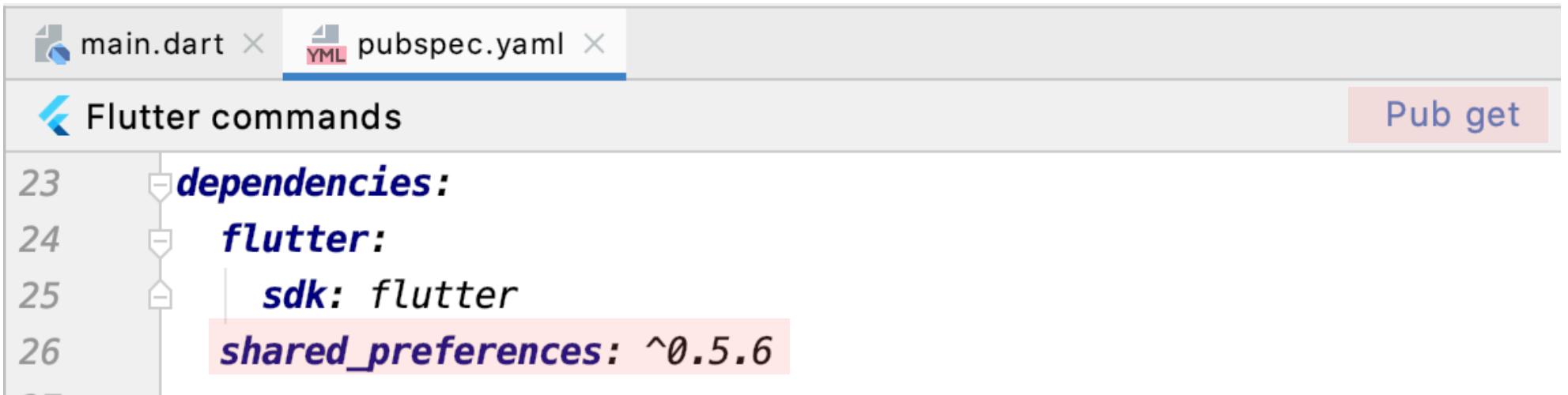
- Shared Preferences
- Dosya İşlemleri
- SQLite
- FutureBuilder ile SQLite Çalışması

Shared Preferences

Shared Preferences

- Key - Value ilişkisi ile basit verileri kalıcı olarak depolayabiliriz.
- Kullanılan veri türleri : *String,int,double,bool,List<String>*
- Uygulama silindiğinde veriler silinmektedir.
- Veritabanı üzerindeki gibi detaylı verileri kayıt etmiceksek hızlı bir çözüm için kullanılabilir.

Shared Preferences Kurulum



The screenshot shows a code editor interface with two tabs at the top: 'main.dart' and 'pubspec.yaml'. The 'pubspec.yaml' tab is active, indicated by a blue underline. Below the tabs is a toolbar with 'Flutter commands' and a 'Pub get' button. The main content area displays the following YAML code:

```
23 dependencies:  
24   flutter:  
25     |   sdk: flutter  
26     |   shared_preferences: ^0.5.6  
27  
shared_preferences: ^2.0.11
```

The line 'shared_preferences: ^0.5.6' is highlighted with a pink rectangle. At the bottom of the code area, the text 'shared_preferences: ^2.0.11' is displayed, likely indicating the current version of the dependency.

```
Future<void> test() async {
  var sp = await SharedPreferences.getInstance();
  //Veri Kaydet
  sp.setString("ad", "ahmet");
  sp.setInt("yas", 23);
  sp.setDouble("boy", 1.78);
  sp.setBool("bekar", true);

  var arkadasListe = <String>[];
  arkadasListe.add("Ece");
  arkadasListe.add("Ali");

  sp.setStringList("arkadasListe", arkadasListe);

  //Veri Silme
  sp.remove("ad");
}

//Veri Okuma
String gelenAd = sp.getString("ad") ?? "isim yok";
int gelenYas = sp.getInt("yas") ?? 0;
double gelenBoy = sp.getDouble("boy") ?? 0.0;
bool gelenBekar = sp.getBool("bekar") ?? false;
var gelenArkadasliste = sp.getStringList("arkadasListe") ?? null;

print("Gelen Ad : $gelenAd");
print("Gelen Yaş : $gelenYas");
print("Gelen Boy : $gelenBoy");
print("Gelen Bekar : $gelenBekar");

for(var a in gelenArkadasliste){
  print("Arkadaş : $a");
}
}
```

Veri kayıt etmek

```
Future<void> veriKaydi() async {  
  
    var sp = await SharedPreferences.getInstance();  
  
    sp.setString("ad", "ahmet");  
    sp.setInt("yas", 18);  
    sp.setDouble("boy", 1.78);  
    sp.setBool("bekarMi", true);  
  
    var arkadasListe = <String>[];  
    arkadasListe.add("Ece");  
    arkadasListe.add("Ali");  
  
    sp.setStringList("arkadasListe", arkadasListe);  
}
```

Veri Okumak

```
Future<void> veriOku() async {  
  
    var sp = await SharedPreferences.getInstance();  
  
    String ad = sp.getString("ad") ?? "isim yok" ;  
    int yas = sp.getInt("yas") ?? 99;  
    double boy = sp.getDouble("boy") ?? 9.99;  
    bool bekarMi = sp.getBool("bekarMi") ?? false;  
  
    var arkadasListe = sp.getStringList("arkadasListe") ?? null;  
  
    print("Ad : $ad");  
    print("Yaş : $yas");  
    print("Boy : $boy");  
    print("Bekar mı ? : $bekarMi");  
  
    for(var a in arkadasListe!){  
        print("Arkadaş : $a");  
    }  
}
```

Veri Silme - Güncelleme

```
Future<void> veriSil() async {  
    var sp = await SharedPreferences.getInstance();  
    sp.remove("ad");  
}
```

```
Future<void> veriGuncelleme() async {  
    var sp = await SharedPreferences.getInstance();  
    sp.setInt("yas", 99);  
}
```

Uygulama : Açılış Sayısı

12:16 ⚡ ⓘ

SP Sayaç Uygulaması

Açılış Sayısı : 2

```
class _MyHomePageState extends State<MyHomePage> {

    int sayac = 0;

    @override
    void initState() {
        super.initState();
        sayacKontrol();
    }

    Future<void> sayacKontrol() async {

        var sp = await SharedPreferences.getInstance();

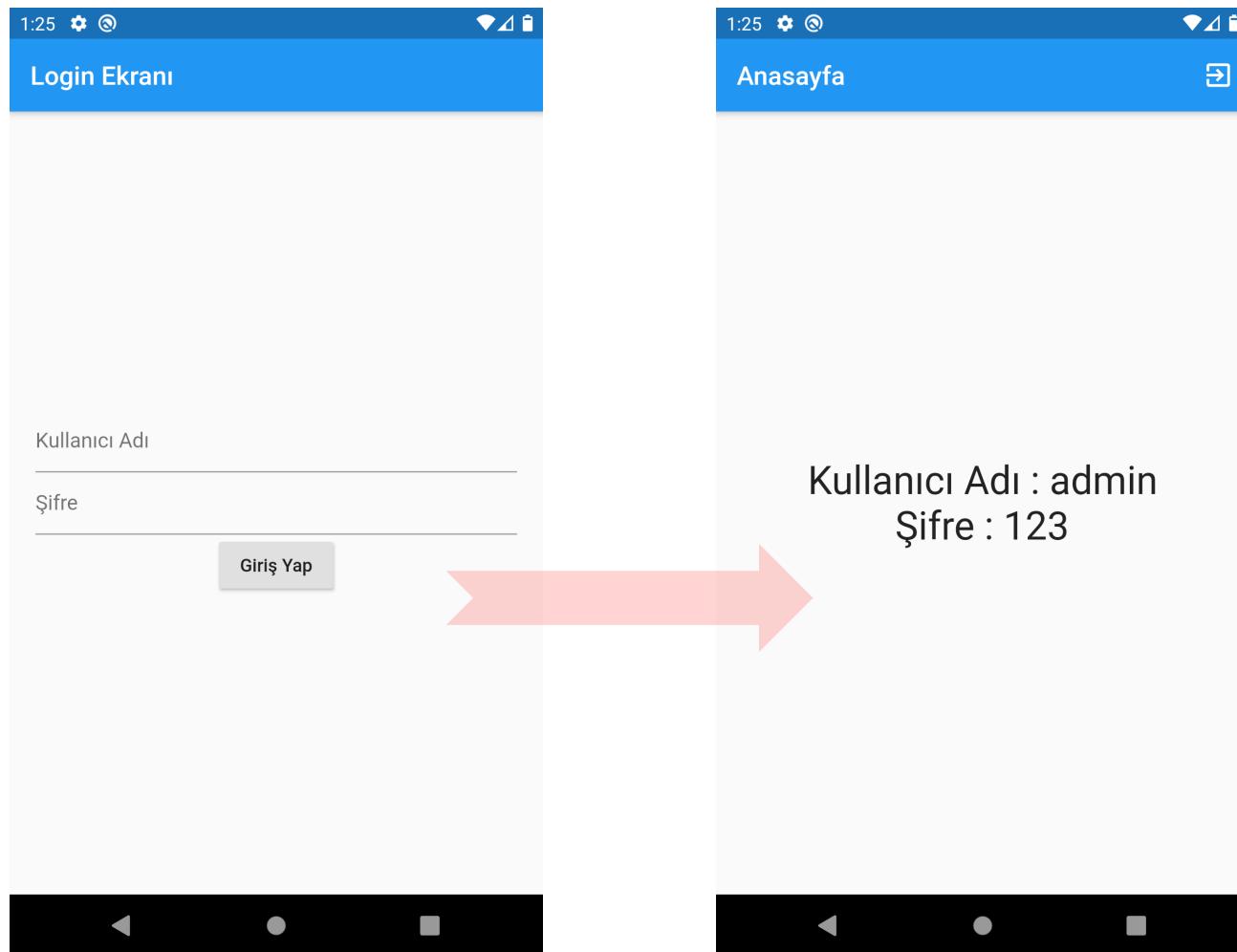
        //En son değeri okuma
        sayac = sp.getInt("sayac") ?? 0;

        //En son değeri bir artırma aynı zamanda
        //setState metodu ile arayüzde güncel halinin görünmesini sağlama
        setState(() {
            sayac+=1 ;
        });

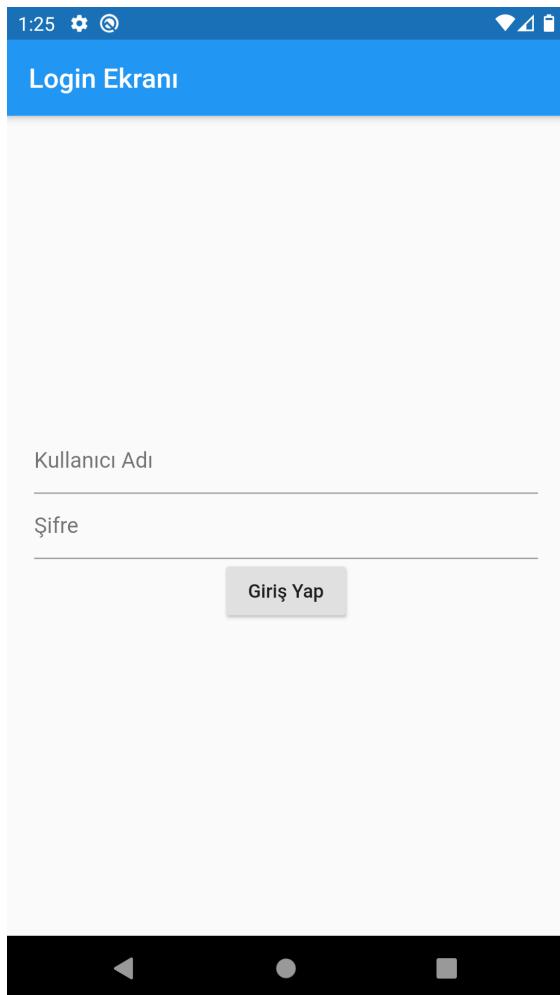
        //En son değeri tekrar okumak için kayıt etme
        sp.setInt("sayac", sayac);
    }
}
```

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text(widget.title),
        ), // AppBar
        body: Center(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    Text("Açılış Sayısı : $sayac",style: TextStyle(fontSize: 50),),
                ], // <Widget>[]
        ), // Column
    ), // Center
); // Scaffold
}
```

Uygulama : Login Ekran



Login Ekranı



```
class LoginEkranı extends StatefulWidget {
  @override
  _LoginEkranıState createState() => _LoginEkranıState();
}

class _LoginEkranıState extends State<LoginEkranı> {

  var tfKullaniciAdi = TextEditingController();
  var tfSifre = TextEditingController();

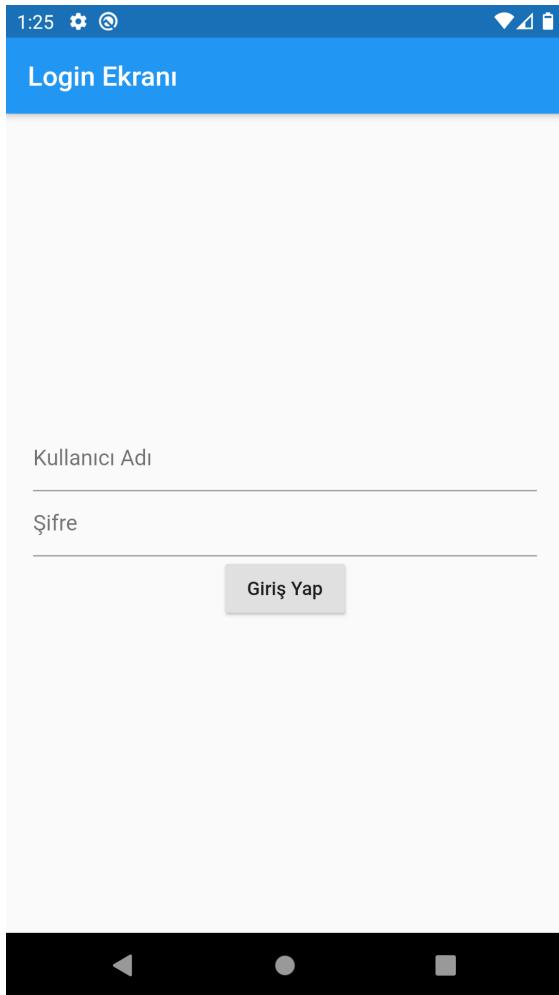
  Future<void> girisKontrol() async {
    var ka = tfKullaniciAdi.text;
    var s = tfSifre.text;

    if(ka == "admin" && s == "123"){
      var sp = await SharedPreferences.getInstance();

      sp.setString("kullaniciAdi", ka);
      sp.setString("sifre", s);

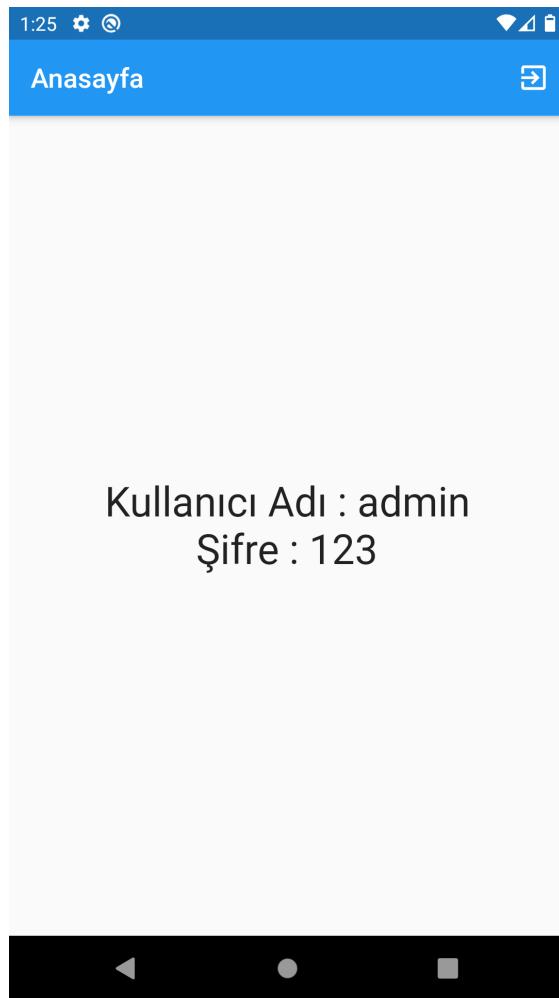
      Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => Anasayfa()));
    }else{
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text("Giriş Hatalı")));
    }
  }
}
```

Login Ekranı



```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text("Login Ekranı"),  
    ), // AppBar  
    body: Center(  
      child: Padding(  
        padding: const EdgeInsets.all(8.0),  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: <Widget>[  
            TextField(  
              controller: tfKullaniciAdi,  
              decoration: InputDecoration(  
                hintText: "Kullanıcı Adı",  
              ), // InputDecoration  
            ), // TextField  
            TextField(  
              obscureText: true,  
              controller: tfSifre,  
              decoration: InputDecoration(  
                hintText: "Şifre",  
              ), // InputDecoration  
            ), // TextField  
            ElevatedButton(  
              child: Text("Giriş Yap"),  
              onPressed: (){  
                girisKontrol();  
              },  
            ), // ElevatedButton
```

Anasayfa



```
class _AnasayfaState extends State<Anasayfa> {
    late String spKullaniciAdi;
    late String spSifre;

    Future<void> oturumBilgisiOku() async {
        var sp = await SharedPreferences.getInstance();
        setState(() {
            spKullaniciAdi = sp.getString("kullaniciAdi") ?? "kullanıcı adı yok";
            spSifre = sp.getString("sifre") ?? "şifre yok";
        });
    }

    Future<void> cikisYap() async {
        var sp = await SharedPreferences.getInstance();
        sp.remove("kullaniciAdi");
        sp.remove("sifre");
        Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => LoginEkranı()));
    }

    @override
    void initState() {
        super.initState();
        oturumBilgisiOku();
    }
}
```

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Anasayfa"),
            actions: [
                IconButton(
                    icon: Icon(Icons.exit_to_app),
                    onPressed: () {
                        cikisYap();
                    },
                ),
            ],
        ),
        body: Center(
            child: Padding(
                padding: const EdgeInsets.all(8.0),
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                        Text("Kullanıcı Adı : $spKullaniciAdi", style: TextStyle(fontSize: 30)),
                        Text("Şifre : $spSifre", style: TextStyle(fontSize: 30)),
                    ],
                ),
            ),
        ),
    );
}
```

İlk Açılan Sayfanın Otomatik Belirlenmesi

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

class MyApp extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ), // ThemeData
    home: FutureBuilder<bool>(
      future: oturumKontrol(),//  

      builder: (context,snapshot) {
        if (snapshot.hasData) {
          bool gecisIzni = snapshot.data!; //Oturum açıksa true değilse false gelir
          return gecisIzni ? Anasayfa() : LoginEkranı();//Duruma göre ilk sayfa seçilir.
        }
        return Container();//Bekleme sırasında gösterilecek ekran ,anlıktır kısa süre.
        //İstenirse splash screen kullanılabilir.
      },
    ), // FutureBuilder
  ); // MaterialApp
}
```

```
Future<bool> oturumKontrol() async {
  var sp = await SharedPreferences.getInstance();

  String spKullaniciAdi = sp.getString("kullaniciAdi") ?? "kullanıcı adı yok";
  String spSifre = sp.getString("sifre") ?? "şifre yok";

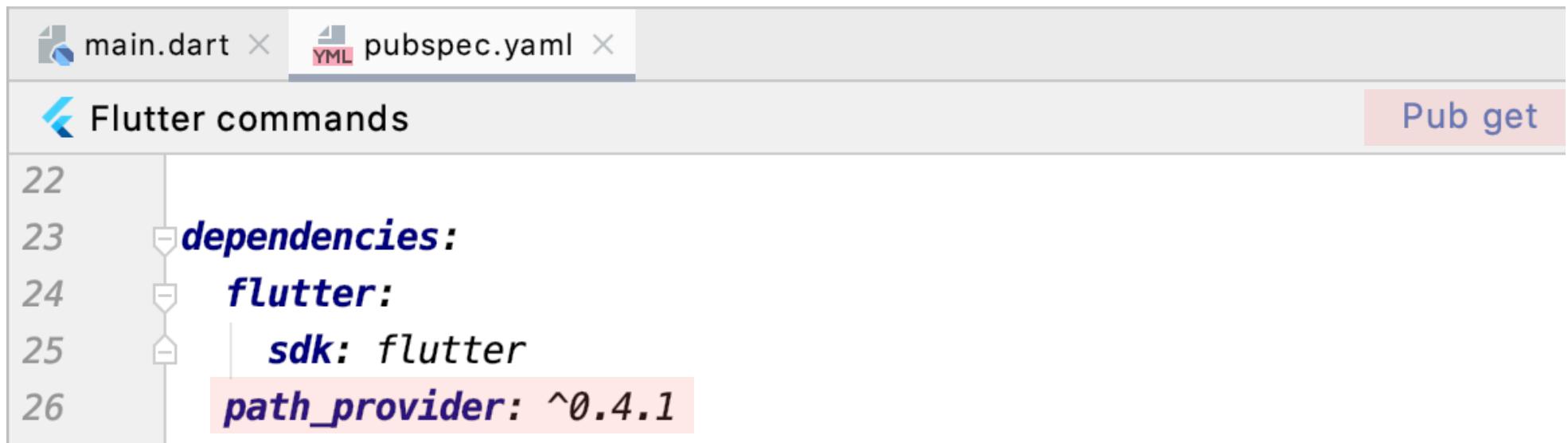
  if(spKullaniciAdi == "admin" && spSifre == "123"){
    return true;
  }else{
    return false;
  }
}
```

Dosya İşlemleri

Dosya İşlemleri

- *path_provider* yapısı sayesinde dosya işlemleri yapabiliriz.
- Örneğin dosya sistemine txt uzantılı dosya kayıt edip içine veriler yazabiliriz.
 - Kalıcı depolama sağlamaktadır.

path_provider Kurulum



The screenshot shows a code editor interface with two tabs at the top: 'main.dart' and 'pubspec.yaml'. The 'pubspec.yaml' tab is active, indicated by a blue underline. Below the tabs is a toolbar with a 'Flutter commands' button and a 'Pub get' button. The main content area displays the following YAML code:

```
22
23 dependencies:
24   flutter:
25     sdk: flutter
26     path_provider: ^0.4.1
```

The line 'path_provider: ^0.4.1' is highlighted with a pink rectangular background.

Veri kayıt etmek

```
Future<void> veriYaz() async{
    var ad = await getApplicationDocumentsDirectory(); //Dosya sistemine erişim

    var uygulamaDosyalamaYolu = await ad.path;

    var dosya = File('$uygulamaDosyalamaYolu/dosyam.txt'); //Dosyaya erişim

    dosya.writeAsString(tfgirdi.text); //Yazma işlemi

    tfgirdi.text = ""; //Yazma işleminden sonra textfield alanını temizleme.

}
```

Veri Okumak

```
Future<void> veriOku() async {
    try{
        var ad = await getApplicationDocumentsDirectory();

        var uygulamaDosyalamaYolu = await ad.path;

        var dosya = File("$uygulamaDosyalamaYolu/dosyam.txt");

        String okunanVeri = await dosya.readAsString();

        tfgirdi.text = okunanVeri;

    }catch(e){
        e.toString();
    }
}
```

Veri Silme

```
Future<void> veriSil() async{
    var ad = await getApplicationDocumentsDirectory();

    var uygulamaDosyalamaYolu = await ad.path;

    var dosya = File('$uygulamaDosyalamaYolu/dosyam.txt');

    if(dosya.existsSync()){//Silmek istediğimiz dosya var mı yok mu kontrolü
        dosya.delete();//Silme işlemi
    }
}
```

Uygulama : Dosya İşlemleri



```
class _MyHomePageState extends State<MyHomePage> {
  var tfgirdi = TextEditingController();

  Future<void> veriYaz() async {
    var ad = await getApplicationDocumentsDirectory();
    var uygulamaDosyalamaYolu = await ad.path;
    var dosya = File("$uygulamaDosyalamaYolu/dosyam.txt");
    dosya.writeAsString(tfgirdi.text);
    tfgirdi.text = "";
  }

  Future<void> veriOku() async {
    try{
      var ad = await getApplicationDocumentsDirectory();
      var uygulamaDosyalamaYolu = await ad.path;
      var dosya = File("$uygulamaDosyalamaYolu/dosyam.txt");
      String okunanVeri = await dosya.readAsString();
      tfgirdi.text = okunanVeri;
    }catch(e){
      e.toString();
    }
  }

  Future<void> veriSil() async {
    var ad = await getApplicationDocumentsDirectory();
    var uygulamaDosyalamaYolu = await ad.path;
    var dosya = File("$uygulamaDosyalamaYolu/dosyam.txt");
    if(dosya.existsSync()){
      dosya.delete();
    }
  }
}
```

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



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ), // AppBar
    body: Center(
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            TextField(
              controller: tfgirdi,
              decoration: InputDecoration(
                hintText: "Veri giriniz",
              ), // InputDecoration
            ), // TextField
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: <Widget>[
                ElevatedButton(
                  child: Text("Yaz"),
                  onPressed: (){
                    veriYaz();
                  },
                ), // ElevatedButton
                ElevatedButton(
                  child: Text("Oku"),
                  onPressed: (){
                    veriOku();
                  },
                ), // ElevatedButton
              ],
            ),
          ],
        ),
      ),
    ),
  );
}
```

```
}, // Row
], // <Widget>[]
), // Column
), // Padding
), // Center
); // Scaffold
}
```

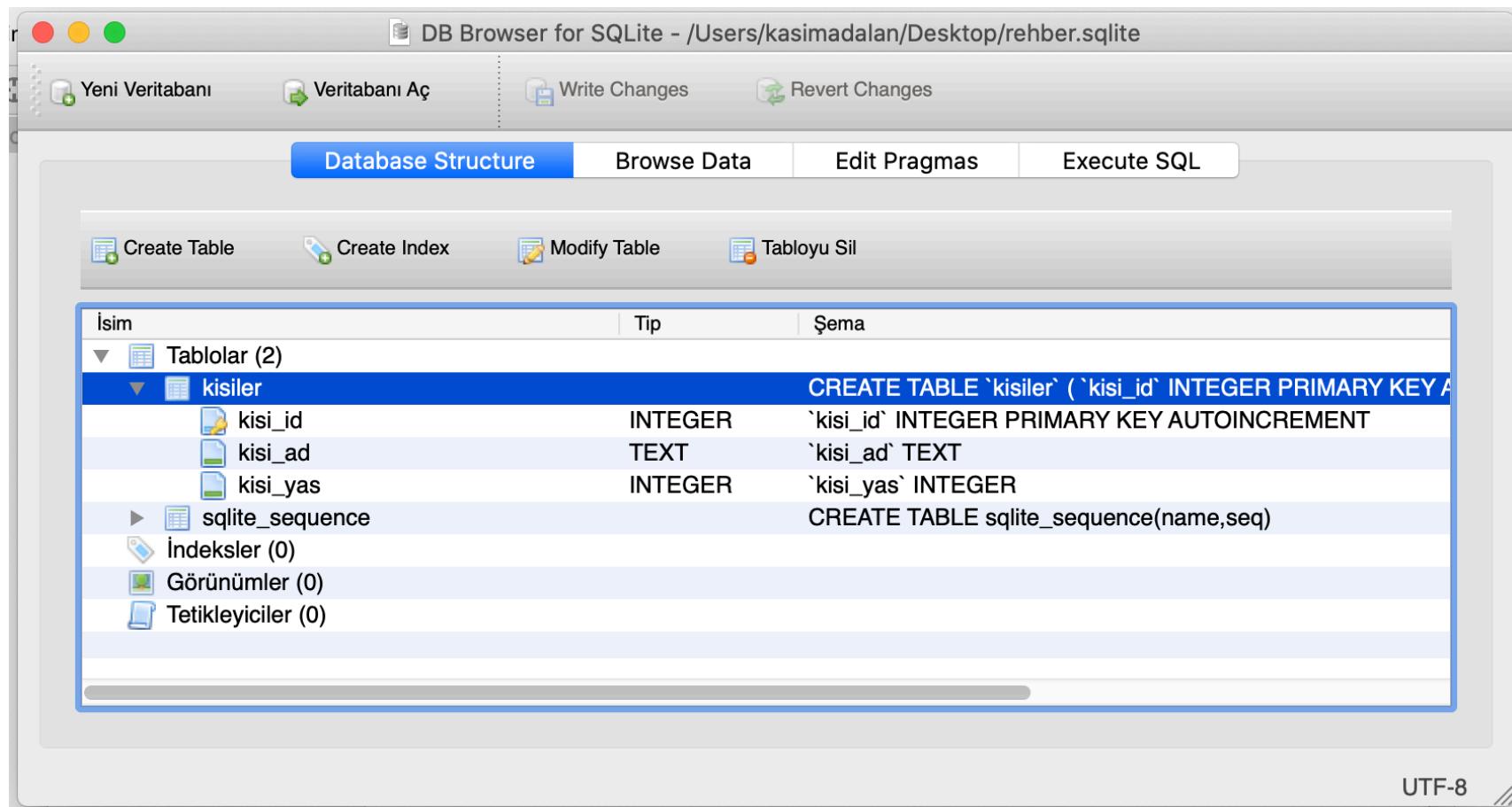
SQLite

SQLLite

- Sqlite kullanımı oldukça kolay bir veri tabanı sistemidir.
- Küçük boyutu ve hızlı yapısı sayesinde mobil cihazlarda kullanımı oldukça yaygındır.
- Sqlite birçok programlama diliyle birlikte kullanılabilir.
- Android ve IOS işletim sistemlerinde çalışabilmektedir.
- Bu iki platformda çalışabilmesi ortak veri tabanı oluşturmada kolaylık sağlamaktadır.
- MySql ve Sql veri tabanlarında kullanılan sql sorgu cümleleri ile uyumludur.

1

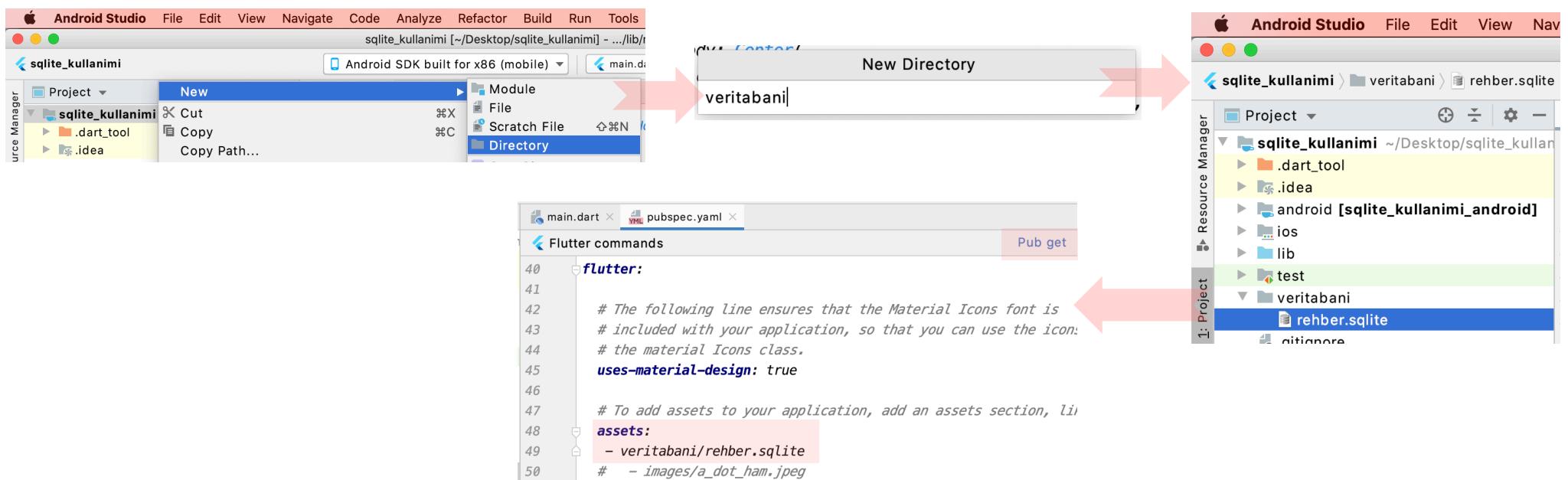
DB Browser ile Veri tabanı Oluşturma



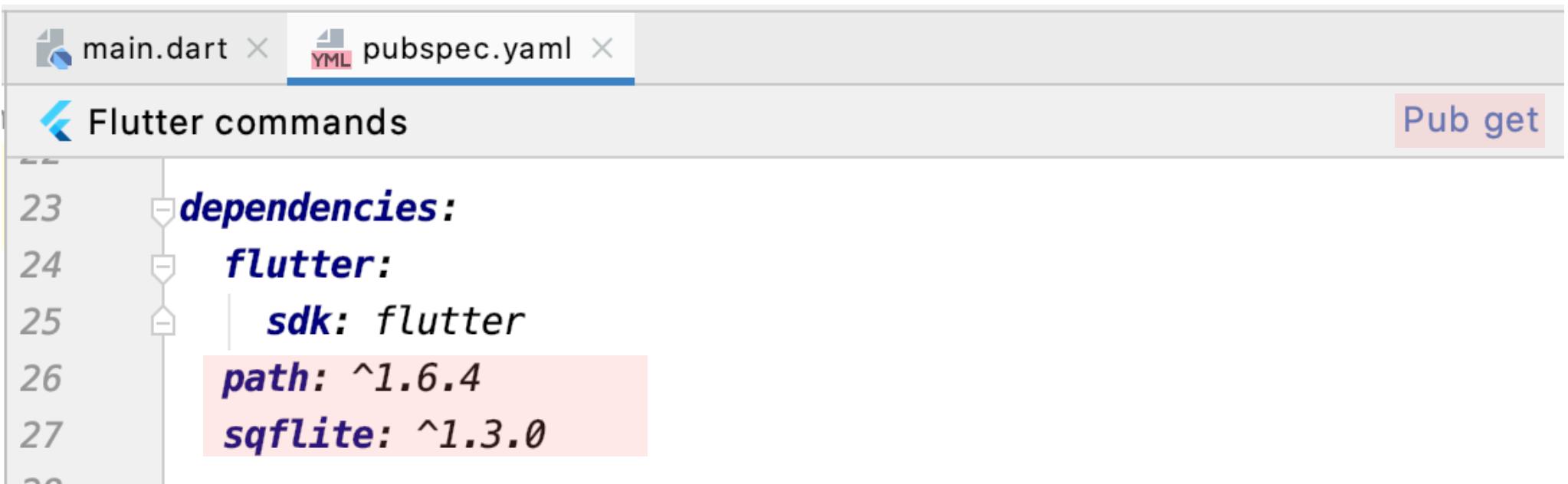
2

Veri tabanını Android Studio üzerinde ekleme

- Android studio üzerinde veritabanı için dosya oluşturmamız gereklidir.
- Dosya içine çalışmak istediğimiz veritabanını yerleştiririz.
- Daha sonra *pubspec.yaml* dosya içinde asset olarak veritabanını belirtmemiz gereklidir.



Kurulum



```
main.dart × pubspec.yaml ×
Flutter commands Pub get
-- 
23   dependencies:
24     flutter:
25       |   sdk: flutter
26       |   path: ^1.6.4
27       |   sqflite: ^1.3.0
28
```

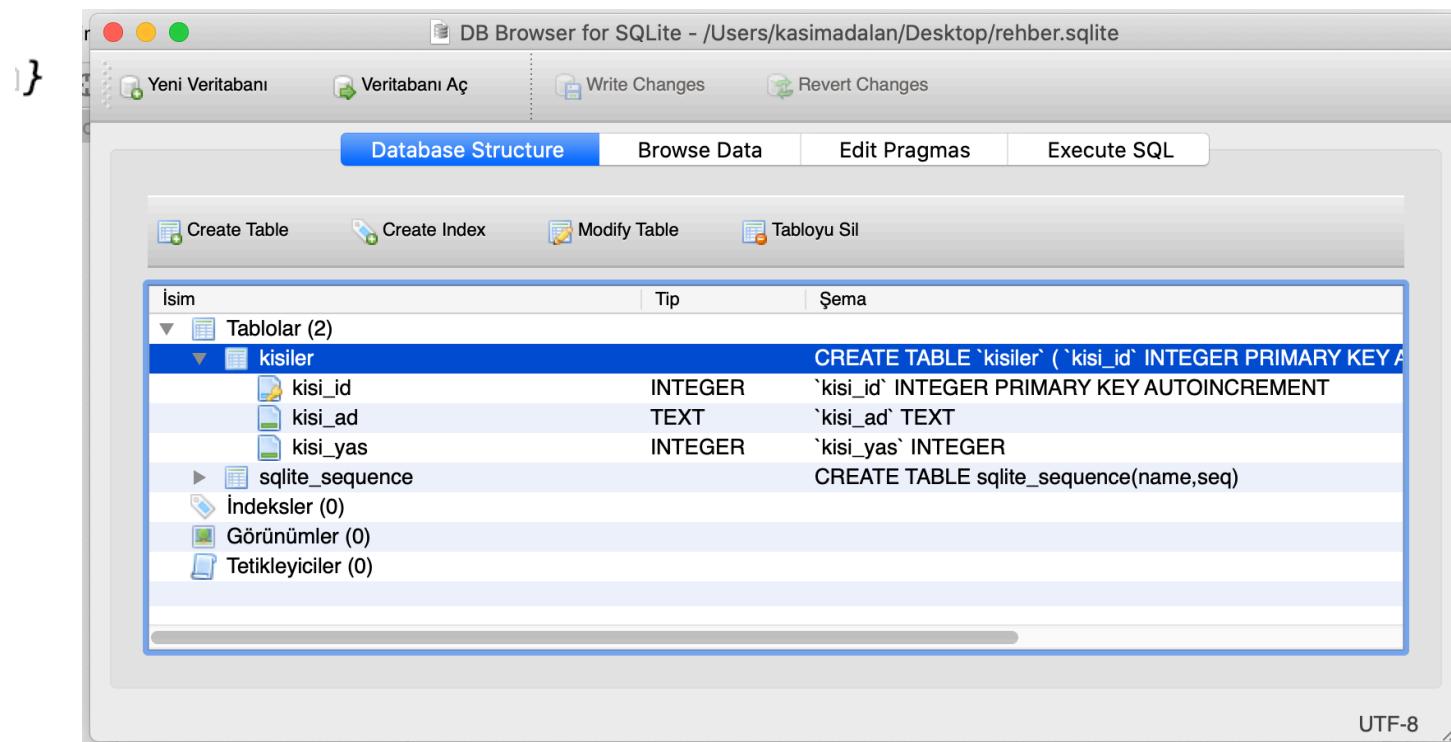
path : dosyalama işlemleri için kullanılacaktır.Kopyalama gibi işlemler için gereklidir.

4

Veri tabanını Dart sınıfları ile modelleme

```
1 class Kisiler{  
2     int kisi_id;  
3     String kisi_ad;  
4     int kisi_yas;
```

```
Kisiler(this.kisi_id, this.kisi_ad, this.kisi_yas);
```



Veri tabanı Kopyalama ve Erişim

```

class VeritabaniYardimcisi{

    static final String veritabaniAdi = "rehber.sqlite";

    static Future<Database> veritabaniErisim() async {
        String veritabaniYolu = join(await getDatabasesPath(), veritabaniAdi);

        if(await databaseExists(veritabaniYolu)){//Veritabanı var mı yok mu kontrolü
            print("Veri tabanı zaten var.Kopyalamaya gerek yok");
        }else{
            //assetten veritabanının alınması
            ByteData data = await rootBundle.load("veritabani/$veritabaniAdi");
            //Veritabanının kopyalama için byte dönüşümü
            List<int> bytes = data.buffer.asUint8List(data.offsetInBytes, data.lengthInBytes);
            //Veritabanının kopyalanması.
            await File(veritabaniYolu).writeAsBytes(bytes, flush: true);
            print("Veri tabanı kopyalandı");
        }
        //Veritabanını açıyoruz.
        return openDatabase(veritabaniYolu);
    }
}

```

Uygulama ilk açıldığında veritabanı kopyalanır, daha sonra uygulama her açıldığında direkt erişim işlemi gerçekleşir.

```

import 'dart:io';
import 'package:flutter/services.dart';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';

```

Kullanımı

```

class _MyHomePageState extends State<MyHomePage> {

    @override
    void initState() {
        super.initState();

        kisileriGoster();
    }

    Future<void> kisileriGoster() async{
        var liste = await Kisilerdao().tumKisiler();

        for(Kisiler k in liste){
            print("*****");
            print("Kisi id : ${k.kisi_id}");
            print("Kisi ad : ${k.kisi_ad}");
            print("Kisi yaşı : ${k.kisi_yas}");
        }
    }
}

```

Tablo Üzerinde İşlem Yapmak için dao sınıfı

```
class Kisilerdao{  
  
    Future<List<Kisiler>> tumKisiler() async {  
        //Veritabanına erişim           Veritabanı erişimi ve kopyalama  
        var db = await VeritabaniYardimcisi.veritabaniErisim();  
        //sql sorgu  
        List<Map<String, dynamic>> maps = await db.rawQuery("SELECT * FROM kisiler");  
  
        return List.generate(maps.length, (i) {  
            //Gelen satırları listeye dönüştürme  
  
            var satir = maps[i];  
            //Her satırın içindeki verileri isimleriyle alarak nesneye dönüştürülür.  
            return Kisiler(satir["kisi_id"], satir["kisi_ad"],satir["kisi_yas"]);  
        }); // List.generate  
    }  
}
```

Veri Okuma

```
Future<List<Kisiler>> tumKisiler() async {
    //Veritabanına erişim
    var db = await VeritabaniYardimcisi.veritabaniErisim();
    //sql sorgu
    List<Map<String, dynamic>> maps = await db.rawQuery("SELECT * FROM kisiler");
    return List.generate(maps.length, (i) { //Gelen satırları listeye dönüştürme
        var satir = maps[i];
        //Her satırın içindeki verileri isimleriyle alarak nesneye dönüştürülür.
        return Kisiler(satir["kisi_id"], satir["kisi_ad"], satir["kisi_yas"]);
    });
}
```

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

Kullanım

```
class _MyHomePageState extends State<MyHomePage> {

    @override
    void initState() {
        super.initState();

        kisileriGoster();
    }

    Future<void> kisileriGoster() async{
        var liste = await Kisilerdao().tumKisiler();

        for(Kisiler k in liste){
            print("*****");
            print("Kisi id : ${k.kisi_id}");
            print("Kisi ad : ${k.kisi_ad}");
            print("Kisi yaşı : ${k.kisi_yas}");
        }
    }
}
```

Veri Kaydı

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

```
Future<void> kisiEkle(String kisi_ad,int kisi_yas) async {  
  
    var db = await VeritabaniYardimcisi.veritabaniErisim();  
  
    //Kayıt için tablo alan adlarına veri ekliyoruz.  
    var bilgiler = Map<String, dynamic>();  
    bilgiler["kisi_ad"] = kisi_ad;  
    bilgiler["kisi_yas"] = kisi_yas;  
  
    //Tablo adıyla bilgileri ekliyoruz.  
    await db.insert("kisiler", bilgiler);  
}
```

Kullanım

```
Future<void> ekle() async{  
    await Kisilerdao().kisiEkle("Sedat", 37);  
}
```

Veri Silme

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

```
Future<void> kisiSil(int kisi_id) async {  
  
    var db = await VeritabaniYardimcisi.veritabaniErisim();  
    //Tablo adı ve hangi satırı sileceksek belirterek silme işlemi yapıyoruz.  
    await db.delete("kisiler", where: "kisi_id = ?", whereArgs: [kisi_id]);  
}  
}
```

Kullanım

```
Future<void> sil() async{  
    await Kisilerdao().kisiSil(4);  
}
```

Veri Güncelleme

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

```
Future<void> kisiGuncelle(int kisi_id, String kisi_ad, int kisi_yas) async {  
  
    var db = await VeritabaniYardimcisi.veritabaniErisim();  
  
    var bilgiler = Map<String, dynamic>();  
    bilgiler["kisi_ad"] = kisi_ad;  
    bilgiler["kisi_yas"] = kisi_yas;  
  
    //Tablo adı ve hangi satırı güncelleme yapacağımızı belirterek güncelleme yapıyoruz.  
    await db.update("kisiler", bilgiler, where: "kisi_id = ?", whereArgs: [kisi_id]);  
}
```

Kullanım

```
Future<void> guncelle() async{  
    await Kisilerdao().kisiGuncelle(4, "Yeni Sedat", 99);  
}
```

Veri Tabanında Kayıt Kontrol

```
Future<int> kayitKontrol(String kisi_ad) async {
    var db = await VeritabaniYardimcisi.veritabaniErisim();
    List<Map<String, dynamic>> maps = await db.rawQuery("SELECT count(*) AS sonuc FROM kisiler WHERE kisi_ad='$kisi_ad'");
    return maps[0]["sonuc"];
}
```

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

Kullanım

```
Future<void> kayitKontrol() async{
    int sonuc = await Kisilerdao().kayitKontrol("ahmet");
    print("Veritabanındaki ahmet sayısı : $sonuc");
}
```

Bir Tane Kayıt Alma

```
Future<Kisiler> kisiGetir(int kisi_id) async {
```

```
    var db = await VeritabaniYardimcisi.veritabaniErisim();
```

```
    List<Map<String, dynamic>> maps = await db.rawQuery("SELECT * FROM kisiler WHERE kisi_id=$kisi_id");
```

```
    var satir = maps[0];
```

```
    return Kisiler(satir["kisi_id"], satir["kisi_ad"], satir["kisi_yas"]);
```

Kullanım

```
}
```

```
Future<void> getir() async{
    var kisi = await Kisilerdao().kisiGetir(1);

    print("*****KİSİ GETİR*****");
    print("Kisi id : ${kisi.kisi_id}");
    print("Kisi ad : ${kisi.kisi_ad}");
    print("Kisi yaşı : ${kisi.kisi_yas}");
}
```

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

Arama İşlemi

```
Future<List<Kisiler>> kisiArama(String aramaKelimesi) async {
```

```
    var db = await VeritabaniYardimcisi.veritabaniErisim();
```

```
    List<Map<String, dynamic>> maps = await db.rawQuery("SELECT * FROM kisiler WHERE kisi_ad like '%$aramaKelimesi%'");
```

```
    return List.generate(maps.length, (i) { //Gelen satırları listeye dönüştürme
```

```
        var satir = maps[i];
```

```
        //Her satırın içindeki verileri isimleriyle alarak nesneye dönüştürülür.
```

```
        return Kisiler(satir["kisi_id"], satir["kisi_ad"], satir["kisi_yas"]);  
                Kullanım
```

```
    }); // List.generate
```

```
}
```

```
Future<void> arama() async{  
    var liste = await Kisilerdao().kisiArama("a");  
  
    for(Kisiler k in liste){  
        print("*****");  
        print("Kisi id : ${k.kisi_id}");  
        print("Kisi ad : ${k.kisi_ad}");  
        print("Kisi yaşı : ${k.kisi_yas}");  
    }  
}
```

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

Rastgele ve Sınırlı Veri Alma

```
Future<List<Kisiler>> rasgele2kisiGetir() async {
    var db = await VeritabaniYardimcisi.veritabaniErisim();
    List<Map<String, dynamic>> maps = await db.rawQuery("SELECT * FROM kisiler ORDER BY RANDOM() LIMIT 2");

    return List.generate(maps.length, (i) { // Gelen satırları listeye dönüştürme
        var satir = maps[i];
        // Her satırın içindeki verileri isimleriyle alarak nesneye dönüştürülür.
        return Kisiler(satir["kisi_id"], satir["kisi_ad"], satir["kisi_yas"]);
    });
}
```

| kisiler | Türü |
|----------|--------|
| kisi_id | int |
| kisi_ad | String |
| kisi_yas | int |

```
Future<void> rasgeleGetir() async{
    var liste = await Kisilerdao().rasgele2kisiGetir();

    for(Kisiler k in liste){
        print("*****");
        print("Kisi id : ${k.kisi_id}");
        print("Kisi ad : ${k.kisi_ad}");
        print("Kisi yaşı : ${k.kisi_yas}");
    }
}
```

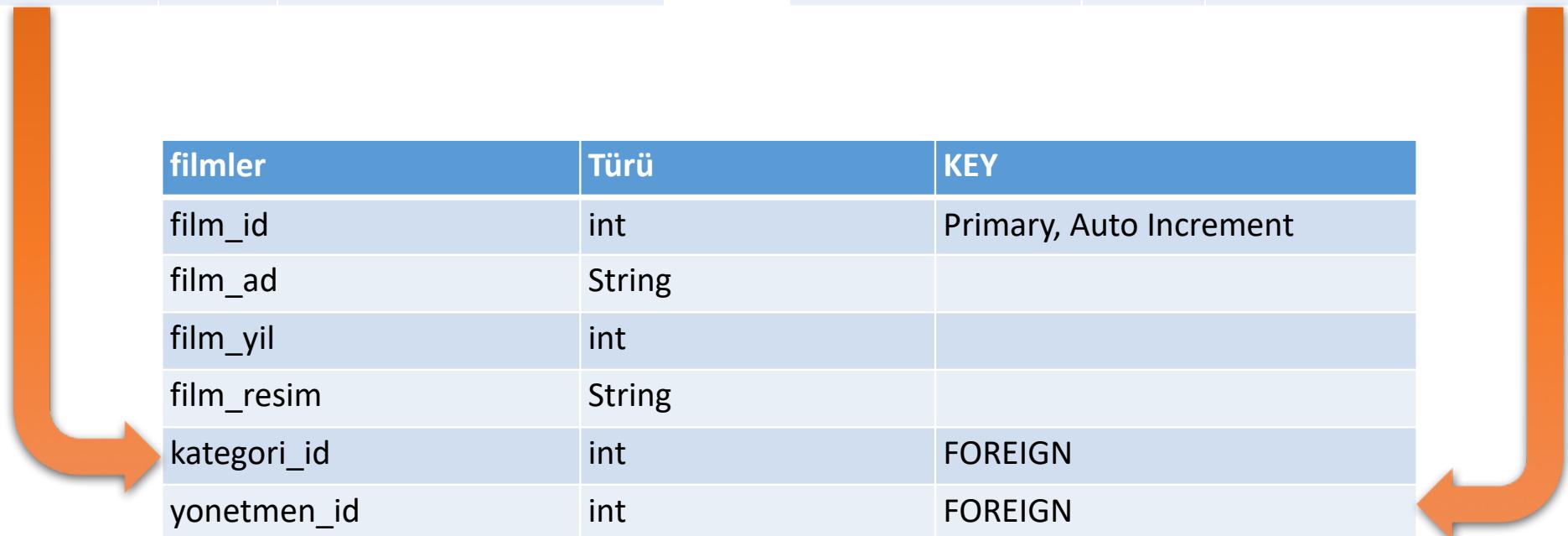
SQLite - Foreign Key

Örnek Veri Tabanı Modeli

| kategoriler | Türü | KEY |
|-------------|--------|-------------------------|
| kategori_id | int | Primary, Auto Increment |
| kategori_ad | String | |

| yonetmenler | Türü | KEY |
|-------------|--------|-------------------------|
| yonetmen_id | int | Primary, Auto Increment |
| yonetmen_ad | String | |

| filmler | Türü | KEY |
|-------------|--------|-------------------------|
| film_id | int | Primary, Auto Increment |
| film_ad | String | |
| film_yil | int | |
| film_resim | String | |
| kategori_id | int | FOREIGN |
| yonetmen_id | int | FOREIGN |



Sınıf Modelleri

```
class Yonetmenler {  
    int yonetmen_id;  
    String yonetmen_ad;  
  
    Yonetmenler(this.yonetmen_id, this.yonetmen_ad);  
}  
  
class Kategoriler {  
    int kategori_id;  
    String kategori_ad;  
  
    Kategoriler(this.kategori_id, this.kategori_ad);  
}  
  
class Filmler {  
    int film_id;  
    String film_ad;  
    int film_yil;  
    String film_resim;  
    Kategoriler kategori;  
    Yonetmenler yonetmen;  
  
    Filmler(this.film_id, this.film_ad, this.film_yil, this.film_resim, this.kategori, this.yonetmen);  
}
```

Tüm Verilerin Alınması (Foreign Key)

```
Future<List<Filmler>> tumFilmler() async {
    var db = await VeritabaniYardimcisi.veritabaniErisim();

    List<Map<String, dynamic>> maps =
        await db.rawQuery("SELECT * FROM filmler,kategoriler,yonetmenler "
            "WHERE filmler.kategori_id = kategoriler.kategori_id and filmler.yonetmen_id = yonetmenler.yonetmen_id");

    return List.generate(maps.length, (i) { //Gelen satırları listeye dönüştürme
        var satir = maps[i];
        //Her satırın içindeki verileri isimleriyle alarak nesneye dönüştürülür.

        var k = Kategoriler(satir["kategori_id"],satir["kategori_ad"]);
        var y = Yonetmenler(satir["yonetmen_id"],satir["yonetmen_ad"]);
        var f = Filmler(satir["film_id"],satir["film_ad"],satir["film_yil"],satir["film_resim"],k,y);

        return f;
    });
}
```

| filmler | Türü | KEY |
|-------------|--------|-------------------------|
| film_id | int | Primary, Auto Increment |
| film_ad | String | |
| film_yil | int | |
| film_resim | String | |
| kategori_id | int | FOREIGN |
| yonetmen_id | int | FOREIGN |

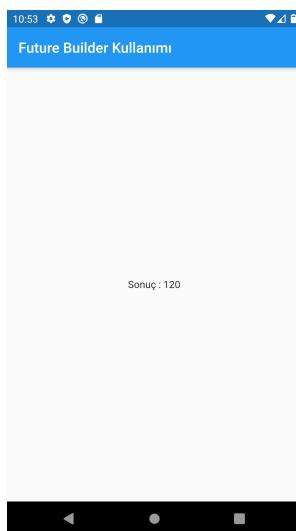
FutureBuilder ile SQLite Çalışması

FutureBuilder

- Asenkron işlemler için kullanılan bir yapıdır.
- async özelliği olan fonksiyonu kullanırken await özelliği ile sadece yapması gereken işlemi bitirene kadar çalışmasını sağlarız.
- Fakat await kullanmak için async özelliği olan fonksiyon içinde olmamız gereklidir.
- async özelliği olan fonksiyonu widget içinde kullanmak istediğimizde async özelliği olması gerekmektedir.Bu özellik widgetlarda yoktur.
- Widget içinde async özelliğini kullanmak için FutureBuilder yapısı gereklidir.

FutureBuilder Kullanımı

```
Future<int> faktoriyelHesapla(int sayi) async{  
  
    var sonuc = 1;  
  
    for(var i=1;i<=sayi;i++){  
        sonuc = sonuc * i ;  
    }  
  
    return sonuc;  
}
```



```
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: Text(widget.title),  
        ), // AppBar  
        body: Center(  
            child: Column(  
                mainAxisAlignment: MainAxisAlignment.center,  
                children: <Widget>[  
                    FutureBuilder<int>(  
                        future: faktoriyelHesapla(5),  
                        builder: (context,snapshot){  
  
                            if(snapshot.hasError){  
                                print("Hata sonunu : ${snapshot.error}");  
                            }  
  
                            if(snapshot.hasData){  
                                return Text("Sonuç : ${snapshot.data}");  
                            }else{  
                                return Center(child: Text("Gösterilecek Veri Yok"));  
                            }  
                        },  
                    ), // FutureBuilder  
                ], // <Widget>[]  
            ), // Column  
        ), // Center  
    ); // Scaffold  
}
```

Fonksiyonun çalışması sonucunda hata oluşuyor mu kontrolü

Fonksiyonun çalışması sonucunda veri var mı yok mu kontrolü

Çalıştıracağı fonksiyonun geri dönüş türü.

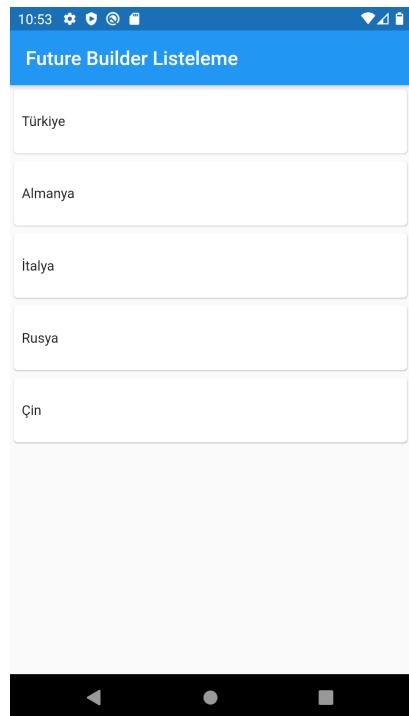
Çalıştırılacak fonksiyon

Fonksiyonun çalışma sonucunu temsil eden değişken.

Eğer gelen veri boş ise yani null ise burası çalışır ve tasarımında istediğimizi gösterebiliriz.

FutureBuilder ile Listeleme Çalışması

```
Future<List<String>> verileriGetir() async{  
  
    var ulkeListesi = ["Türkiye", "Almanya", "İtalya", "Rusya", "Çin"];  
  
    //Bu metodun lokal veritabanı veya internet üzerindeki veritabanından  
    //bu listeyi aldığımızı düşünelim.  
    return ulkeListesi;  
}
```



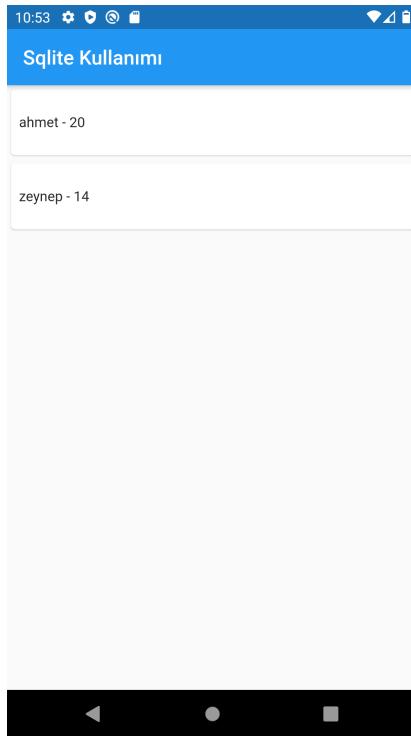
```
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: Text("")),  
        body: FutureBuilder<List<String>>(  
            future: verileriGetir(),  
            builder: (context,snapshot){  
                if(snapshot.hasData){  
                    var ulkeListesi = snapshot.data;  
  
                    return ListView.builder(  
                        itemCount: ulkeListesi!.length,  
                        itemBuilder: (context,indeks){  
                            var ülke = ulkeListesi[indeks];  
                            return Card(  
                                child: Padding(padding: const EdgeInsets.all(8.0),  
                                    child: SizedBox(height: 50,  
                                        child: Row( children: [Text(ulke),],),  
                                    ), // SizedBox  
                                ), // Padding  
                            ); // Card  
                        },  
                    ); // ListView.builder  
                }else{  
                    return Center();  
                }  
            },  
        ), // FutureBuilder  
    ); // Scaffold
```

Eğer gelen veri boş ise
yani null ise tasarım
boş görünşün

FutureBuilder ile SQLite Çalışması

```
Future<List<Kisiler>> kisileriGetir() async{
  var liste = await Kisilerdao().tumKisiler();

  return liste;
}
```



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ), // AppBar
    body: FutureBuilder<List<Kisiler>>(
      future: kisileriGetir(),
      builder: (context,snapshot){
        if(snapshot.hasData){
          var kisilerListesi = snapshot.data;

          return ListView.builder(
            itemCount: kisilerListesi!.length,
            itemBuilder: (context,indeks){
              var kisi = kisilerListesi[indeks];
              return Card(
                child: Padding(padding: const EdgeInsets.all(8.0),
                  child: SizedBox(height: 50,
                    child: Row(
                      children: <Widget>[Text("${kisi.kisi_ad} - ${kisi.kisi_yas}"),],
                    ), // Row
                  ), // SizedBox
                ), // Padding
              ); // Card
            }); // ListView.builder
        }else{
          return Center();
        }
      },
    ), // FutureBuilder
  ); // Scaffold
}
```

Eğer gelen veri boş ise yani null ise tasarım boş görünüşün

Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan