Flutter ile Uygulama Geliştirme Kursu | Android & IOS

İleri Flutter

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

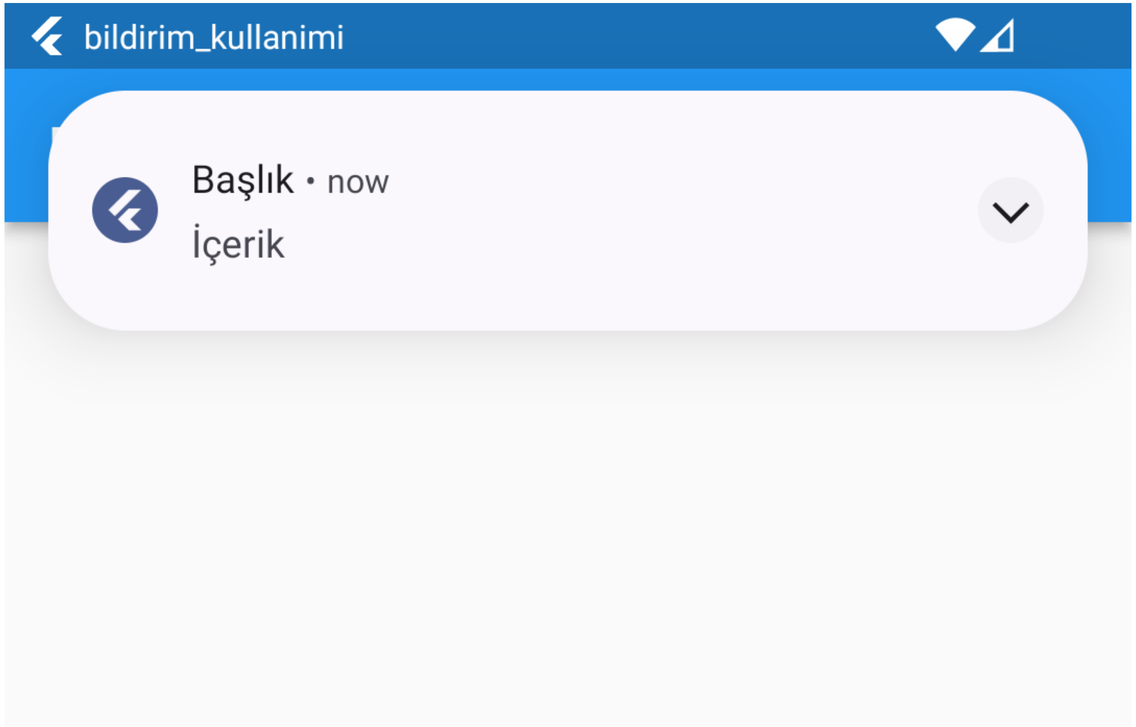
Android - IOS Developer and Trainer

Eğitim İçeriği

- Notification oluşturma
  - Duruma bağlı oluşan
  - Zamana duyarlı
- Timer

Notification - Bildirim

Notification - Bildirim

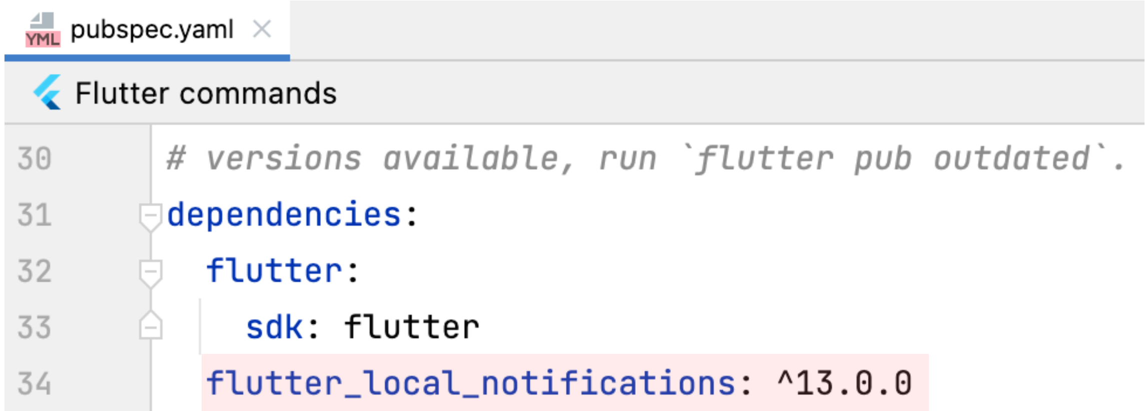


bildirim_kullanimi

Başlık • now

İçerik

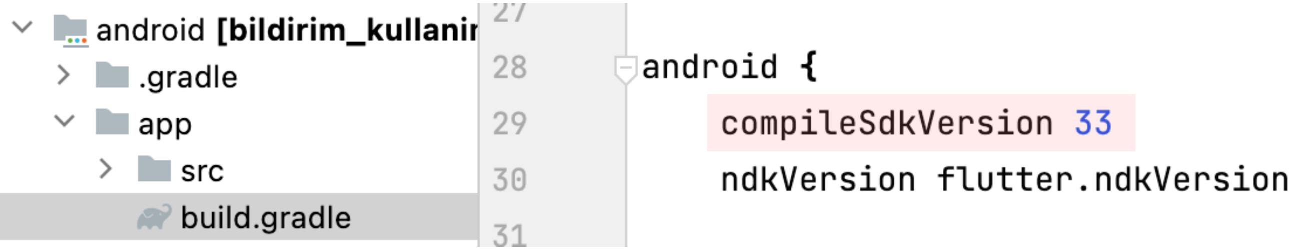
Kurulum



```yaml
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
  flutter_local_notifications: ^13.0.0
```

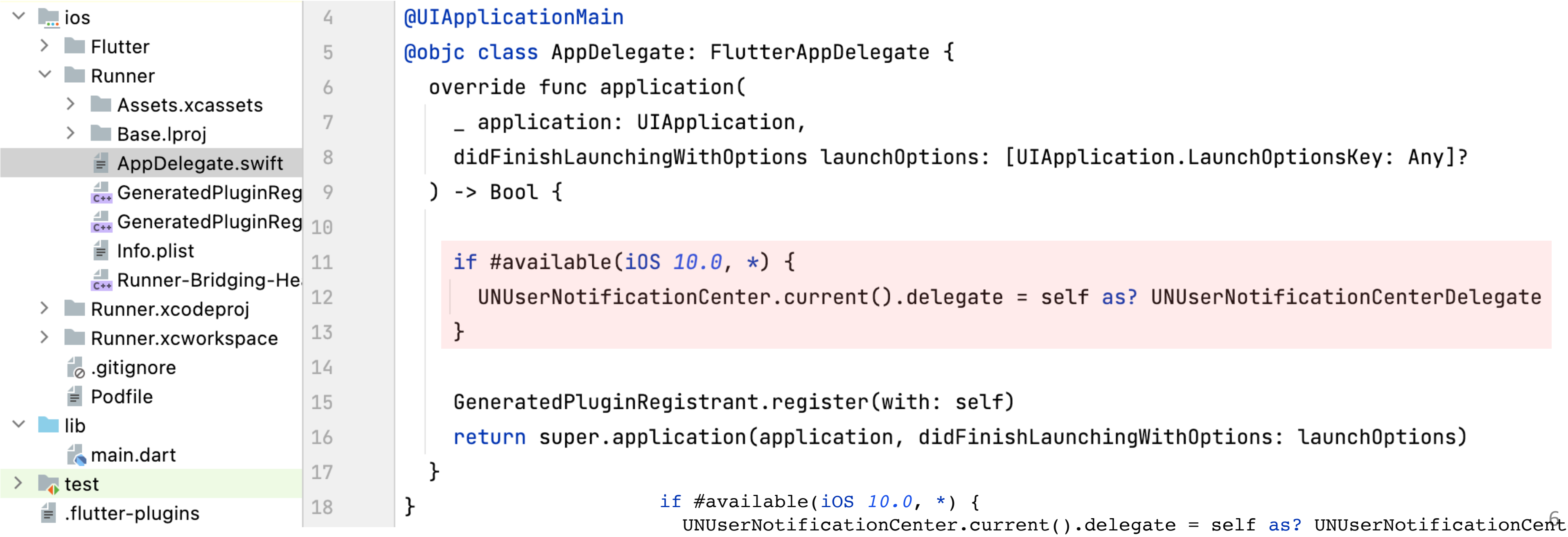
```yaml
flutter_local_notifications: ^13.0.0
```

Android Kurulum



```
android {
    compileSdkVersion 33
    ndkVersion flutter.ndkVersion
```

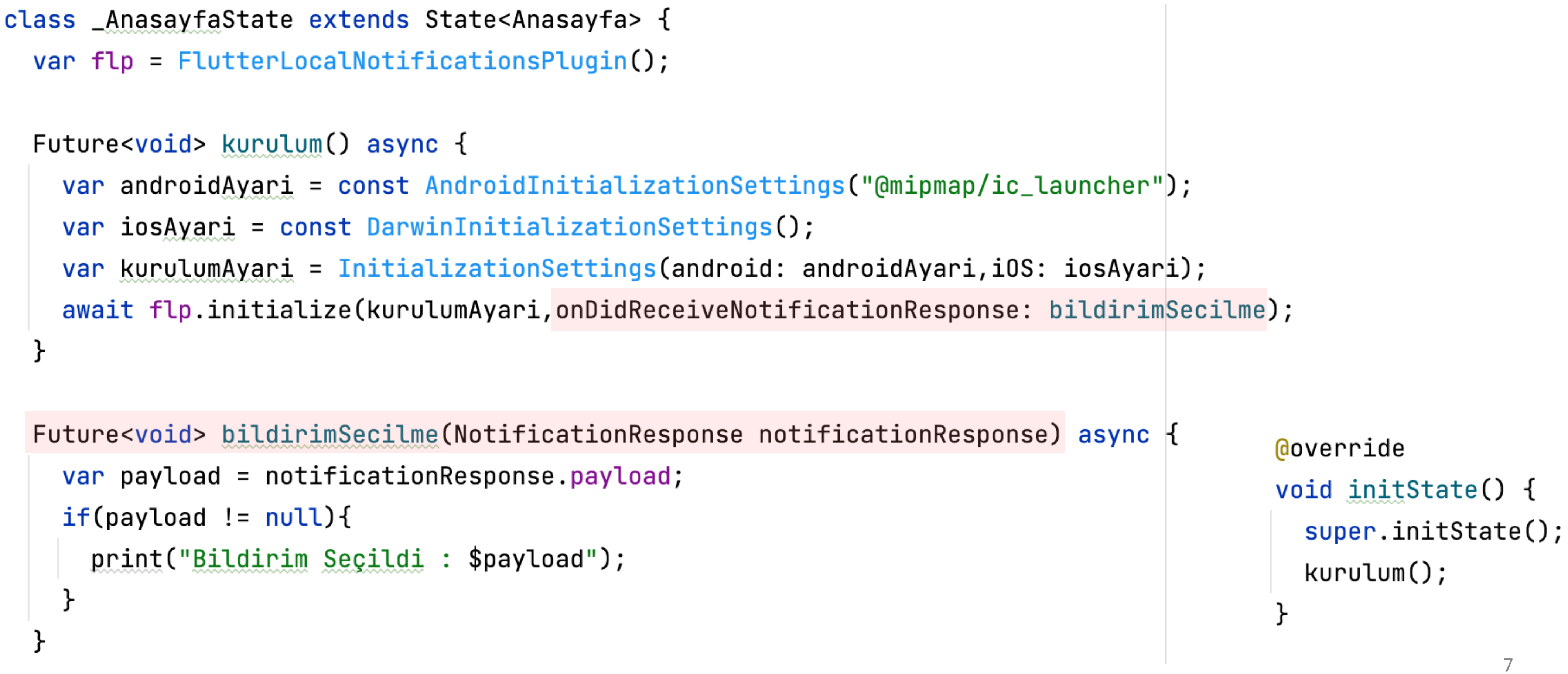
IOS Kurulum



```
@UIApplicationMain
@objc class AppDelegate: FlutterAppDelegate {
  override func application(
    _ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
  ) -> Bool {

    if #available(iOS 10.0, *) {
      UNUserNotificationCenter.current().delegate = self as? UNUserNotificationCenterDelegate
    }

    GeneratedPluginRegistrant.register(with: self)
    return super.application(application, didFinishLaunchingWithOptions: launchOptions)
  }
}
```

```
if #available(iOS 10.0, *) {
  UNUserNotificationCenter.current().delegate = self as? UNUserNotificationCenterDelegate
}
```

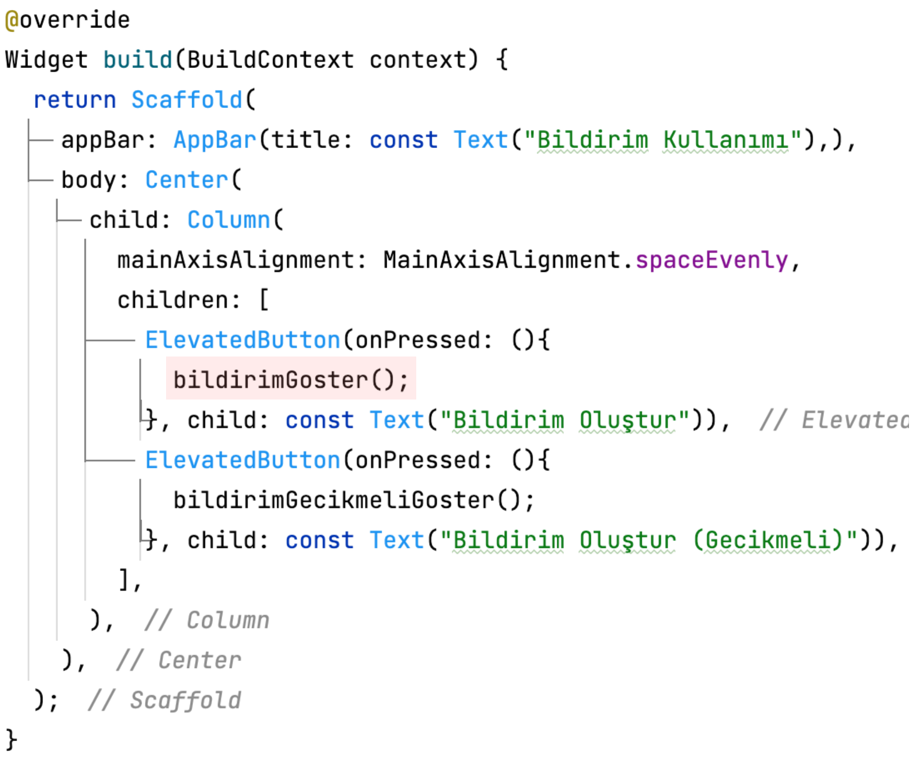
Yazılımsal Kurulum



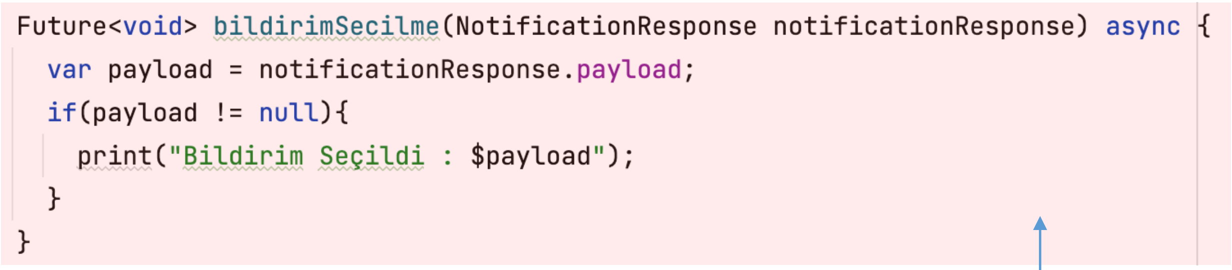
```
class _AnasayfaState extends State<Anasayfa> {
  var flp = FlutterLocalNotificationsPlugin();

  Future<void> kurulum() async {
    var androidAyari = const AndroidInitializationSettings("@mipmap/ic_launcher");
    var iosAyari = const DarwinInitializationSettings();
    var kurulumAyari = InitializationSettings(android: androidAyari,iOS: iosAyari);
    await flp.initialize(kurulumAyari,onDidReceiveNotificationResponse: bildirimSecilme);
  }

  Future<void> bildirimSecilme(NotificationResponse notificationResponse) async {
    var payload = notificationResponse.payload;
    if(payload != null){
      print("Bildirim Seçildi : $payload");
    }
  }
```

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

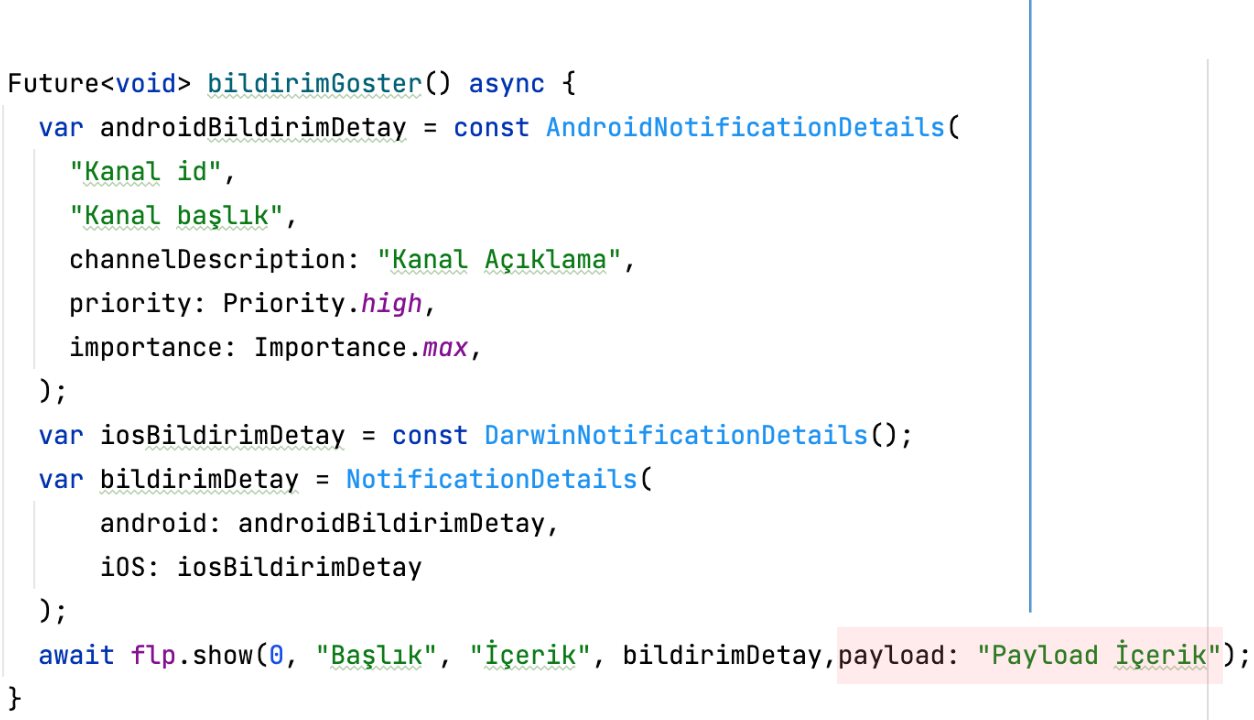
Bildirim Gösterme



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text("Bildirim Kullanımı"),),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          ElevatedButton(onPressed: (){
            bildirimGoster();
          }, child: const Text("Bildirim Oluştur")),  // Elevatec
          ElevatedButton(onPressed: (){
            bildirimGecikmeliGoster();
          }, child: const Text("Bildirim Oluştur (Gecikmeli)")),
        ],
      ),  // Column
    ),  // Center
  );  // Scaffold
}
```



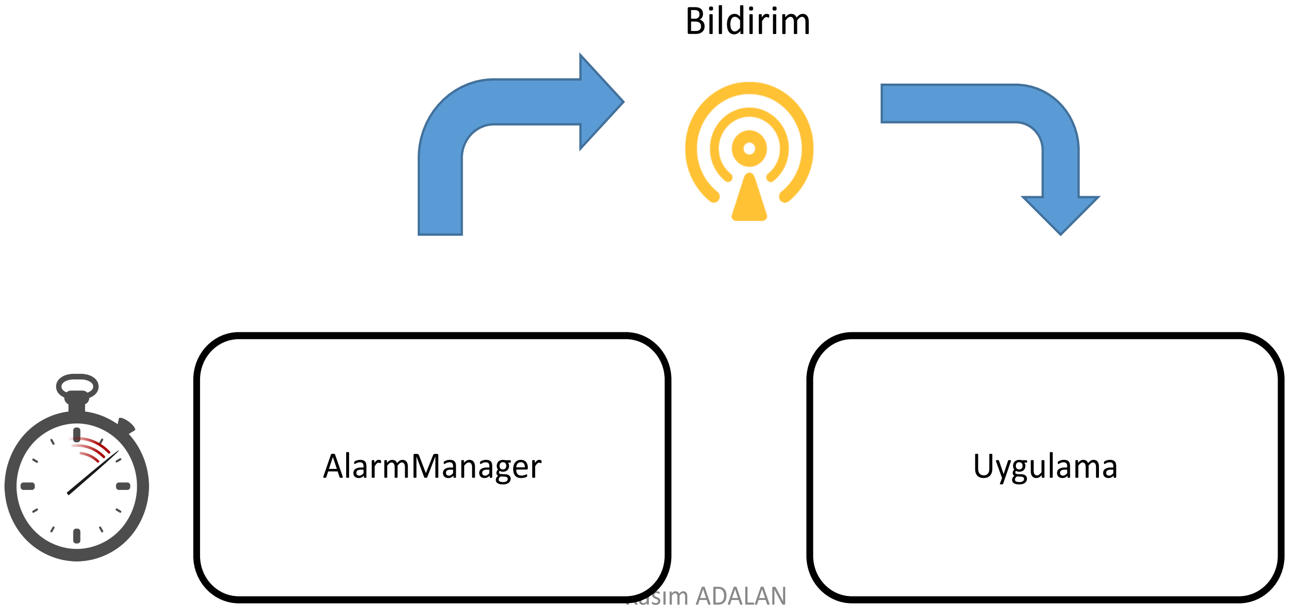
```
Future<void> bildirimSecilme(NotificationResponse notificationResponse) async {
  var payload = notificationResponse.payload;
  if(payload != null){
    print("Bildirim Seçildi : $payload");
  }
}
```



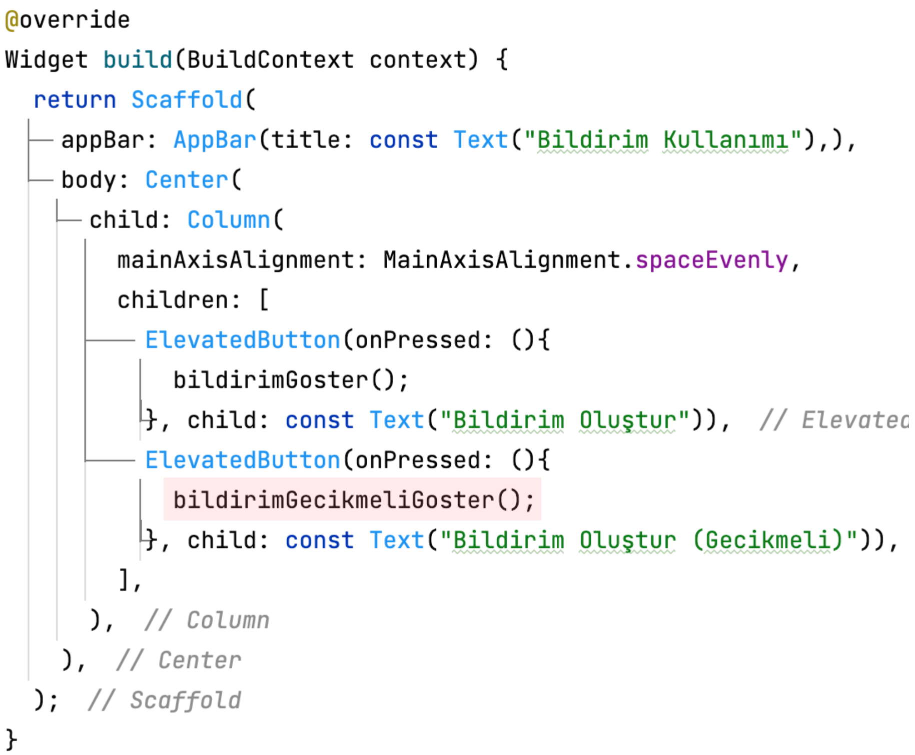
```
Future<void> bildirimGoster() async {
  var androidBildirimDetay = const AndroidNotificationDetails(
    "Kanal id",
    "Kanal başlık",
    channelDescription: "Kanal Açıklama",
    priority: Priority.high,
    importance: Importance.max,
  );
  var iosBildirimDetay = const DarwinNotificationDetails();
  var bildirimDetay = NotificationDetails(
      android: androidBildirimDetay,
      iOS: iosBildirimDetay
  );
  await flp.show(0, "Başlık", "İçerik", bildirimDetay,payload: "Payload İçerik");
}
```

Zamana Duyarlı Bildirim

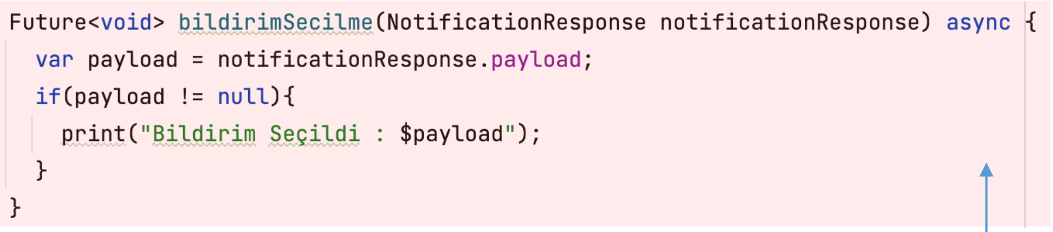
- Alarm manager alt yapısını kullanarak istenilen zamanda gecikmeli bildirim oluşturabiliriz.
- Android için ek kurulumlar gerekmektedir.



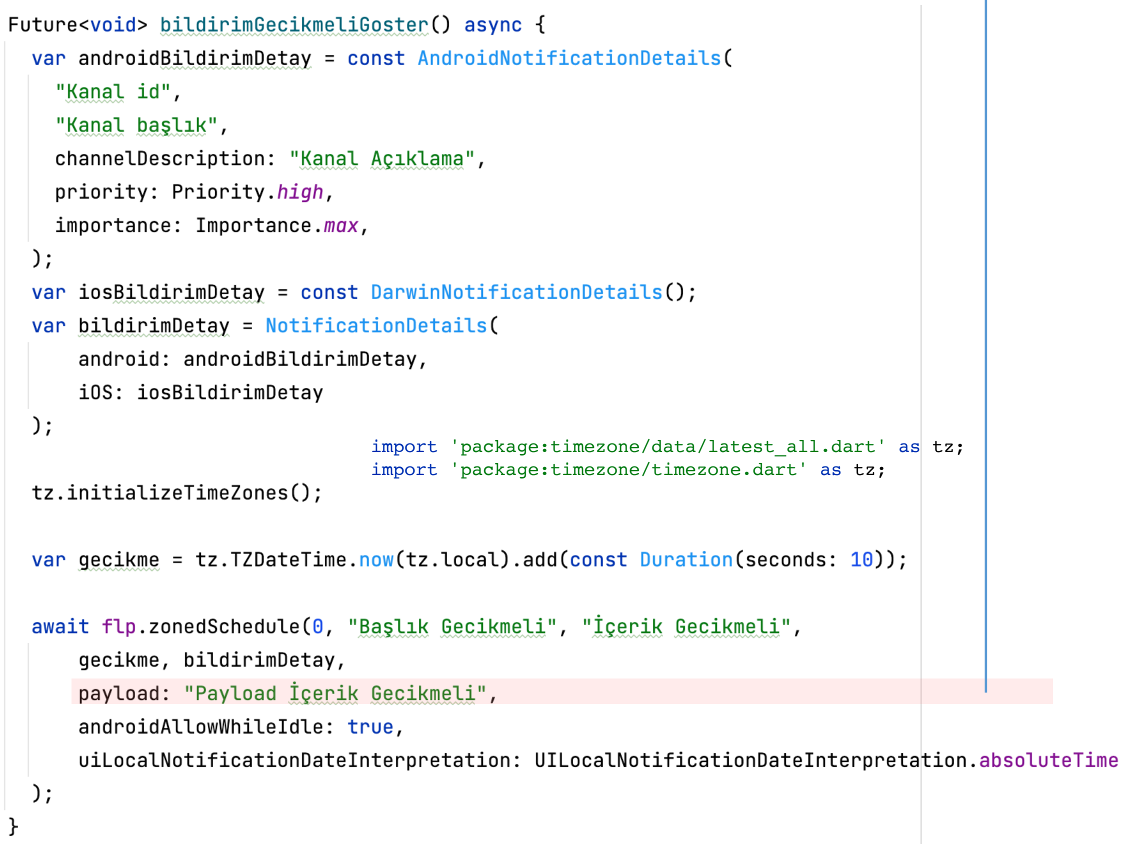
Gecikmeli Bildirim Gösterme



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text("Bildirim Kullanımı"),),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          ElevatedButton(onPressed: (){
            bildirimGoster();
          }, child: const Text("Bildirim Oluştur")),  // Elevatec
          ElevatedButton(onPressed: (){
            bildirimGecikmeliGoster();
          }, child: const Text("Bildirim Oluştur (Gecikmeli)")),
        ],
      ),  // Column
    ),  // Center
  );  // Scaffold
}
```



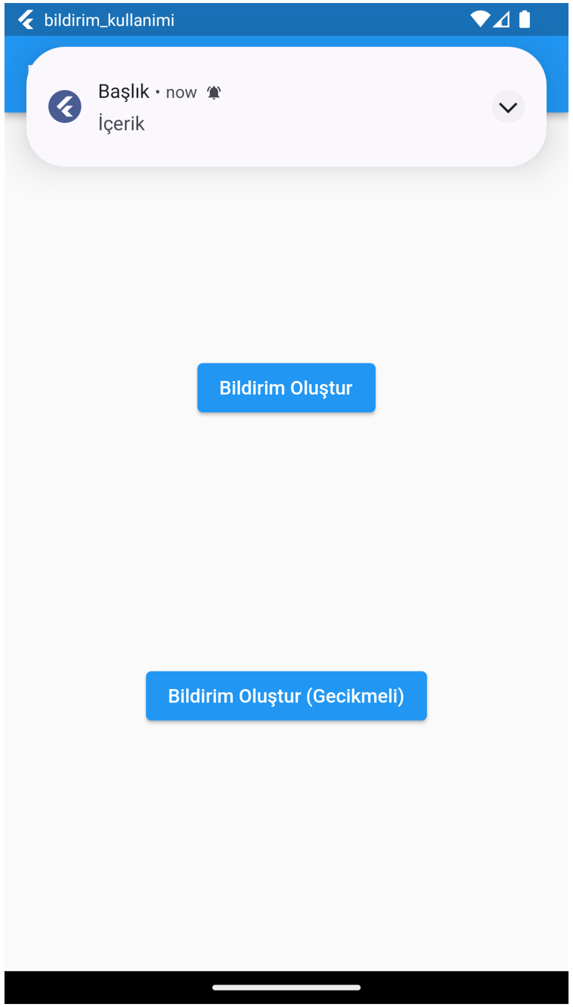
```
Future<void> bildirimSecilme(NotificationResponse notificationResponse) async {
  var payload = notificationResponse.payload;
  if(payload != null){
    print("Bildirim Seçildi : $payload");
  }
}
```



```
Future<void> bildirimGecikmeliGoster() async {
  var androidBildirimDetay = const AndroidNotificationDetails(
    "Kanal id",
    "Kanal başlık",
    channelDescription: "Kanal Açıklama",
    priority: Priority.high,
    importance: Importance.max,
  );
  var iosBildirimDetay = const DarwinNotificationDetails();
  var bildirimDetay = NotificationDetails(
      android: androidBildirimDetay,
      iOS: iosBildirimDetay
  );

  tz.initializeTimeZones();

  var gecikme = tz.TZDateTime.now(tz.local).add(const Duration(seconds: 10));

  await flp.zonedSchedule(0, "Başlık Gecikmeli", "İçerik Gecikmeli",
      gecikme, bildirimDetay,
      payload: "Payload İçerik Gecikmeli",
      androidAllowWhileIdle: true,
      uiLocalNotificationDateInterpretation: UILocalNotificationDateInterpretation.absoluteTime
  );
}
```

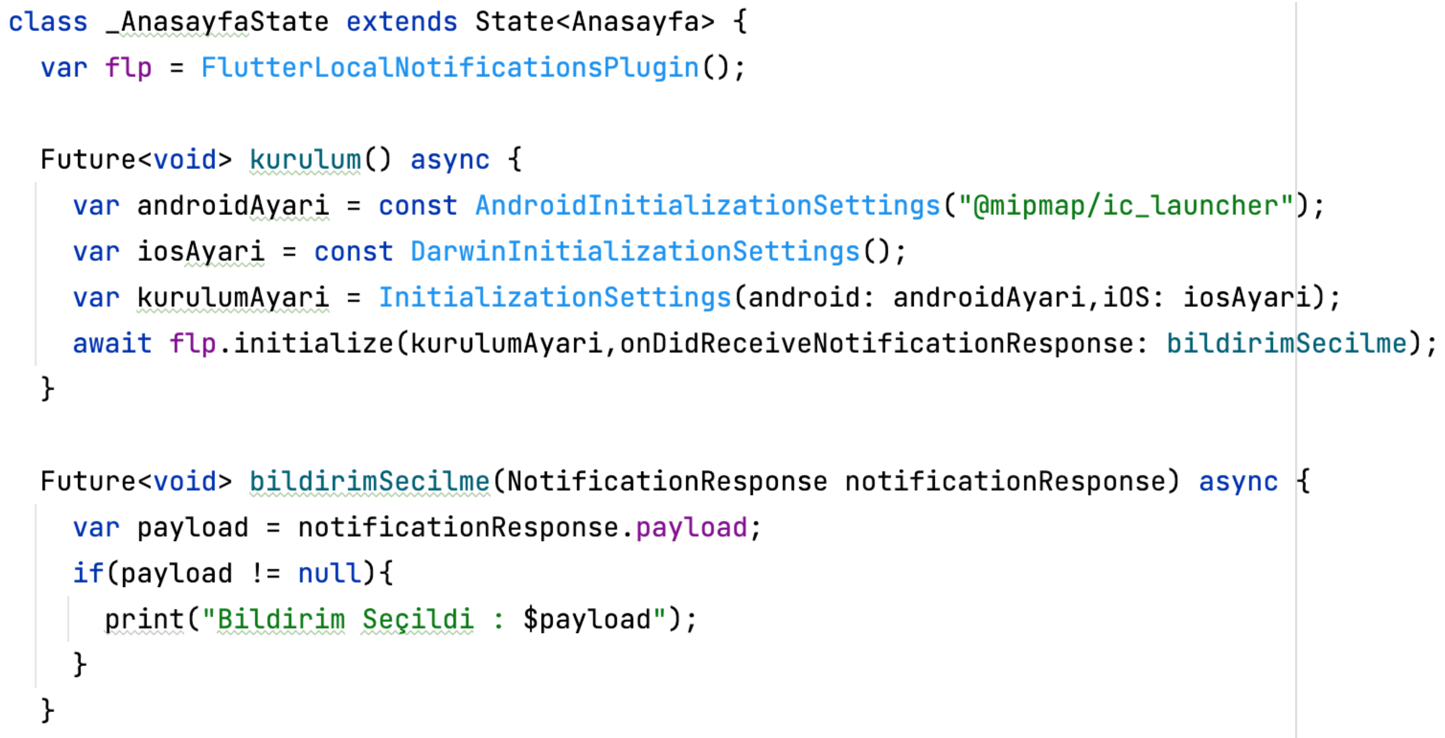
```
import 'package:timezone/data/latest_all.dart' as tz;
import 'package:timezone/timezone.dart' as tz;
```

Final Kod

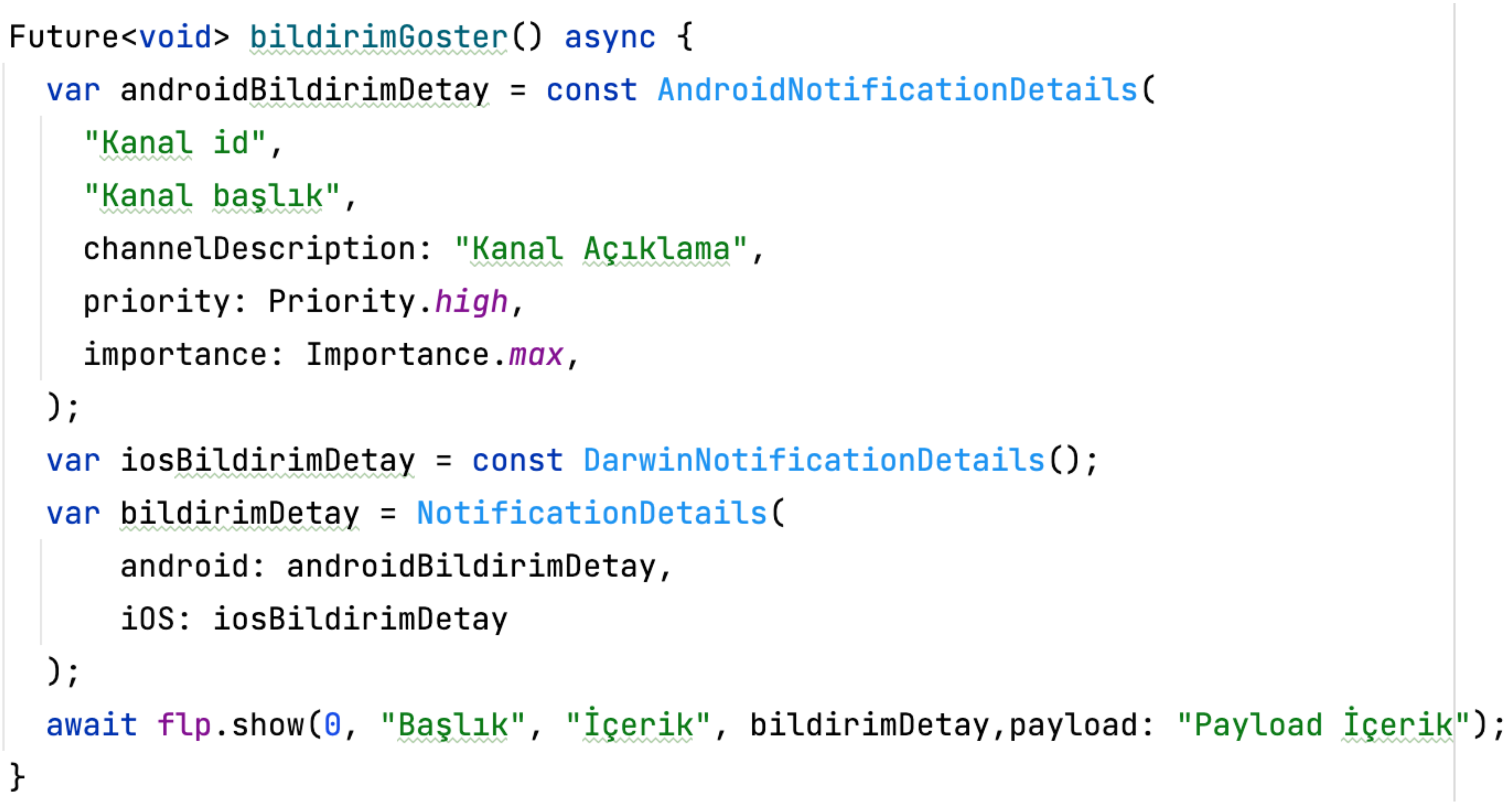


Başlık · now

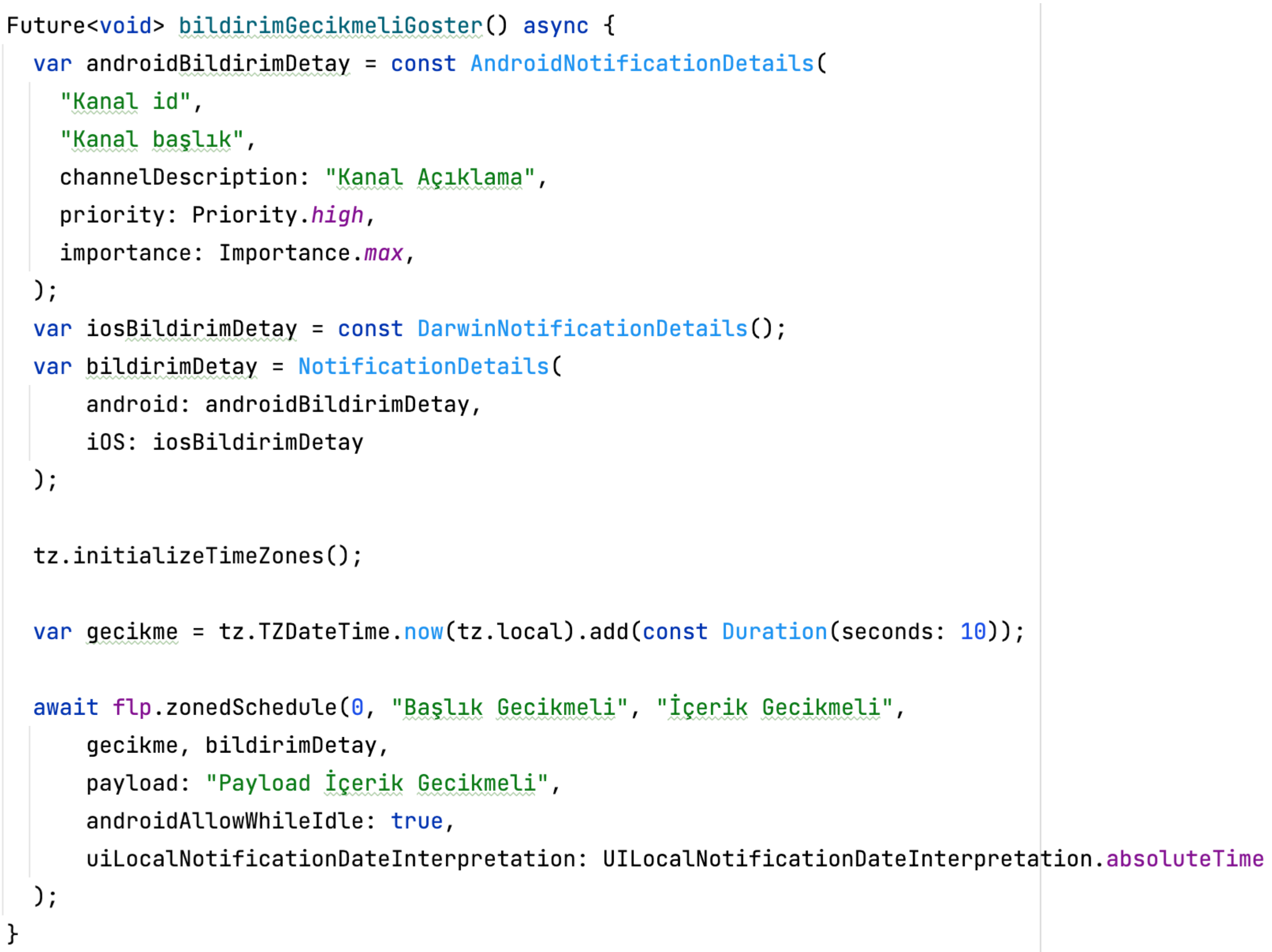
İçerik



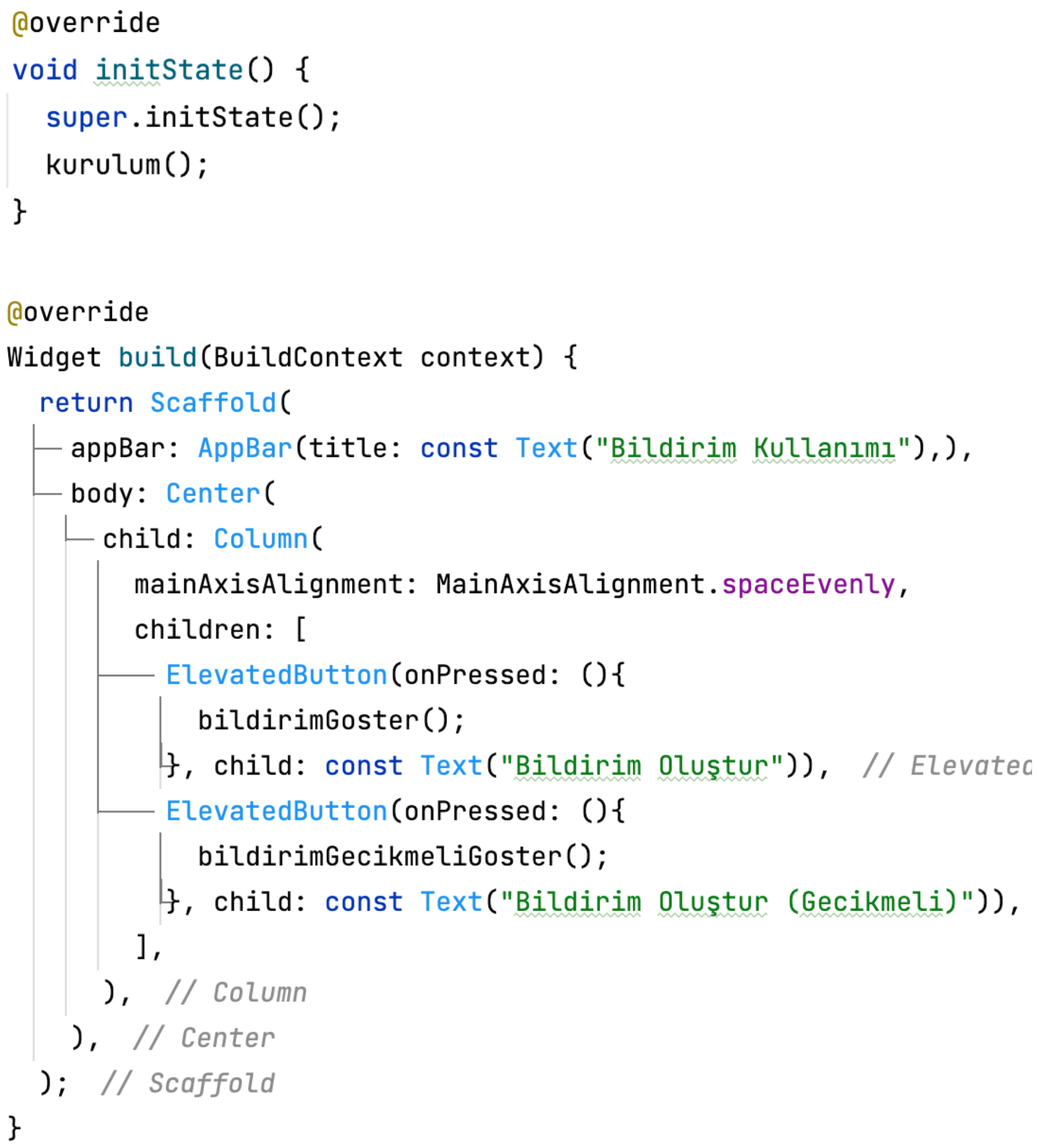
```
class _AnasayfaState extends State<Anasayfa> {
  var flp = FlutterLocalNotificationsPlugin();

  Future<void> kurulum() async {
    var androidAyari = const AndroidInitializationSettings("@mipmap/ic_launcher");
    var iosAyari = const DarwinInitializationSettings();
    var kurulumAyari = InitializationSettings(android: androidAyari,iOS: iosAyari);
    await flp.initialize(kurulumAyari,onDidReceiveNotificationResponse: bildirimSecilme);
  }

  Future<void> bildirimSecilme(NotificationResponse notificationResponse) async {
    var payload = notificationResponse.payload;
    if(payload != null){
      print("Bildirim Seçildi : $payload");
    }
  }
```



```dart
Future<void> bildirimGoster() async {
  var androidBildirimDetay = const AndroidNotificationDetails(
    "Kanal id",
    "Kanal başlık",
    channelDescription: "Kanal Açıklama",
    priority: Priority.high,
    importance: Importance.max,
  );
  var iosBildirimDetay = const DarwinNotificationDetails();
  var bildirimDetay = NotificationDetails(
      android: androidBildirimDetay,
      iOS: iosBildirimDetay
  );
  await flp.show(0, "Başlık", "İçerik", bildirimDetay,payload: "Payload İçerik");
}
```



```
Future<void> bildirimGecikmeliGoster() async {
  var androidBildirimDetay = const AndroidNotificationDetails(
    "Kanal id",
    "Kanal başlık",
    channelDescription: "Kanal Açıklama",
    priority: Priority.high,
    importance: Importance.max,
  );
  var iosBildirimDetay = const DarwinNotificationDetails();
  var bildirimDetay = NotificationDetails(
      android: androidBildirimDetay,
      iOS: iosBildirimDetay
  );

  tz.initializeTimeZones();

  var gecikme = tz.TZDateTime.now(tz.local).add(const Duration(seconds: 10));

  await flp.zonedSchedule(0, "Başlık Gecikmeli", "İçerik Gecikmeli",
      gecikme, bildirimDetay,
      payload: "Payload İçerik Gecikmeli",
      androidAllowWhileIdle: true,
      uiLocalNotificationDateInterpretation: UILocalNotificationDateInterpretation.absoluteTime
  );
}
```

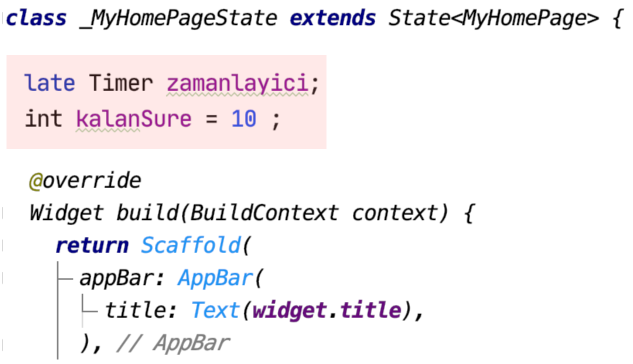


```
@override
void initState() {
  super.initState();
  kurulum();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text("Bildirim Kullanımı"),),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          ElevatedButton(onPressed: (){
            bildirimGoster();
          }, child: const Text("Bildirim Oluştur")),  // Elevatec
          ElevatedButton(onPressed: (){
            bildirimGecikmeliGoster();
          }, child: const Text("Bildirim Oluştur (Gecikmeli)")),
        ],
      ),  // Column
    ),  // Center
  );  // Scaffold
}
```

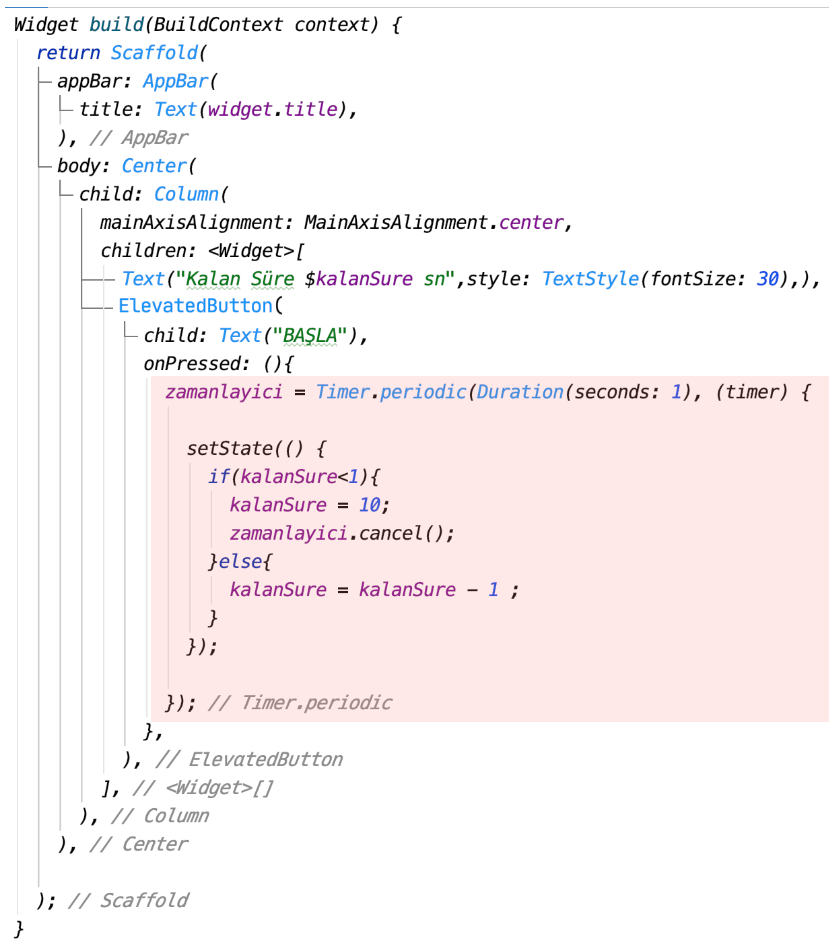
Timer

Timer

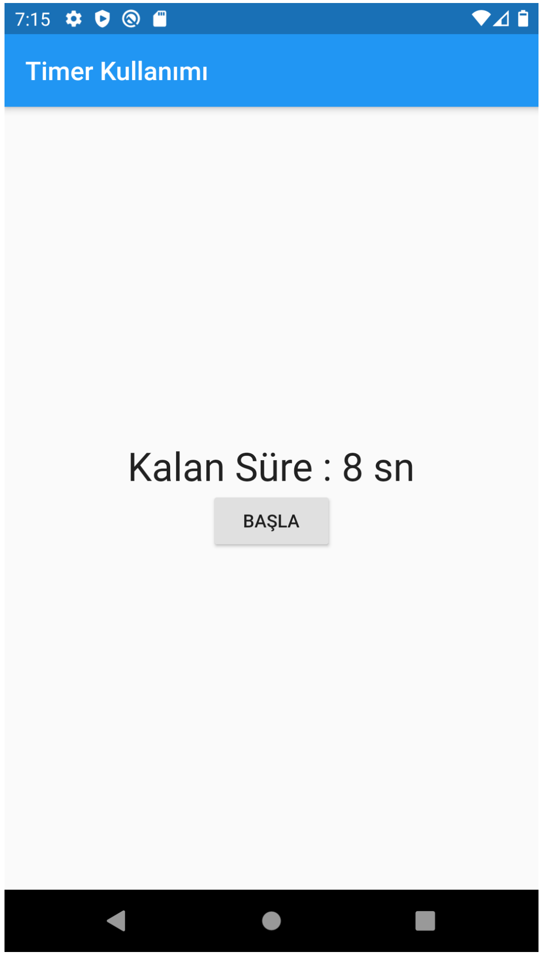
- Belirlenen başlangıç değerinden geri sayım yapabilen bir yapıdır.



```
class _MyHomePageState extends State<MyHomePage> {

  late Timer zamanlayici;
  int kalanSure = 10 ;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ), // AppBar
```



```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ), // AppBar
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Text("Kalan Süre $kalanSure sn",style: TextStyle(fontSize: 30),),
          ElevatedButton(
            child: Text("BAŞLA"),
            onPressed: (){
              zamanlayici = Timer.periodic(Duration(seconds: 1), (timer) {

                setState(() {
                  if(kalanSure<1){
                    kalanSure = 10;
                    zamanlayici.cancel();
                  }else{
                    kalanSure = kalanSure - 1 ;
                  }
                });

              }); // Timer.periodic
            },
          ), // ElevatedButton
        ], // <Widget>[]
      ), // Column
    ), // Center

  ); // Scaffold
}
```



Teşekkürler...

kasım-adalan

kasimadalan@gmail.com

kasimadalan