YouTube BM DERSLERİ

www.youtube.com/BMdersleri

MAKÜ
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Flutter ile Mobil Programlamaya Giriş

Flutter

14.HAFTA
İNTERNETTEN VERİ ÇEKME

Hazırlayan : **Zeynep İrem KESLER 1911404048**

Tarih : 03/06/2022

Sürüm : v1

Ders Yürütücüsü : Doç. Dr. İsmail KIRBAŞ

İÇİNDEKİLER

İnternetten Veri Çekme

İnternetten veri alışveriş çoğu uygulama için gereklidir. Dart ve Flutter bu tür bir iş için araçlar sağlar.

1-http Paketini Ekle: http paketi internetten veri almanın en basit yolunu sağlar.

Paketi kurmak için dosyanın dependencies bölümüne http paketini ekleyin.

http paketi

```
dependencies:
  http: <latest_version>
```

İnternetten Veri Çekme

- http paketini içe aktarın.

```
import 'package:http/http.dart' as http;
```

- Ayrıca, **AndroidManifest.xml** dosyanıza İnternet iznini ekleyin.

```
<!-- Required to fetch data from the internet. -->
<uses-permission android:name="android.permission.INTERNET" />
```

İnternetten Veri Çekme

2-http Paketini Kullanarak Bir Ağ İsteğinde Bulunun: ***http.get()*** yöntemi kullanarak JSONPlaceholder'dan bir örnek albümün nasıl alınacağını kapsar.

```
Future<http.Response> fetchAlbum() {
  return http.get(Uri.parse('https://jsonplaceholder.typicode.com/albums/1'));
}
```

- ***http.get()*** metodu, response içeren bir future döndürür.
- ***http.Response*** sınıfı, başarılı bir http çağrısından alınan verileri içerir.

İnternetten Veri Çekme

```dart
class Album {
  final int userId;
  final int id;
  final String title;

  const Album({
    required this.userId,
    required this.id,
    required this.title,
  });

  factory Album.fromJson(Map<String, dynamic> json) {
    return Album(
      userId: json['userId'],
      id: json['id'],
      title: json['title'],
    );
  }
}
```

3-Yanıtı Özel Bir Dart Nesnesine Dönüştürün: Bir ağ isteğinde bulunmak kolay olsa da, bir raw Future<http.Response> ile çalışmak pek uygun değildir. Daha kolay olması adına ***http.Response*** bir Dart nesnesine dönüştürün.

'Album' Sınıfı Oluşturun:

- İlk olarak ağ isteğinden gelen verileri içeren bir ***albüm*** sınıfı oluşturun.

İnternetten Veri Çekme

http.Response ' u bir Album ' a Dönüştürün:

fetchAlbum() ' u güncelleyip bir ***Future<Album>:*** fonksiyonuna döndürmek için adımları takip edin:

1. ***dart:convert*** paketi ile ***JSON Map'i*** response gövdesine dönüştürün.
2. Sunucu, durum kodu 200 olan bir OK yanıtı verirse, ***fromJson()*** fabrika metodunu kullanarak 'Album' içine JSON Map'i dönüştürün.
3. Sunucu, durum kodu 200 olan bir OK yanıtı vermezse, artık internetten bir albüm getiren bir işleve sahipsiniz.

```
Future<Album> fetchAlbum() async {
  final response = await http
      .get(Uri.parse('https://jsonplaceholder.typicode.com/albums/1'));

  if (response.statusCode == 200) {
    // If the server did return a 200 OK response,
    // then parse the JSON.
    return Album.fromJson(jsonDecode(response.body));
  } else {
    // If the server did not return a 200 OK response,
    // then throw an exception.
    throw Exception('Failed to load album');
  }
}
```

İnternetten Veri Çekme

4. Verileri Alın: ***initState()*** ya da ***didChangeDependencies()*** metodlarının ikisinin de içinde ***fetchAlbum()*** metodunu çağır. ***initState()*** metodu tam olarak bir kez çağrılır ve bir daha asla çağrılmaz.

```
class _MyAppState extends State<MyApp> {
  late Future<Album> futureAlbum;

  @override
  void initState() {
    super.initState();
    futureAlbum = fetchAlbum();
  }
  // ...
}
```

İnternetten Veri Çekme

5. Verileri Görüntüleyin: Verileri ekranda görüntülemek için ***FutureBuilderwidget***'ı kullanın. Widget , FutureBuilder Flutter ile birlikte gelir ve eşzamansız veri kaynaklarıyla çalışmayı kolaylaştırır.

İki parametre sağlamalısınız:

1. 'Future' birlikte çalışmak istediğiniz kişi. Bu durumda, ***fetchAlbum()*** fonksiyonu future'da döndürülür.
2. Bir ***builder*** fonksiyonu, yüklemenin başarılı veya hatalı durumuna bağlı olarak Flutter'a ne oluşturulacağını söyleyen bir fonksiyondur.

```
FutureBuilder<Album>(
  future: futureAlbum,
  builder: (context, snapshot) {
    if (snapshot.hasData) {
      return Text(snapshot.data!.title);
    } else if (snapshot.hasError) {
      return Text('${snapshot.error}');
    }

    // By default, show a loading spinner.
    return const CircularProgressIndicator();
  },
)
```

İnternetten Veri Çekme

***initState()* İçinde Neden *fetchAlbum()* Çağrılır?**

- Kullanışlı olmasına rağmen, bir build() metoda API çağrısı koymanız önerilmez.

- Flutter, görünümdeki herhangi bir şeyi değiştirmesi gerektiğinde build() metodu çağırır ve bu sıklıkla olur. fetchAlbum() metodu, build() içine yerleştirilirse uygulamanın yavaşlamasına neden olan her yeniden oluşturmada tekrar tekrar çağrılır.
- fetchAlbum()'un saklanması Future ile yalnızca bir kez yürütülür ve sonraki yeniden oluşturmalar için önbelleğe alınır.

Web Database

Yardımcı Kaynaklar

- Adım Adım Flutter İle Mobil Uygulamalar (Rakıcı Oğuz , 2021)
- https://flutter.dev/

İlginiz için teşekkürler…

Hazırlayan : **Zeynep İrem KESLER 1911404048**
E-posta : zeynepiremkesler@gmail.com
Tarih : 03/06/2022
Ders Yürütücüsü : Doç. Dr. İsmail KIRBAŞ
E-posta : ismkir@gmail.com