

29.10.2025 Ders Notları

Meta Android Developer Professional Certificate - Ders Notları

Tarih: 29.10.2025

Konular:

- Ktor ile HTTP/HTTPS çağrıları yapma
- JSON veri formatı ve Kotlin'de kullanımı
- Görsel yükleme (Glide)
- Senkron ve asenkron işlemler
- REST API iletişimi

Ktor ile Network İşlemleri

Uygulamalar verileri sunuculardan almak için **HTTP çağrıları** yapıyor. Kotlin'de bunun için **Ktor client** kullanılıyor.

HTTPS kullanmak önemli çünkü:

- HTTP'ye göre daha güvenli
- Veriler şifreleniyor
- Yeni Android versiyonları HTTP'yi tamamen blokluyor

Ktor client oluşturma:

```
val client = HttpClient(Android) {  
    install(ContentNegotiation) {  
        json()  
    }  
}
```

Engine kavramı: Ktor multi-platform bir library, bu yüzden her platform için farklı engine'ler var. Android için `Android` engine kullanılıyor.

HTTP Response İşleme

Request başarılı olursa `HttpResponse` instance'ı dönüyor. Burada:

- Status code (200 = başarılı)

- Response body
- Request/response zamanları var

Body'yi almak için:

```
val responseBody = response.body<MenuCategory>()
```

Not: JSON response'ları işlemek için client'ı JSON ile configure etmek gerekiyor - bunu daha detaylı araştırmalıyım.

Senkron vs Asenkron İşlemler

Senkron: İkinci call, birincinin bitmesini bekler

Asenkron: Paralel çalışır

Asenkron için:

```
val deferred1 = async { client.get("url1") }  
val deferred2 = async { client.get("url2") }  
  
val result1 = deferred1.await()  
val result2 = deferred2.await()
```

`async` block arka planda çalışıyor ve `Deferred` instance dönüyor. Sonucu almak için `await()` kullanılıyor.

JSON - JavaScript Object Notation

JSON veri transferinde en çok kullanılan format. Sebepleri:

- Hafif (lightweight)
- JavaScript object syntax'ına benziyor
- JavaScript'te işlemesi kolay
- Language independent

JSON Syntax temelleri:

- İsimler çift tırnak içinde
- `:` ile key-value ayrılıyor
- `{}` object'leri sarmalıyor
- `[]` array'leri sarmalıyor

Little Lemon menu örneği:

```
{
  "appetizers": ["Bruschetta", "Fried Calamari"],
  "salads": ["Greek", "Caesar", "Grilled Chicken"],
  "drinks": ["Lemonade", "Iced Tea", "Soda"]
}
```

Kotlin'de JSON İşleme

GSON library kullanarak JSON ↔ Kotlin object dönüşümü:

1. Önce Gradle'a dependency ekle:

```
implementation("com.google.code.gson:gson:2.8.9")
```

2. Data class'ları oluştur:

```
@Serializable
data class Reservation(val table: Int, val time: String)

data class AllReservations(val date: String, val reservations: List<Reservation>)
```

3. JSON string → Kotlin object:

```
val gson = Gson()
val allReservations = gson.fromJson(jsonString, AllReservations::class.java)
```

4. Kotlin object → JSON string:

```
val jsonString = gson.toJson(allReservations)
```

Kişisel not: `@Serializable` annotation'ını doğru kullanmayı tam anlamadım, buna tekrar bakmam lazım.

Görsel Yükleme (Image Fetching)

Görsel yüklemek için **Glide** kullanıyoruz. Alternatifler: Picasso, Coil.

Glide özellikleri:

- JPEG, PNG, GIF, WEBP destekliyor
- Otomatik caching yapıyor
- Memory management otomatik
- Compose için `GlideImage` composable'ı var

Önemli: Büyük görseller veya çok fazla görsel aynı anda memory problemleri yaratabilir. Glide otomatik olarak:

- Cache'ler
- Resize eder
- Memory'den temizler

Transformations: `CenterCrop`, `FitCenter`, `CircleCrop` gibi built-in seçenekler var.

Pratik Örnek - Little Lemon Menu Fetch

Little Lemon app'inde menu fetch etmek için:

1. Ktor client oluştur
2. `getMenu` fonksiyonu yaz (suspend function olarak)
3. Response'u parse et
4. UI'da göster

```
private suspend fun getMenu(category: String): List<String> {  
    val response: Map<String, MenuCategory> =  
client.get("https://littlelemon.com/menu").body()  
    return response[category]?.menu ?: emptyList()  
}
```

Not: Suspend function'lar sadece coroutine içinden çağrılabilir - bunun lifecycle'ını daha iyi anlamam gerekiyor.

Button onClick'te:

```
viewModelScope.launch {  
    val menuItems = getMenu("Salads")  
    menuItemsLiveData.value = menuItems  
}
```

Genel Tekrar Notları

Bu hafta öğrendiklerim:

- ☒ Ktor ile network request
- ☒ JSON formatı ve Kotlin'de kullanımı
- ☒ Görsel yükleme (Glide)
- ☒ Senkron/asenkron işlemler
- ☒ REST API communication

Eksik hissettiğim noktalar:

- Coroutine lifecycle management
 - Error handling network call'larda
 - JSON parsing'te daha complex senaryolar
 - Glide customization options
-