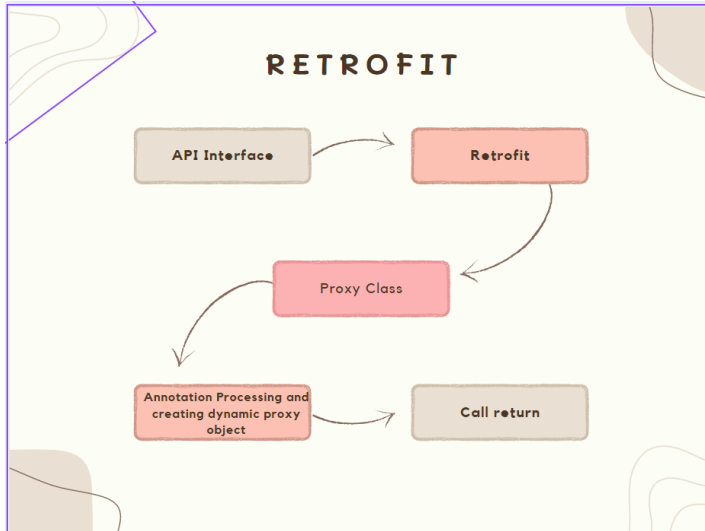


UMUT YUSUF ÇINAR
GELECEĞİ YAZANLAR ÖDEV – 6
RETROFIT İÇERİSİNDE KULLANILAN
HTTPS METOD ÇEŞİTLERİ,
EN ÇOK KULLANILAN ANNOTATION YAPILARI VE
DURUM KODLARI

RETROFIT NEDİR?

Retrofit, Android'de REST API'lerle iletişim kurmayı önemli ölçüde kolaylaştıran yaygın olarak kullanılan **bir HTTP istemci kütüphanesidir**. Bu kütüphane, sunucuya istek göndermek ve verileri almak için çeşitli HTTP metotları kullanır.



GET

Tanımı: Belirtilen URI'deki kaynağı almak için kullanılır. Sunucudan veri istemek için kullanılır. Genellikle veri almak için kullanılır ve **sunucu üzerinde herhangi bir değişiklik yapmaz**.

Kullanım Alanları: Bir kullanıcının profilini veya bir makalenin içeriğini gibi statik verileri almak. Bir listeleme işlemi gerçekleştirmek (örneğin, tüm kullanıcıları veya tüm makaleleri listelemek).

```

import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.GET

// Örnek API servisi için veri sınıfı
data class ExampleData(val id: Int, val name: String)

// Retrofit tarafından kullanılacak API interface'i
interface ApiService {
    @GET("exampleEndpoint")
    suspend fun getExampleData(): ExampleData
}

```

***Kod örneklerini, dokümanın okunabilirliğini ve anlaşılabilirliğini artırmak amacıyla yapay zekaya hazırlattım.**

POST:

Tanımı: Bu metod, belirtilen URI'ye yeni bir kaynak eklemek için kullanılır. Genellikle form verileri veya JSON verileri gibi gönderilen verileri sunucuya iletir.

Kullanım Alanları: Yeni bir kullanıcı kaydetmek veya bir makale oluşturmak gibi yeni bir veri oluşturmak. Bir formdan veya dosyadan veri göndermek.

```

import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.Body
import retrofit2.http.POST

// Örnek veri sınıfı
data class ExampleData(val id: Int, val name: String)

// Retrofit tarafından kullanılacak API interface'i
interface ApiService {
    @POST("exampleEndpoint")
    suspend fun postExampleData(@Body data: ExampleData): ExampleData
}

```

PUT:

Tanımı: Belirtilen URI'deki kaynağı güncellemek için kullanılır. Genellikle bir kaynağı tamamen değiştirmek için kullanılır.

Kullanım Alanları: Bir kullanıcının profilini veya bir makalenin içeriğini güncellemek gibi mevcut bir veriyi değiştirmek.

```
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.Body
import retrofit2.http.PUT

// Örnek veri sınıfı
data class ExampleData(val id: Int, var name: String)

// Retrofit tarafından kullanılacak API interface'i
interface ApiService {
    @PUT("exampleEndpoint/{id}")
    suspend fun putExampleData(@Body data: ExampleData): ExampleData
}
```

DELETE:

Tanımı: Bu metod, belirtilen URI'deki kaynağı silmek için kullanılır. Genellikle bir kaynağı kaldırmak için kullanılır. **Kullanılırken dikkatli olunması gerekir.**

Kullanım Alanları: Bir kullanıcıyı silmek veya bir makaleyi silmek gibi bir veriyi silmek.

```
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.DELETE
import retrofit2.http.Path

// Retrofit tarafından kullanılacak API interface'i
interface ApiService {
    @DELETE("exampleEndpoint/{id}")
    suspend fun deleteExampleData(@Path("id") id: Int)
}
```

RETROFIT'TE EN ÇOK KULLANILAN ANNOTATION'LAR:

@Headers: HTTP isteğine özel başlıklar eklemek için kullanılır. Örneğin, belirli bir kimlik doğrulama bilgilerini geçmek için kullanılabilir.

@Query: HTTP GET isteği sırasında query parametrelerini belirtmek için kullanılır. Query (sorgu) parametreleri, URL içinde **"?key=value"** şeklinde belirtilir.

@Path: URL içindeki değişken parçalarını belirtmek için kullanılır. Dinamik URL oluşturmak için kullanılır ve genellikle **RESTful servislerle çalışırken path parametrelerini temsil etmek için kullanılır.**

@Body: HTTP isteği gövdesini belirtmek için kullanılır. Genellikle POST, PUT gibi metodlarla kullanılır ve gönderilen verinin biçimini belirler.

YAYGIN OLARAK KULLANILAN HTTP DURUM KODLARI NELERDİR?

HTTP Durum Kodu Nedir?

HTTP durum kodları, bir istemcinin bir HTTP isteği gönderdiği sunucudan aldığı yanıtları belirtmek için kullanılır.

200 OK: İstek başarılı bir şekilde gerçekleştirildi ve istenen bilgiler sunucu tarafından başarıyla döndürüldü.

302 Found (Redirect): İstek yapılan kaynak başka bir konuma taşındı. Tarayıcı genellikle bu konuma otomatik olarak yönlendirilir.

304 Not Modified: Tarayıcı, sunucudan kaynağın son güncellenme tarihinden sonra herhangi bir değişiklik olmadığını belirten bu kodu alır. Bu durumda, tarayıcı önbellekten veriyi alabilir.

401 Unauthorized: Kullanıcı kimlik doğrulaması gerektiren bir kaynağa erişmeye çalıştığında sunucu tarafından gönderilir. Kullanıcı kimlik doğrulaması başarısız olduğunda bu durum kodu alınır.

403 Forbidden: Sunucu, istemcinin istenen kaynağa erişimini reddettiğini belirtir. Genellikle yetki eksikliği durumunda alınır.

404 Not Found: Son kullanıcıların en çok rastladığı durum kodudur. İstek yapılan kaynak sunucuda bulunamadı. Genellikle URL hataları veya geçersiz bağlantılar için kullanılır.

500 Internal Server Error: Sunucu, isteği işlerken bir hata oluştuğunu belirtir. Bu genellikle sunucu tarafındaki bir uygulama hatasını gösterir.

KAYNAKÇA

<https://square.github.io/retrofit/>

Google Çeviri

Yapay Zekâ Araçları