

RETROFIT

Retrofit, Android için bir HTTP kütüphanesidir. RESTful API'lerle etkileşim kurmak için kullanılır. HTTP isteklerini yapılandırmak ve sunucudan gelen yanıtları işlemek için kullanılır.

Retrofit Anotasyonları

1. **@GET:** Belirli bir URL'ye GET isteği göndermek için kullanılır. Method'un dönüş tipi geri dönüş değeri olarak belirtilir.

```
@GET("users")
fun getUsers(): Call<List<User>>
```

2. **@POST:** Belirli bir URL'ye POST isteği göndermek için kullanılır. Method parametreleri, POST isteği gövdesine dahil edilir.

```
@POST("user/create")
fun createUser(@Body user: User): Call<User>
```

3. **@PUT:** Belirli bir URL'ye PUT isteği göndermek için kullanılır. Benzer şekilde, method parametreleri, PUT isteği gövdesine dahil edilir.

```
@PUT("user/{id}")
fun updateUser(@Path("id") userId: Int, @Body user: User): Call<User>
```

4. **@DELETE:** Belirli bir URL'ye DELETE isteği göndermek için kullanılır. Silinecek kaynağın kimliği genellikle URL'de belirtilir.

```
@DELETE("user/{id}")
fun deleteUser(@Path("id") userId: Int)
```

5. **@Path:** URL'deki değişken bölümlerini belirtmek için kullanılır. Bu annotation, dinamik olarak değişen URL parçalarını belirtmek için kullanılır. Örneğin, bir kullanıcının kimliği gibi değişken URL parçalarını belirtmek için kullanılabilir.

```
@GET("users/{id}")
fun getUserById(@Path("id") userId: Int): Call<User>
```

6. **@Query:** URL sorgu parametrelerini belirtmek için kullanılır. Bu annotation, GET isteğine sorgu parametreleri eklemek için kullanılır. Örneğin, bir arama sorgusu gibi parametreleri belirtmek için kullanılabilir.

```
@GET("users")
fun searchUsers(@Query("q") query: String): Call<List<User>>
```

7. **@Body:** POST veya PUT isteklerinde gövde verilerini belirtmek için kullanılır. Bu annotation, isteğin gövde kısmına gönderilecek nesneyi belirtmek için kullanılır.

```
@POST("user/create")
fun createUser(@Body user: User): Call<User>
```

8. **@Header:** HTTP başlık parametrelerini belirtmek için kullanılır. Bu annotation, belirli bir başlık adıyla bir başlık değeri eklemek için kullanılır.

```
@GET("user")
fun getUser(@Header("Authorization") token: String): Call<User>
```

HTTP Status Kodları

1. **200 OK:** Her şey yolunda, sunucu talebi başarıyla yerine getirdi.
2. **201 Created:** Sunucu, yeni bir kaynak oluşturdu.
3. **204 No Content:** Sunucu, talebi başarıyla işledi ancak yanıt olarak içerik göndermedi.
4. **400 Bad Request:** Sunucu, talebi anlayamadı veya geçersiz veri aldı.
5. **401 Unauthorized:** Kullanıcı, kimlik doğrulaması gerektiren bir kaynağa erişmeye çalışırken yetkilendirilmedi.
6. **403 Forbidden:** İstek, sunucu tarafından reddedildi. Kullanıcının kaynağa erişim izni yok.
7. **404 Not Found:** İstenen kaynak bulunamadı.
8. **500 Internal Server Error:** Sunucuda bir hata oluştu ve istek tamamlanamadı.
9. **502 Bad Gateway:** Sunucu, geçersiz bir yanıt aldı.
10. **503 Service Unavailable:** Sunucu geçici olarak hizmet dışı veya aşırı yüklü.