

HTTP Method Çeşitleri

GET

- Sunucudan veri alma işlemi için kullanılır.
- Veriler URL içerisinde veya istek gövdesinde gönderilebilir.
- Örnek: Bir web sitesinin ana sayfasını veya bir kullanıcının profilini getirmek.

POST

- Sunucuya veri gönderme işlemi için kullanılır.
- Yeni bir kaynak oluşturmak veya var olan bir kaynağı güncellemek için kullanılabilir.
- Veriler genellikle istek gövdesinde gönderilir.
- Örnek: Yeni bir kullanıcı kaydetmek veya bir gönderi oluşturmak.

PUT

- Var olan bir kaynağı güncellemek için kullanılır.
- Kaynağın tam bir temsilini istek gövdesinde göndermeniz gerekir.
- POST'tan farkı, idempotent olmasıdır, yani aynı isteği birden fazla kez göndermek sunucu üzerinde yan etkiye neden olmaz.
- Örnek: Bir kullanıcının profilini güncellemek veya bir gönderiyi değiştirmek.

DELETE

- Sunucudan bir kaynağı silmek için kullanılır.
- Silinecek kaynağın kimliğini URL içerisinde veya istek gövdesinde göndermeniz gerekir.
- Örnek: Bir kullanıcının hesabını silmek veya bir gönderiyi silmek.

NOT: POST ve PUT birbirine çok benzer fakat aralarında ince bir fark vardır. Hiç olmayan bir şeyi göndermek POST, var olan bir şeyi tekrar göndermek PUT ile ifade edilebilir. Örneğin bir kullanıcının profil fotoğrafı hiç yoksa bunu POST ile ekleyebiliriz. Fakat var olan bir fotoğrafı güncellemek için PUT kullanırız.

Retrofit İçerisinde En Çok Kullanılan Annotations'lar

@GET

- **@GET** annotation'u, bir URL'ye bir GET isteği gönderecek bir yöntem tanımlamak için kullanılır. Annotation'un içindeki URL, sunucuya gönderilecek isteğin yolunu belirtir.

```
@GET("/users/{id}")
suspend fun getUser(@Path("id") userId: Int): User
```

@POST

- **@POST** annotation'u, bir URL'ye bir POST isteği gönderecek bir yöntem tanımlamak için kullanılır. Annotation'un içindeki URL, sunucuya gönderilecek isteğin yolunu belirtir. İstek gövdesi, **@Body** annotation'u ile belirtilir.

```
@POST("/users")
suspend fun createUser(@Body user: User): User
```

@PUT

- **@PUT** annotation'u, bir URL'ye bir PUT isteği gönderecek bir yöntem tanımlamak için kullanılır. Annotation'un içindeki URL, güncellenecek kaynağın yolunu belirtir. İstek gövdesi, **@Body** annotation'u ile belirtilir.

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

@DELETE

- **@DELETE** annotation'u, bir URL'ye bir DELETE isteği gönderecek bir yöntem tanımlamak için kullanılır. Annotation'un içindeki URL, silinecek kaynağın yolunu belirtir.

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

@Path

- **@Path** annotation'u, URL'deki parametreleri yöntem parametrelerine eşleştirmek için kullanılır. Annotation'un içindeki değer, URL'deki parametre adıyla eşleşmelidir.

```
@GET("/users/{id}")
suspend fun getUser(@Path("id") userId: Int): User

@POST("/users/{name}")
suspend fun createUser(@Path("name") userName: String): User
```

@Query

- **@Query** annotation'u, URL'deki sorgu parametrelerini yöntem parametrelerine eşleştirmek için kullanılır. Annotation'un içindeki değer, sorgu parametre adıyla eşleşmelidir.

```
@GET("/users")
fun getUsers(@Query("page") page: Int, @Query("size") size: Int): Call<List<User>>
```

Üstteki kod örneklerinde coroutine yapısında var olan suspend keywordünü kullandık. Retrofit 2.6.0 sürümü ile birlikte coroutine uyumlu hale getirildi.

En Çok Kullanılan HttpStatus Kodları

200 OK: İstek başarıyla tamamlandı ve istenen veriler yanıt olarak gönderildi.

301 Moved Permanently: Kaynak kalıcı olarak yeni bir URL'ye taşındı. Tarayıcı ve istemci otomatik olarak yeni URL'ye yönlendirilir.

302 Found: Kaynak geçici olarak yeni bir URL'ye taşındı. Tarayıcı ve istemci yeni URL'ye yönlendirilir, ancak orijinal URL korunur.

400 Bad Request: İstekte bir hata var. Hata mesajı genellikle yanıt gövdesinde yer alır.

401 Unauthorized: İstek, kimlik doğrulama gerektiriyor ve kimlik bilgileri geçersiz veya eksik.

403 Forbidden: Kullanıcı, istenilen kaynağa erişmeye yetkili değil.

404 Not Found: İsteğe uyan bir kaynak bulunamadı.

405 Method Not Allowed: Kullanılan HTTP metodu, istenilen kaynak için izin verilmiyor.

415 Unsupported Media Type: Gönderilen içerik tipi, sunucu tarafından desteklenmiyor.

500 Internal Server Error: Sunucuda beklenmedik bir hata oluştu.

502 Bad Gateway: Sunucu, başka bir sunucudan aldığı bir isteği işlerken bir hata ile karşılaştı.

503 Service Unavailable: Sunucu şu anda kullanılamıyor ve istek daha sonra tekrarlanabilir.