

Emre Cura – Ödev 6

HTTP Method Çeşitleri ve Kullanım Alanları

GET metodu, sunucudan veri almak için kullanılır. Bu metod, bir web sayfası, bir API veya herhangi bir veri kaynağından veri çekmek için kullanılır. Örneğin, bir kullanıcı listesini veya ürün detaylarını almak için GET metodunu kullanabilirsiniz. GET isteği, sadece veri çekme işlemi yapar ve sunucuda herhangi bir değişiklik yapmaz.

POST metodu, sunucuya yeni veri eklemek için kullanılır. Örneğin, bir formu doldurup gönderdiğinizde veya bir dosya yüklediğinizde POST metodu devreye girer. POST isteği, sunucuya veri gönderir ve genellikle bu veri sunucuda yeni bir kaynak oluşturur. Örneğin, yeni bir kullanıcı oluşturma işlemi POST metoduyla yapılır.

PUT metodu, var olan veriyi güncellemek için kullanılır. PUT isteği, belirtilen kaynağı günceller veya eğer kaynak mevcut değilse yeni bir kaynak oluşturur. Bu metod, bir kullanıcının bilgilerini güncellemek veya bir ürünün detaylarını değiştirmek için kullanılır.

DELETE metodu, sunucudan veri silmek için kullanılır. Bu metod, belirli bir kaynağı veya veri parçasını sunucudan kaldırmak için kullanılır. Örneğin, bir kullanıcıyı veya ürünü silmek için DELETE metodunu kullanabilirsiniz. DELETE isteği, sunucudaki belirli bir kaynağı kaldırır.

Retrofit İçerisinde Çok Kullanılan Anotasyonlar

@GET anotasyonu, HTTP GET isteği yapmak için kullanılır. Örneğin, kullanıcı listesini almak için **@GET("users")** şeklinde bir tanım yapılır ve bu metod, kullanıcı listesini döner.

@POST anotasyonu, HTTP POST isteği yapmak için kullanılır. Örneğin, yeni bir kullanıcı oluşturmak için **@POST("users")** şeklinde bir tanım yapılır ve bu metod, kullanıcı verisini sunucuya gönderir. Gönderilecek veri, genellikle **@Body** anotasyonu ile belirtilir.

@PUT anotasyonu, HTTP PUT isteği yapmak için kullanılır. Örneğin, bir kullanıcının bilgilerini güncellemek için **@PUT("users/{id}")** şeklinde bir tanım yapılır. Bu metod, kullanıcı id'sini ve güncellenecek veriyi alır.

@DELETE anotasyonu, HTTP DELETE isteği yapmak için kullanılır. Örneğin, bir kullanıcıyı silmek için **@DELETE("users/{id}")** şeklinde bir tanım yapılır. Bu metod, silinecek kullanıcı id'sini alır ve belirtilen kullanıcıyı siler.

@Path anotasyonu, URL içindeki değişkenleri belirtmek için kullanılır. Örneğin, belirli bir kullanıcıyı almak için **@GET("users/{id}")** şeklinde bir tanım yapılır ve **{id}** değeri metod parametresi olarak geçer.

@Query anotasyonu, URL'e sorgu parametreleri eklemek için kullanılır. Örneğin, belirli bir yaş grubundaki kullanıcıları almak için **@GET("users")** metoduna **@Query("age")** eklenir.

@Body anotasyonu, HTTP isteğinin gövdesini belirtmek için kullanılır. Örneğin, yeni bir kullanıcı oluşturmak için gönderilecek veriyi **@Body** ile tanımlarsınız.

En Çok Kullanılan HTTP Durum Kodları ve İşlevleri

200 OK durumu, bir HTTP isteğinin başarıyla tamamlandığını belirtir. Bu durum kodu, GET, POST, PUT ve DELETE isteklerinin başarıyla sonuçlandığını gösterir. İstek başarılı bir şekilde işlendiğinde ve istenen veri döndüğünde kullanılır.

201 Created durumu, sunucunun yeni bir kaynak oluşturduğunu belirtir. Bu durum kodu, genellikle POST istekleri ile ilişkilidir. Örneğin, yeni bir kullanıcı oluşturulduğunda sunucu bu kodu döner.

204 No Content durumu, isteğin başarıyla tamamlandığını ancak döndürülecek veri olmadığını belirtir. Bu durum kodu, genellikle DELETE isteklerinden sonra kullanılır.

400 Bad Request durumu, isteğin geçersiz olduğunu ve sunucunun isteği işleyemediğini belirtir. Bu kod, genellikle yanlış formatta veri gönderildiğinde veya zorunlu parametreler eksik olduğunda kullanılır.

401 Unauthorized durumu, kimlik doğrulamanın gerekli olduğunu ve başarısız olduğunu belirtir. Bu durum kodu, kullanıcının yetkilendirme bilgilerinin eksik veya yanlış olduğunu gösterir.

403 Forbidden durumu, isteğin anlaşıldığını ancak yetkilendirilmediğini belirtir. Bu durum kodu, kullanıcının belirli bir kaynağa erişim yetkisinin olmadığını gösterir.

404 Not Found durumu, istenen kaynağın bulunamadığını belirtir. Bu kod, URL'nin yanlış olduğu veya kaynağın mevcut olmadığı durumlarda kullanılır.

500 Internal Server Error durumu, sunucuda beklenmedik bir hata meydana geldiğini belirtir. Bu kod, sunucu tarafında bir hata oluştuğunda döner ve genellikle sunucu tarafında bir sorunu işaret eder.