

View.OnApplyWindowInsetsListener Kullanımı

OnApplyWindowInsetsListener, bir görünümde pencere kenar boşluklarını özelleştirilmiş bir şekilde uygulamak için kullanılan dinleyicidir.

Pencere kenar boşlukları, Android cihazlarda sistem arayüz elemanlarının (durum çubuğu, gezinme çubuğu gibi) uygulama içerik alanı üzerinde kapladığı yerleri temsil eder. Özellikle tam ekran modları veya şeffaf durum çubukları gibi özellikler kullanıldığında, bu kenar boşlukları uygulama tasarımını doğrudan etkileyebilir.

Android'de **setOnApplyWindowInsetsListener** isimli bir metod kullanılarak, bu kenar boşlukları ile ilgili bilgileri yakalayan ve bu bilgilere göre belirli işlemler gerçekleştiren bir "dinleyici" tanımlanabilir. Bu dinleyici, sistemin varsayılan davranışını değiştirmek için kullanılabilir. Örneğin, bir görünümün kenar boşluklarını (padding) dinamik olarak ayarlamak ya da belirli bir görünümde kenar boşluklarını yok sayarak tam ekran deneyimi sağlamak için bu özellikten yararlanılabilir.

Bir uygulama geliştiricisi, View'in varsayılan **onApplyWindowInsets** metodundan farklı, özelleştirilmiş bir davranış uygulamak istiyorsa, bir **OnApplyWindowInsetsListener** arayüzü uygulayabilir. Bu arayüz set edildiğinde, View'un kendi **onApplyWindowInsets** metodunun yerine, dinleyicinin **onApplyWindowInsets** metodu çağrılır. Bu, geliştiricilere, sistem UI elemanlarına göre görünümün nasıl tepki vermesi gerektiğini tamamen kontrol etme yeteneği verir.

setOnApplyWindowInsetsListener

```
public static void setOnApplyWindowInsetsListener(  
    @NonNull View view,  
    @Nullable OnApplyWindowInsetsListener listener  
)
```

Parameters

@NonNull View view:

Burada @NonNull ifadesi, view parametresinin boş olamayacağını (null olamayacağını) belirtir. Yani bu parametre için bir View nesnesi kesinlikle sağlanmalıdır.

@Nullable OnApplyWindowInsetsListener listener :

Uygulanan pencere kenar boşlukları (window insets) için kullanılacak izleyici (listener). @Nullable ifadesi ise bu listener parametresinin boş (null) olabileceğini belirtir. Yani bu izleyiciyi sağlamak isteğe bağlıdır; eğer bir izleyici sağlanmazsa, bu özellik kullanılmayacaktır.

Yeni Gelen Güncelleme Sonrası onCreate Fonksiyonu

Son güncelleme ile birlikte, Android uygulamalarının ekran tasarımı ve kullanıcı deneyimi üzerinde daha fazla kontrol sağlamak amacıyla `setOnApplyWindowInsetsListener` fonksiyonu ve "Edge-to-Edge" içerik gösterimi özelliği ön plana çıkmıştır.

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    enableEdgeToEdge()  
    setContentView(R.layout.activity_profile)  
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
        val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
        insets ^setOnApplyWindowInsetsListener  
    }  
}
```

Şekil 1. Güncellenmiş onCreate Fonksiyonu

Edge-to-Edge (Enable to Edge)

"Edge-to-Edge" ya da ekranın kenarlarına kadar uzanan içerik gösterimi, Android'in modern versiyonlarında daha popüler hale gelmiştir. Bu özellik, uygulama içeriğinin ekranın tamamını kaplamasına, böylece daha daldırıcı bir deneyim sunmasına olanak tanır. Özellikle gezinme çubukları ve durum çubukları şeffaf hale getirildiğinde, içerikler bu alanlara kadar genişleyebilir. Bu mod aktif edildiğinde, uygulamanın içeriği sistemin durum ve gezinme çubuklarının altına, yani cihazın fiziksel kenarlarına kadar yayılabilir.



Şekil 2. Enable to Edge pasif



Şekil 3. Enable to Edge aktif

setOnApplyWindowInsetsListener

Android'deki bir birleştirme ViewCompat.setOnApplyWindowInsetsListener işlevini kullanarak, belirli bir görünüm için pencere kenarlarını (pencere ekleri) özelleştirilmiş bir şekilde işlemektedir. İşte bu kodun adım adım açıklaması:

1. ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -> ... }: Bu satır, R.id.main ID'sine sahip View için bir OnApplyWindowInsetsListener. Bu listener, sistem bu View'a uygulama planladığı kenar çerçevelerini (pencere kenar çerçeveleri) yakalar.

2. val systemBars=insets.getInsets(WindowInsetsCompat.Type.systemBars()):

BuradagetInsets()metoduçağrılırveparametreolarak WindowInsetsCompat.Type.systemBars verilir. Bu işlem, sistem çubukları tarafından (durum çubuğu ve navigasyon çubuğu gibi) kullanılan çözümlerin çözümüne döner. systemBars değişkeni, bu çubukların her bir taraf için kenarlarını (sol, üst, sağ, alt) içeren bir nesneyi saklar.

3.v.setPadding(systemBars.left,systemBars.top,systemBars.right,systemBars.bottom): Bu satırda, elde edilen kenar bölümleri, View'un padding'ine uygulanır. Bu, İçeriği görüntüle, cihazın durum çubuğu ve navigasyon çubuğu gibi sistem UI elemanları tarafından yapılandırılmaması için yapılır. Örneğin, eğer durum çekerken 24 piksel büyüklüğündeyse, View'un üst padding'i de 24 piksel olarak bölümlere ayrılır, böylece içerik durum parametrelerin altında başlar.

4. insets: Bu, onApplyWindowInsets yönteminin dönüş değeri olarak insets nesnesini olduğu gibi geri döner. Bu, işlenen eklerin başka şekillerde kullanılmasına izin verir. Yani, bu dinleyicinin bu ekleri kullanıp iş saklaması sonrasında, sistem bu eklerin bilgilerini daha fazla işlemek üzere başka görünümlere de sağlayabilir.



*Şekil 4. Enable to Edge pasif ve
setOnApplyWindowInsetsListener aktif*



*Şekil 5. Enable to Edge aktif ve
setOnApplyWindowInsetsListener aktif*

Bu özellikler, özellikle medya oynatma veya görsel zenginlik gerektiren uygulamalarda, kullanıcıların içeriği daha keyifli ve etkili bir şekilde tüketmelerini sağlar. Son güncelleme bu iki önemli özelliği geliştirerek Android platformunda uygulama geliştiricilerine, kullanıcı arayüzü üzerinde daha derinlemesine bir hakimiyet ve esneklik sunmuştur.