setOnApplyWindowInsetsListener() Nedir, Ne İçin Kullanılır?

Android Studio'ya gelen yeni güncelleme ile varsayılan onCreate() metoduna bazı farklılıklar oldu.

```
package com.commonsware.jetpack.hello

import ...

class MainActivity : AppCompatActivity() {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
  }
}
```

Eski Varsayılan onCreate

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        enableEdgeToEdge()

        setContentView(R.layout.activity_main)

        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
        }
    }
}
```

Yeni Varsayılan onCreate

Görsel 1

Şimdi bu farklılıklara bir göz atalım:

enableEdgeToEdge():

Bu fonksiyon activity'nin layout'a set edilmeden önce çağırılıyor. Bunun nedeni activity'nin layout'unu oluşturmadan önce ekranın kenarlarına yönelik sistem çubuklarının ve kenar boşluklarının (system bars ve insets) görünümünü düzenlemek için gereken ayarlamaları yapmaktır. Daha sonra **setContentView()** çağırılır. Kısacası bu fonksiyon uygulamanın cihaz ekranının tüm genişliği ve yüksekliğinin kullanılmasını sağlıyor.

Insets Nedir?

Insets, ekranın kenarlarındaki sistem bileşenleri tarafından kaplanan veya kullanılan alanları belirtir.

setOnApplyWindowInsetsListener():

Android'de ekranın kenarlarında (sistem çubukları gibi) yer alan bileşenlerin boyutları değiştiğinde bir View'in içeriğinin bu değişikliklere uygun olarak ayarlanmasını sağlar.

Adım adım **setOnApplyWindowInsetsListener()** kullanılan kodu inceleyelim:

ViewCompat:

```
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
}
```

Görsel 2

ViewCompat View içerisindeki özelliklere erişim için kullanılan bir sınıf. Bu sınıf sayesinde setOnApplyWindowInsetsListener() erişim sağlıyoruz.

```
public static void setOnApplyWindowInsetsListener(@NonNull final View view,
                                                   final @Nullable OnApplyWindowInsetsListener listener) {
    if (Build.VERSION.SDK_INT >= 21) {
        Api21Impl.setOnApplyWindowInsetsListener(view, listener);
    }
}
```

Görsel 3

ViewCompat sınıfınıdaki **setOnApplyWindowInsetsListener()** içeriğine baktığımızda bir view ve bir OnApplyWindowListener aldığını görüyoruz. Görsel 3'e baktığımızda view olarak **findViewById(R.id.main)** verildiğini görüyoruz. OnApplyWindowListener olarak lamda gösterimine gidilip **v , insets ->** verilmiş.

```
public interface OnApplyWindowInsetsListener {
    When set on a View, this listener method will be called instead of the view's own onApplyWindowInsets
    method.
    Params: v – The view applying window insets
            insets – The insets to apply
    Returns: The insets supplied, minus any insets that were consumed
    @NonNull
    WindowInsetsCompat onApplyWindowInsets(@NonNull View v, @NonNull WindowInsetsCompat insets);
}
```

Görsel 4

Görsel 4'e bakarak v değerinin bir **View**, insets değerinin bir **WindowInsetsCompat** olduğunu anlayabiliriz.

Tekrardan Görsel 2'ye dönelim: **systemBars** WindowInsets'ten sistem çubuklarına ilişkin boyutları alır. **v.setPadding** View'in içeriğini sistem çubukları tarafından kaplanan alan kadar içeriğe kaydırır. Bu sayede içeriğin sistem çubukları tarafından kapatılmasını engeller.

Son olarak metodun dönüş tipi WindowInsetsCompat olduğu için insets return edilmiş.

Tekrardan Görsel 3'e geri dönersek, Eğer SDK 21 den büyükse verdiğimiz view ve listener **setOnApplyWindowInsetsListener()** metodunu tetikler. Şimdi bu metodu inceleyelim:

```
@DoNotInline
static void setOnApplyWindowInsetsListener(final @NonNull View v,
                                           final @Nullable OnApplyWindowInsetsListener listener) {
    // For backward compatibility of WindowInsetsAnimation, we use an
    // OnApplyWindowInsetsListener. We use the view tags to keep track of both listeners
    if (Build.VERSION.SDK_INT < 30) {
        v.setTag(R.id.tag_on_apply_window_listener, listener);
    }

    if (listener == null) {
        // If the listener is null, we need to make sure our compat listener, if any, is
        // set in-lieu of the listener being removed.
        View.OnApplyWindowInsetsListener compatInsetsAnimationCallback =
                (View.OnApplyWindowInsetsListener) v.getTag(
                        R.id.tag_window_insets_animation_callback);
        v.setOnApplyWindowInsetsListener(compatInsetsAnimationCallback);
        return;
    }

    v.setOnApplyWindowInsetsListener(new View.OnApplyWindowInsetsListener() {
        WindowInsetsCompat mLastInsets = null;

        @Override
        public WindowInsets onApplyWindowInsets(final View view,
                                                final WindowInsets insets) {
            WindowInsetsCompat compatInsets = WindowInsetsCompat.toWindowInsetsCompat(
                    insets, view);
            if (Build.VERSION.SDK_INT < 30) {
                callCompatInsetAnimationCallback(insets, v);

                if (compatInsets.equals(mLastInsets)) {
                    // We got the same insets we just return the previously computed insets.
                    return listener.onApplyWindowInsets(view, compatInsets)
                            .toWindowInsets();
                }
            }
            mLastInsets = compatInsets;
            compatInsets = listener.onApplyWindowInsets(view, compatInsets);

            if (Build.VERSION.SDK_INT >= 30) {
                return compatInsets.toWindowInsets();
            }

            // On API < 30, the visibleInsets, used to built WindowInsetsCompat, are
            // updated after the insets dispatch so we don't have the updated visible
            // insets at that point. As a workaround, we re-apply the insets so we know
            // that we'll have the right value the next time it's called.
            requestApplyInsets(view);
            // Keep a copy in case the insets haven't changed on the next call so we don't
            // need to call the listener again.

            return compatInsets.toWindowInsets();
        }
    });
}
```

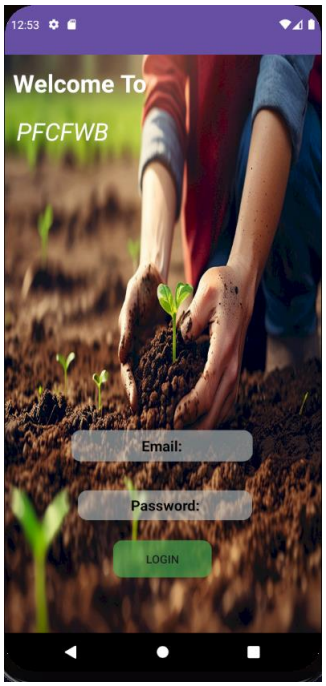
Görsel 5

Fonksiyon içinde, öncelikle belirli koşullara göre **OnApplyWindowInsetsListener** parametresi View'e atanır **onApplyWindowInsets** metodu, WindowInsets olayı gerçekleştiğinde çağrılır ve View'in içeriğini bu olaya göre günceller. Farklı API sürümlerine göre metod farklı durumları ve kontrolleri içeriyor. Son olarak fonksiyon içinde belirli işlemler yapılır ve WindowInsets olayı gerçekleştiğinde View'in içeriği dinamik olarak güncellenir.

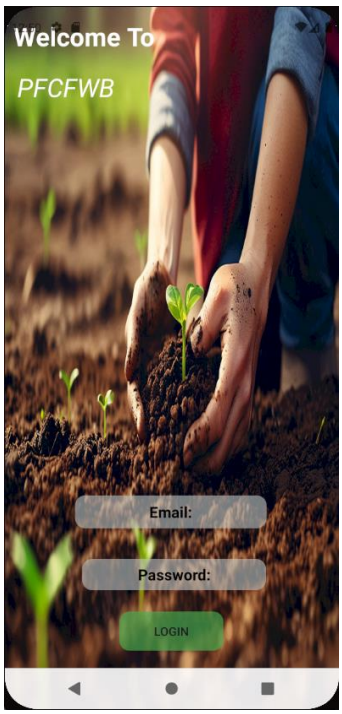
Özet:

enableEdgeToEdge() fonksiyonu kullanımı sonrası UI bileşenlerinden ekrana yakın kısımda olanlar status bar veya navigation bar ile çakışabilir. Bu engellemek için **setOnApplyWindowInsetsListener()** metodunu kullanarak WindowInsets'ten sistem çubuklarına padding veriliyor. Bu sayede bu çakışmayı engellemiş olunuyor.

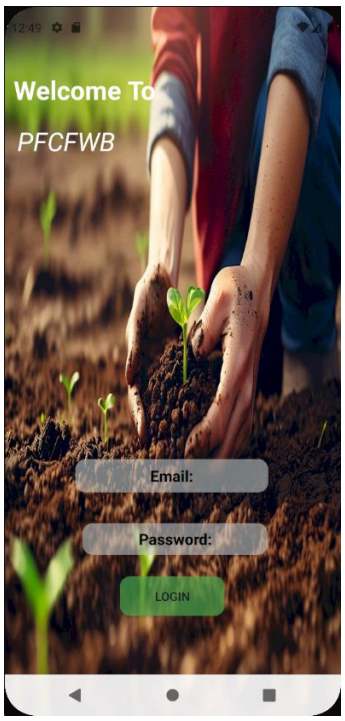
***enableEdgeToEdge*() ve setOnApplyWindowInsetsListener()** kullanılmamış**:**



***enableEdgeToEdge*()** kullanılmış, **setOnApplyWindowInsetsListener()** kullanılmamış:



***enableEdgeToEdge*()** ve
setOnApplyWindowInsetsListener() kullanılmış



Yusuf Mücahit Solmaz

Kaynakça:

https://developer.android.com/develop/ui/views/layout/edge-to-edge?authuser=2

https://developer.android.com/develop/ui/views/layout/edge-to-edge?source=post_page-----ce8b59769ed9----------------------------#kotlin

https://medium.com/androiddevelopers/windowinsets-listeners-to-layouts-8f9ccc8fa4d1

https://www.tabnine.com/code/java/methods/android.view.View/setOnApplyWindowInsetsListener

https://medium.com/@rawatsumit115/how-to-detect-visibility-and-height-of-the-keyboard-using-window-insets-in-android-kotlin-e3d8b7d2bc56

https://developer.android.com/reference/android/view/View.OnApplyWindowInsetsListener?source=post_page-----ce8b59769ed9----------&authuser=2