

# Anotacion EView

La documentación de esta anotación la veremos en el siguiente enlace:

<https://github.com/androidannotations/androidannotations/wiki/Enhance-custom-views#custom-views-with-eview>

La anotación "@EView" permite crear un nuevo objeto custom de UI. Para usarla deberemos crear una clase que herede de "View". Por ejemplo:

```
@EView
public class CustomButton extends android.support.v7.widget.AppCompat.Button {

    public CustomButton(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

}
```

## (1) Uso de la anotación EView

Crea un nuevo paquete en "com.example.openwebinar.enhancedcomponents" y lo llamaremos "views". En este path crearemos una nueva clase Java que la llamaremos "CustomButton" y luego crearemos otra diferente llamada "CustomTextView". La primera clase será:

```
@EView
public class CustomButton extends android.support.v7.widget.AppCompat.Button {
```

La segunda será:

```
@EView
public class CustomTextView extends android.support.v7.widget.AppCompat.TextView {
```

Estas clases únicamente tendrán el constructor y tendrán lo siguiente

**CustomButton:** "setText("My Custom Button");"

**CustomTextView:** "setText("Random octal (" + octal + ")");" // (octal will be a random octal number)

Seguidamente crearemos un nuevo layout al cual llamaremos "activity\_fourth.xml" y añadiremos los custom object creado previamente, recuerda que se debe emplear la clase generada por AndroidAnnotation:

```
<com.example.openwebinar.enhancedcomponents.views.CustomTextView_
    android:id="@+id/customTextView_"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

        app:layout_constraintBottom_toTopOf="@+id/customButton_"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<com.example.openwebinar.enhancedcomponents.views.CustomButton_
    android:id="@+id/customButton_"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="444dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/customTextView_" />

```

Llama a esta nueva actividad desde "MainActivity" usando "setContentView".

## **(2) Test it!**

Lanza la aplicación en el emulador y podrás ver el nuevo objeto de UI.