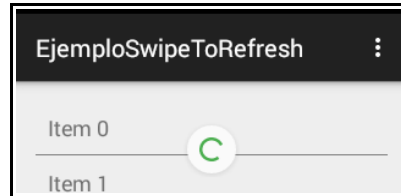
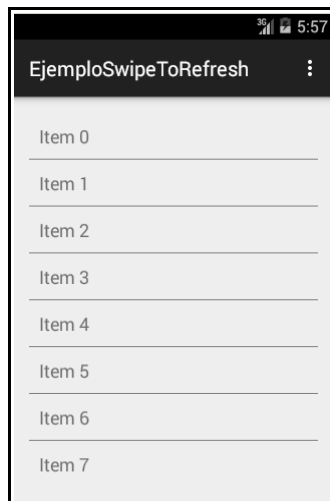


Swipe to refresh

Mediante este widget, podremos agregar el patrón de diseño mediante el cual el usuario hace un gesto hacia abajo de una lista y al soltar, ésta se recarga, mostrando un ícono animado de progreso.



Suponemos que nuestro proyecto posee una lista construida con un RecyclerView.



Ahora podemos "envolver" al recyclerView del archivo xml de layout con el widget SwipeRefreshLayout

```
<android.support.v4.widget.SwipeRefreshLayout
    android:id="@+id/swipe"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</android.support.v4.widget.SwipeRefreshLayout>
```

Luego implementamos la interface OnRefreshListener para escuchar el evento de refresh

```
public class MainActivity extends AppCompatActivity implements
    SwipeRefreshLayout.OnRefreshListener {
```

Mediante la cual estamos obligados a implementar el método `onRefresh`:

```
@Override
public void onRefresh() {
    Log.d("activity", "Refresh list");
}
```

Aquí es donde generalmente se lanzará un request a internet para obtener más información para cargar en la lista, o para actualizar la existente.

Para conectar el listener del evento con el widget `SwipeRefreshLayout`, en el método `onCreate` de la Activity, ejecutamos `setOnRefreshListener`, pasándole "this" en este caso.

```
private SwipeRefreshLayout s;

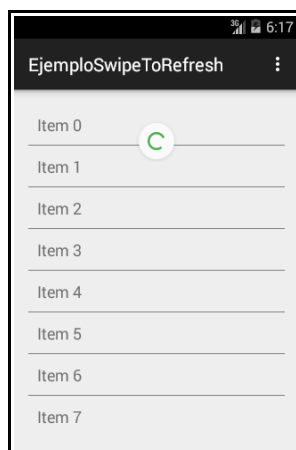
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    s = (SwipeRefreshLayout) findViewById(R.id.swipe);
    s.setOnRefreshListener(this);
    s.setColorSchemeResources(R.color.red, R.color.green, R.color.blue);
}
```

Los colores pasados como recursos, deberemos definirlos en el archivo `colors.xml` que deberemos crear en la carpeta `res/values`, con el siguiente contenido:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="red">#f44336</color>
    <color name="green">#4caf50</color>
    <color name="blue">#03a9f4</color>
</resources>
```

Cuando el usuario realice el gesto, aparecerá en pantalla el ícono de progreso, el cual quedará de manera indeterminada hasta que ejecutemos el método `setRefreshing` pasándole el valor `false`.



En este ejemplo, simularemos la cancelación del estado de “cargando” presionando el botón de back, para ello, sobrescribimos en la Activity el siguiente método:

```
@Override
public void onBackPressed() {
    if(s.isRefreshing())
        s.setRefreshing(false);
    else
        super.onBackPressed();
}
```

Al presionar back, si el widget `SwipeRefreshLayout` se encuentra mostrando la animación, el método `isRefreshing` devuelve true, por lo que seteamos el estado en false mediante `setRefreshing`.