


Open · Opened 3 days ago by  **Rubén Montero**

Mostrar el progreso

Resumen

- Mostraremos una *ProgressBar* mientras la petición de `/clips` está activa

Descripción

Uno de los principios básicos creando aplicaciones Android es dar *feedback visual* al usuario. No puede ser que nuestra aplicación esté esperando por la respuesta de un servidor y el usuario no lo sepa. ¡Podría pensar que la *app* se ha congelado!

Un [ProgressBar](#) es un componente ideal para mostrar al usuario la idea de *algo está en progreso*.

Puede mostrar una barra que avanza, o bien un *círculo* que gira indefinidamente.

La tarea

Añade un `<ProgressBar>` en `activity_main.xml`:

```
<ProgressBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

También, en dicho `<ProgressBar>`, **añade** los *atributos* necesarios para que:

- Esté centrado horizontalmente.
- Esté posicionado verticalmente como tú desees (arriba, en el centro, abajo...)
- Tenga un `android:id` (elígelo libremente)

A continuación, **añade** en `MainActivity.java` un atributo privado `ProgressBar` e **inicialízalo** en `onCreate` con `findViewById`, como se indica a continuación:

```
public class MainActivity extends AppCompatActivity {
    private RequestQueue queue;
    private Context context = this;
    private ConstraintLayout mainLayout;
+   private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.queue = Volley.newRequestQueue(context);
        this.mainLayout = findViewById(R.id.main_layout);
+       this.progressBar = findViewById( /* El ID que tú hayas decidido */ );

        // Solicitamos una lista de clips
        // ...
    }
}
```

setVisibility

Usaremos `.setVisibility` para mostrar u ocultar el `ProgressBar`.

Al principio del método `requestCliList` en `MainActivity.java`, **añade** la línea `progressBar.setVisibility(View.VISIBLE)`:

```
private void requestCliList() {
    progressBar.setVisibility(View.VISIBLE);
    // ...
}
```

Y después, en `onResponse` y `onErrorResponse`, **invoca** nuevamente `setVisibility`, esta vez indicando `View.INVISIBLE`:

```
@Override
```

```
public void onResponse(JSONArray response) {  
    progressBar.setVisibility(View.INVISIBLE);  
    // ...  
}
```

```
@Override  
public void onErrorResponse(VolleyError error) {  
    progressBar.setVisibility(View.INVISIBLE);  
    // ...  
}
```

¡Listo! Ahora tienes un `ProgressBar` que se muestra inicialmente cuando la *app* lanza una petición HTTP a <http://raspi:8000/clips> y se oculta cuando dicha petición se completa.

Si ejecutas la *app*, verás fugazmente aparecer el `ProgressBar`. También puedes [configurar un delay de red en el emulador](#).

¿Y qué hay de la petición a `/health`?

La petición HTTP que configuramos en las tareas iniciales vamos a dejarla como está. Aunque sería una buena práctica, no vamos a preocuparnos de añadir un `ProgressBar` para esa petición.

Por último

Sube tus cambios al repositorio en un nuevo *commit*.



Rubén Montero @ruben.montero changed milestone to [%Sprint 2](#) 3 days ago