


Open Opened 2 days ago by  **Rubén Montero**

¿Es un pájaro? ¿Es un avión? ¿Es un botón!

Resumen

- Crearemos un `<FloatingActionButton>` en nuestra `activity_status.xml`
- Desde el código Java, haremos que muestre un diálogo incompleto al pulsarse

Descripción

Vamos a añadir la funcionalidad de *Cambiar el estado*. Así, los usuarios podrán almacenar un estado distinto al "Hey there! I'm using Sessions" por defecto.

El objetivo final es que los usuarios puedan mandar una petición HTTP `PUT` como la del siguiente ejemplo:

```
PUT /users/pepe.depura/status
Session-Token: a26cbe5ead

{
  "status": "Estado actualizado! Hurra!"
}
```

En esta tarea, de momento, no se mandará dicha petición. Sólo comenzaremos a trabajar en la parte visual.

Como novedad, aprenderemos a mostrar un *diálogo* al usuario. Existen [varias formas de mostrar diálogos](#) al usuario. Nosotros vamos a optar por la clase `AlertDialog`, ya que con poco código la tendremos funcionando.

La tarea

Primero, **añade** un `<FloatingActionButton>` en `activity_status.xml`:

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/button_open_dialog"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:src="@android:drawable/ic_input_add"
    android:layout_margin="15dp"
/>
```

StatusActivity.java

Luego, en `StatusActivity.java`, **añade** un atributo tipo `FloatingActionButton` e **inicialízalo** debidamente en `onCreate`. También, **asígnale** un `OnClickListener`:

```
public class StatusActivity extends AppCompatActivity {
    /* ... */
    private FloatingActionButton buttonChangeStatus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        /* ... */
        buttonChangeStatus = findViewById(R.id.button_open_dialog);
        buttonChangeStatus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
    }
}
```

Queremos que se muestre un diálogo y permita al usuario introducir un texto. El diálogo se mostrará cuando el usuario haga *click* en el botón flotante.

Lamentablemente, *no* podemos hacer `new` directamente:

```
@Override
public void onClick(View view) {
    // Esto NO funciona
    AlertDialog myDialog = new AlertDialog();
}
```

Así que... **usa** la clase `AlertDialog.Builder` así:

```
@Override
public void onClick(View view) {
    AlertDialog.Builder myBuilder = new AlertDialog.Builder(context);
}
```

Completa el código dentro de `onClick` de esta manera, para definir algunas propiedades del *diálogo*:

```
@Override
public void onClick(View view) {
    AlertDialog.Builder myBuilder = new AlertDialog.Builder(context);
    myBuilder.setPositiveButton("Modificar", null); // Esto añade un botón al diálogo
    AlertDialog myDialog = myBuilder.create(); // Esta línea es como 'new AlertDialog'
    myDialog.show();
}
```

(Como ves, hay que invocar `.show()`; sobre la instancia de `AlertDialog` para mostrarlo).

¡Puedes verificar que aparece un diálogo al pulsar en el botón flotante!

Por último

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



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