

Displaying Toasts

[Edit Page](#)[Page History](#)

Jump to Section

- Overview
- Simple Toast
- Custom Toast
- References

Overview

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout.



Simple Toast

First, instantiate a Toast object with one of the `makeText()` methods. This method takes three parameters: the application Context, the text message, and the duration for the toast.

```
// also supports Toast.LENGTH_LONG
Toast.makeText(getApplicationContext(), "some message", Toast.LENGTH_SHORT).show();
```

You can configure the position of a Toast. A standard toast notification appears near the bottom of the screen, centered horizontally. You can change this position with the `setGravity` method and specifying a Gravity constant.

```
Toast toast = Toast.makeText(getApplicationContext(), "some message", Toast.LENGTH_SHORT);
toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
toast.show();
```

Custom Toast

You can also create a Toast that uses a custom XML layout rather than just displaying plain text. First, simply define the XML view in `res/layout` in a file such as `toast_layout.xml`:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/toast_layout_root">
    <ImageView android:src="@drawable/droid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="8dp"
    />
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#FFF"
    />
</LinearLayout>
```

Notice that the ID of the `LinearLayout` element is `"toast_layout_root"`. You must use this ID to inflate the layout from the XML:

```
private void displayToast(String message) {
```

```
// Inflate toast XML layout
View layout = getLayoutInflater().inflate(R.layout.toast_layout,
                                           (ViewGroup) findViewById(R.id.toast_layout_root));
// Fill in the message into the textview
TextView text = (TextView) layout.findViewById(R.id.text);
text.setText(message);
// Construct the toast, set the view and display
Toast toast = Toast.makeText(getApplicationContext(), "some message", Toast.LENGTH_SHORT);
toast.setView(layout);
toast.show();
}
```

And then you can display the custom toast using `displayToast("Message");` .

References

- <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>
- <http://developer.android.com/reference/android/widget/Toast.html>
- <http://www.mkymong.com/android/android-toast-example/>