



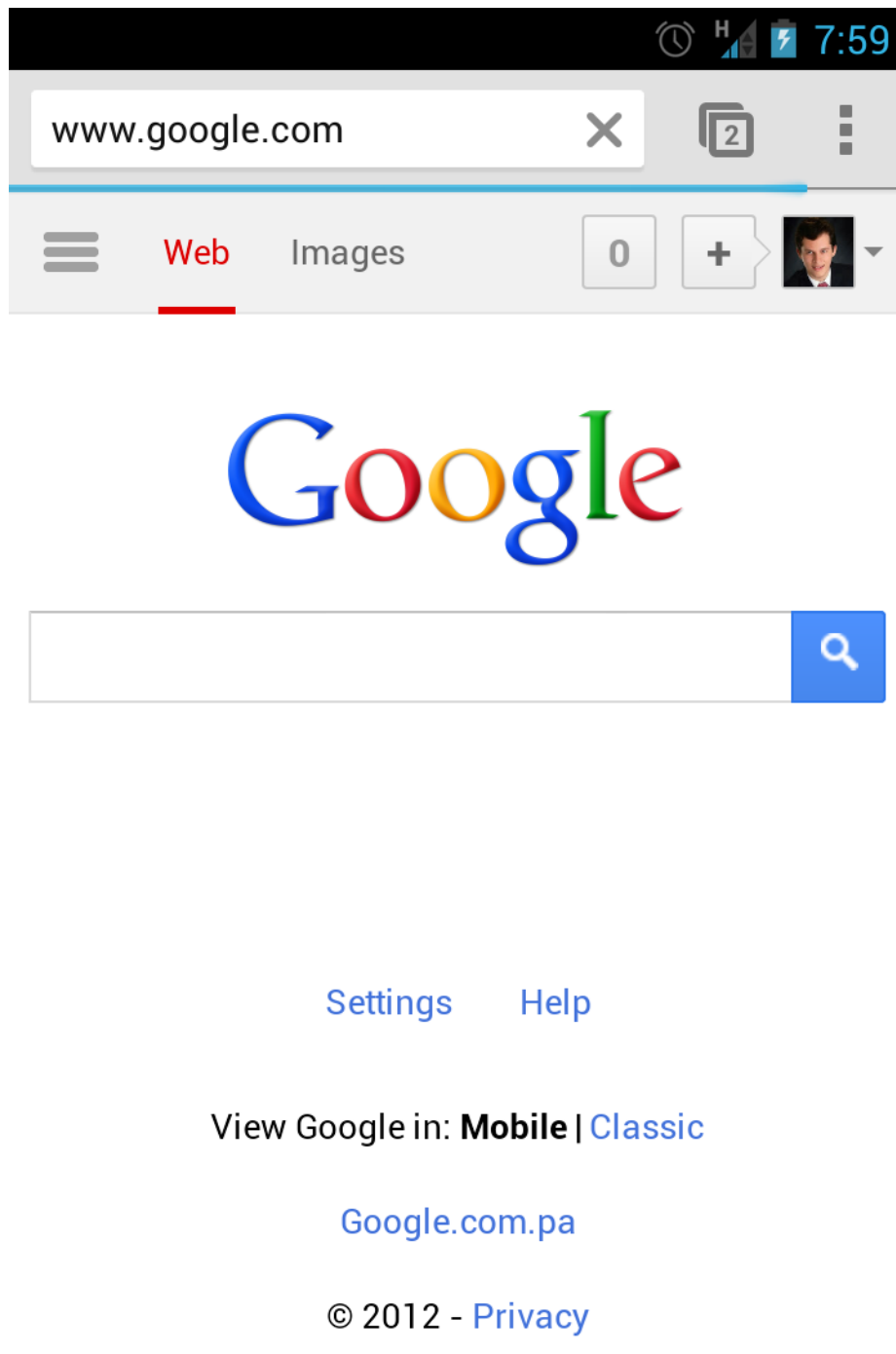
Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour ×

ProgressBar under Action Bar

Question Summary: How can I make a progressbar integrated inside the ActionBar, like on the Chrome App?

Details: Look at this screenshot from Chrome:



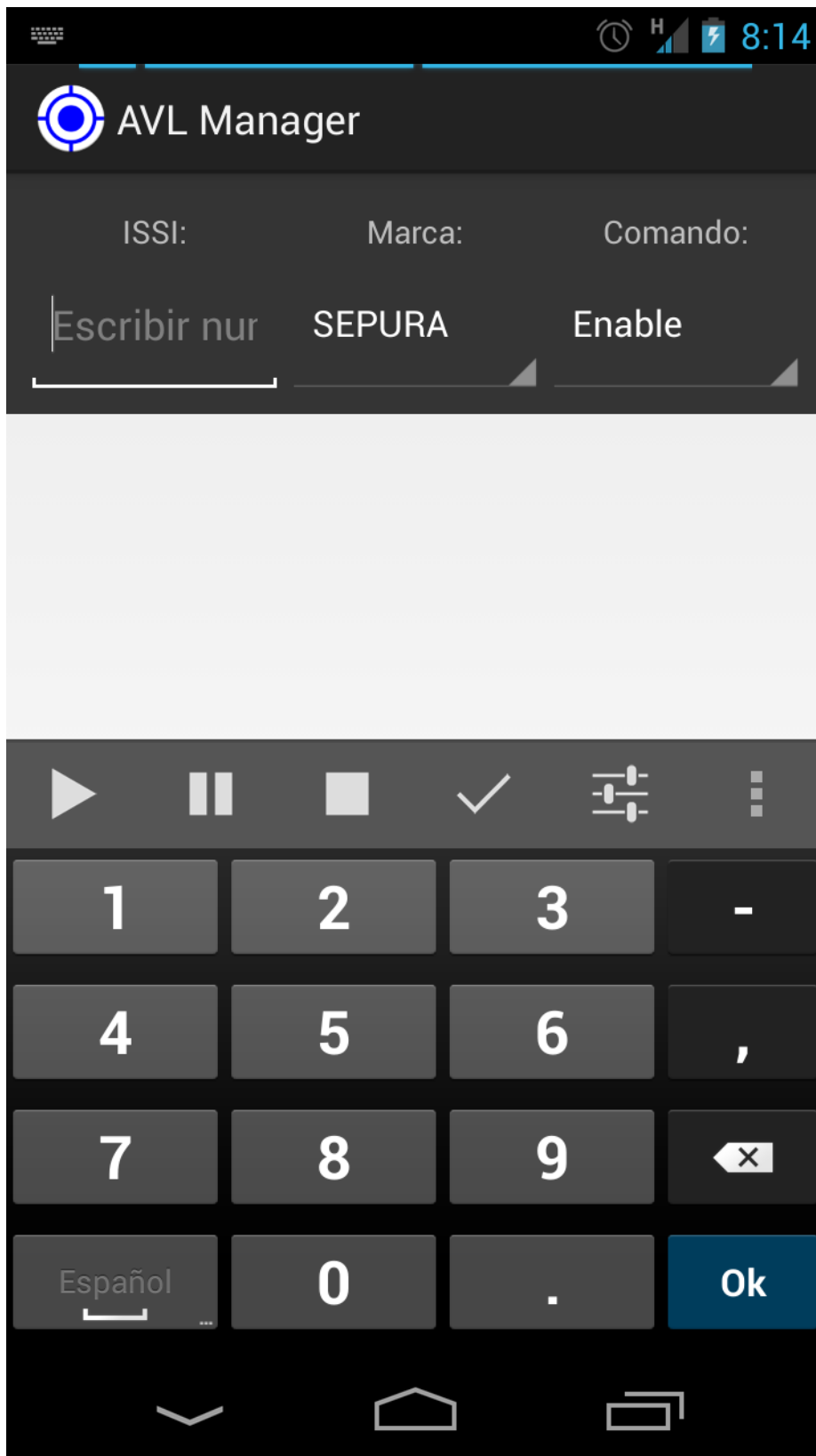
I want to create an Action bar just like this. Just under the action bar there's a ProgressBar that fills according to page load. I've seen this example from many apps, like Feedly, but I haven't been able to create my own implementation. I tried using Android's own APIs to create it:

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    //Request Permission to display the Progress Bar...
    this.requestWindowFeature(Window.FEATURE_PROGRESS);
    this.setWindowContentView(R.layout.activity_main)
    super.onCreate(savedInstanceState);

    this.setProgressBarIndeterminate(true);
}
```

But this code only causes the ProgressBar to show **over** the Action Bar, like so:



So, how can I make my ProgressBar appear under the Action Bar, like on the Chrome App?

 android [android-actionbar](#)

asked Dec 18 '12 at 13:24

 [zubietaroberto](#)
745 ● 9 ● 13

possible duplicate question of this answer : stackoverflow.com/a/10755545/794088 – [petey](#) Feb 25 '13 at 16:30

have you found the solution yet? – [Eric](#) Apr 1 '13 at 7:54

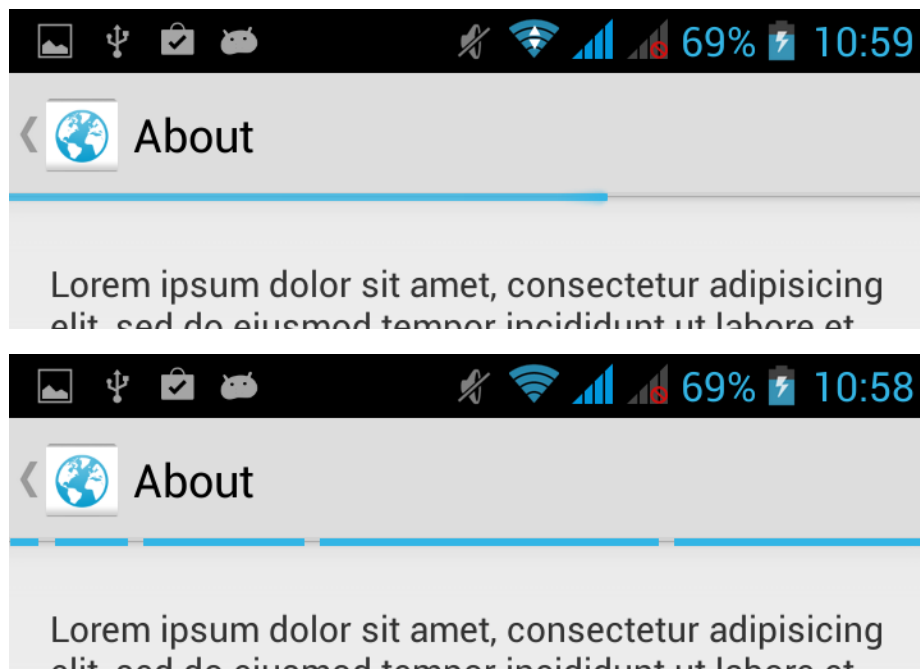
[add a comment](#)

5 Answers

I wasn't fully satisfied with the accepted answer above, so I did some extra research myself.

The trick I believe they used is that they retrieved the top view in the view hierarchy called `DecorView` and added the progress bar in there. That way, the progress bar displays both over the action bar AND the content area. Note that the S.D.'s answer puts the progress bar into content area and 'steals' space from actual content, what can lead to unexpected results.

Sample **screenshots** of this implementation:



Code

Just put this code into `onCreate` method of some activity and it should work:

```
// create new ProgressBar and style it
final ProgressBar progressBar = new ProgressBar(this, null, android.R.attr.progressBarStyleHorizontal);
progressBar.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT, 24));
progressBar.setProgress(65);

// retrieve the top view of our application
final FrameLayout decorView = (FrameLayout) getWindow().getDecorView();
decorView.addView(progressBar);

// Here we try to position the ProgressBar to the correct position by looking
// at the position where content area starts. But during creating time, sizes
// of the components are not set yet, so we have to wait until the components
// has been laid out
// Also note that doing progressBar.setY(136) will not work, because of different
// screen densities and different sizes of actionBar
ViewTreeObserver observer = progressBar.getViewTreeObserver();
observer.addOnGlobalLayoutListener(new OnGlobalLayoutListener() {
    @Override
    public void onGlobalLayout() {
        View contentView = decorView.findViewById(android.R.id.content);
        progressBar.setY(contentView.getY() - 10);

        ViewTreeObserver observer = progressBar.getViewTreeObserver();
        observer.removeGlobalOnLayoutListener(this);
    }
});
```

You can play with `LayoutParams`' height argument to set the progress bar wider or narrower, but you might have to adjust the `-10` offset.

Styling

Unfortunately, you can see the ugly gray background of the progress bar. To remove it, simply searching for the background by id and trying to hide it doesn't work. To remove the background, I had to create identical drawable of the system version and remove the background item.

TL;DR: Create file `progress_horizontal_holo_no_background_light.xml` and paste this drawable:

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@android:id/secondaryProgress">
        <scale android:scaleWidth="100%"
            android:drawable="@drawable/progress_secondary_holo_light" />
    </item>

    <item android:id="@android:id/progress">
        <scale android:scaleWidth="100%"
            android:drawable="@drawable/progress_primary_holo_light" />
    </item>

</layer-list>
```

Copy appropriate .png drawables from `sdk/platforms/android-xx/data/res/drawable-xxx/` to your project and then in the code you can add:

```
progressBar.setProgressDrawable(getResources().getDrawable(R.drawable.progress_holo_light));
```

Extra: Indeterminate Progress Bar

Pre-KitKat versions of indeterminate progress bars are pretty ugly and laggy. You can download new smooth progress bar called `ButteryProgressBar`. Just search for it on google (I cannot post any more links, because I am new here :[), add the class into your project and you can simply replace the previous `ProgressBar` with this code and have crispy indeterminate progress bar:

```
final ButteryProgressBar progressBar = new ButteryProgressBar(this);
progressBar.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT, 24));
```

You might also need to simplify this code:

```
final TypedArray ta = c.obtainStyledAttributes(attrs, R.styleable.ButteryProgressBar);
try {
    mBarColor = ta.getColor(R.styleable.ButteryProgressBar_barColor,
        c.getResources().getColor(android.R.color.holo_blue_light));
    mSolidBarHeight = ta.getDimensionPixelSize(
        R.styleable.ButteryProgressBar_barHeight,
        Math.round(DEFAULT_BAR_HEIGHT_DP * mDensity));
    mSolidBarDetentWidth = ta.getDimensionPixelSize(
        R.styleable.ButteryProgressBar_detentWidth,
        Math.round(DEFAULT_DETENT_WIDTH_DP * mDensity));
} finally {
    ta.recycle();
}
```

to this code:

```
mBarColor = c.getResources().getColor(android.R.color.holo_blue_light);
mSolidBarHeight = Math.round(DEFAULT_BAR_HEIGHT_DP * mDensity);
mSolidBarDetentWidth = Math.round(DEFAULT_DETENT_WIDTH_DP * mDensity);
```

Hope I helped :)

edited Jun 11 at 7:55

answered Jun 8 at 3:13



Antimonit

171 ● 1 ● 5

The method `removeGlobalOnLayoutListener(ViewTreeObserver.OnGlobalLayoutListener)` from the type `ViewTreeObserver` is deprecated – [Arjun U.](#) Jul 7 at 6:41

@ArjunU. what should we use instead> – [Patrick Geyer](#) Aug 18 at 21:59

@PatrickGeyer see stackoverflow.com/questions/16189525/... – [Arjun U.](#) Aug 20 at 6:51

[add a comment](#)

Extend Activities from following class to have a `ProgressBar` at the top(below `ActionBar`) of each, and a `getProgressBar()` method:

Parent class:

```

public abstract class ProgressActivity extends Activity {
    private ProgressBar mProgressBar;

    @Override
    public void setContentView(View view) {
        init().addView(view);
    }

    @Override
    public void setContentView(int layoutResID) {
        getLayoutInflater().inflate(layoutResID, init(), true);
    }

    @Override
    public void setContentView(View view, ViewGroup.LayoutParams params) {
        init().addView(view, params);
    }

    private ViewGroup init(){
        super.setContentView(R.layout.progress);
        mProgressBar = (ProgressBar) findViewById(R.id.activity_bar);
        return (ViewGroup) findViewById(R.id.activity_frame);
    }

    protected ProgressBar getProgressBar(){
        return mProgressBar;
    }
}

```

Layout (progress.xml):

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:orient
    android:layout_width="match_parent" android:layout_height="match_parent">
<ProgressBar android:id="@+id/activity_bar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="-8dp"
    style="@android:style/Widget.DeviceDefault.ProgressBar.Horizontal"
    />
<FrameLayout android:id="@+id/activity_frame"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    />
</LinearLayout>

```

edited Oct 15 '13 at 4:05

answered Feb 25 '13 at 18:19



S.D.
12.8k ● 16 ● 50

1 It works... for the most part. The ProgressBar does look separate from the actionBar. I had to add the `android:layout_marginTop="-8dp"` to the ProgressBar in order to simulate the "shadow" effect. But for this method, it is probably more efficient to add it to the main layout itself. Yet now it works as intended. Thanks for the hack. – [zubietaroberto](#) Jul 5 '13 at 17:04

@zubietaroberto Yes, You can add it to main layout if you are just using it for one activity. If you want multiple activities to have this feature, you can just extend the class shown above. – [S.D.](#) Oct 15 '13 at 4:05

Its work for me, but the progress bar appear under the tab host, how so i can make it above the tabhost and under action bar? – [DeckyFx](#) Dec 6 '13 at 8:06

[add a comment](#)

I have compiled code from this thread as well as other threads on StackOverflow and created a project that can be used to implement ButteryProgressbar as well as "pull down to refresh".
<https://github.com/pauldmpls/Gmail-like-Pull-Down-to-Refresh>

Feel free to use the code in your application.

A big thanks to you guys.

answered Jul 26 at 17:20



Shantanu Paul
63 ● 8

[add a comment](#)

Well , I did something similar for one of my pet apps and am not sure if that's the only way or the best way to

do it , but it definitely works.

To begin with , use an overlay action bar and then set the background drawable to "null" so that the overlay action bar is transparent.

Now in you activity layout , set the top margin for your root view to "actionbar height" You can do it like this.

```
<RelativeLayout
    ...
    android:layout_marginTop="?android:attr/actionBarSize" />
```

Now the action is not going to hide any of your activity content.

The next thing you have to do now is - add a header view on top of your root view with height same as the action bar. Set a background color to it. This would now be the color of your action bar and since the action bar aligns perfectly on top pf this header view.

Now you can easily put a progress bar in the header view and align it to the bottom of the header view. To the user this would look as if the progress bar is on the action bar itself , just like chrome ..

answered Apr 6 at 20:23



user2405598

61 ● 1 ● 3

[add a comment](#)

Please use below code. just use one default style in progress bar as `style="?android:attr/progressBarStyleHorizontal"`

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <ProgressBar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        style="?android:attr/progressBarStyleHorizontal"
        android:indeterminate="true"

    />

</RelativeLayout>
```

edited Aug 26 at 5:54



Mygod

47 ● 8

[add a comment](#)

answered May 12 at 12:47



Rank

329 ● 2 ● 11

Not the answer you're looking for? Browse other questions tagged [android](#)

[android-actionbar](#) or [ask your own question](#).