

Animation demo walk through

First: see the slide in [Tips on how to be a better programmer](#) about interfaces

1. Create an "Empty Activity" project.

2. Create a new Resource folder res/anim

3. Create an animation, call it rotate.xml (or whatever you want), put in the anim folder.

4. Replace rotate.xml content with the following;

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

    <rotate
        android:duration="1300"
        android:fromDegrees="0"
        android:interpolator="@android:anim/cycle_interpolator"
        android:pivotX="50%"
        android:pivotY="50%"
        android:repeatMode="restart"
        android:toDegrees="360" />

</set>
```

In activity main.xml

5. create a button and an imageview in the layout. For imageview make sure you set the source (find any png or jpg and drop in res\drawable) or use the ic_launcher icon.

```
android:src="@drawable/go"
```

6. make a doAnimate (or whatever you want to call it) handler for the button and create the handler in the java (remember the alt-enter trick for Android Studio)

```
android:onClick="doAnimate"
```

In MainActivity.java

7. Implement Animation.AnimationListener. Hover over red lines, hit alt-enter and select implement methods to autocreate interface methods.

```
public class MainActivity extends AppCompatActivity implements Animation.AnimationListener {
```

8. Get ref to imageView to rotate and load the animation

```
//class member vars
private Animation animRotate1;
private ImageView myView;

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

    //hold on t
    myView = (ImageView)findViewById(R.id.imageView);

    // load the animation
    animRotate1 = AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.rotate);
    //set listener (VERY IMPORTANT, YOU FORGET nothing works!)
    animRotate1.setAnimationListener(this);
}
```

9. finally start the animation (in the doAnimate() handler you created earlier).

```
public void doAnimate(View view) {
    //finally start the animation on the view
    myView.startAnimation(animRotate1);
}
```