**For Gradle users**

Open your build.gradle and make sure that Maven Central repository is declared into repositories section:

repositories {

mavenCentral()

}

Then, include the library as dependency:

compile 'com.github.gcacace:signature-pad:1.2.1'

defaultConfig {

……  
 multiDexEnabled true  
}

## Usage

Please see the */SignaturePad-Example* app for a more detailed code example of how to use the library.

1. Add the SignaturePad view to the layout you want to show.

<com.github.gcacace.signaturepad.views.SignaturePad

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/signature\_pad"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:penColor="@android:color/black"

/>

1. Configure attributes.

* penMinWidth - The minimum width of the stroke (default: 3dp).
* penMaxWidth - The maximum width of the stroke (default: 7dp).
* penColor - The color of the stroke (default: Color.BLACK).
* velocityFilterWeight - Weight used to modify new velocity based on the previous velocity (default: 0.9).
* clearOnDoubleClick - Double click to clear pad (default: false)

1. Configure signature events listener

An OnSignedListener can be set on the view:

mSignaturePad = (SignaturePad) findViewById(R.id.signature\_pad);

mSignaturePad.setOnSignedListener(new SignaturePad.OnSignedListener() {

@Override

public void onStartSigning() {

//Event triggered when the pad is touched

}

@Override

public void onSigned() {

//Event triggered when the pad is signed

}

@Override

public void onClear() {

//Event triggered when the pad is cleared

}

});

1. Get signature data

* getSignatureBitmap() - A signature bitmap with a white background.
* getTransparentSignatureBitmap() - A signature bitmap with a transparent background.
* getSignatureSvg() - A signature Scalable Vector Graphics document.