

你不知道一些神奇Android Api



wutongke (/u/0e0821e94979) [+ 关注](#)

2017.01.15 15:21* 字数 995 阅读 8798 评论 28 喜欢 337

(/u/0e0821e94979)

这将是一个关于Android Api的系列文章，AntSoft的Android 团队一直紧随技术前沿，在 *Budapest University of Technology and Economics* 培训Android技术已经有8年多的时间。公司里有个传统就是每周进行技术分享，这里将介绍一些Android平台上有意思的API。

当前Android已经有了非常多可用的依赖库（Library），但其实Android platform的一些API有些鲜为人知，但非常有用的方法和类，去研究一下这些API是非常有意思的。

我们知道Android API依赖的Java SE API也非常庞大，根据统计，Java SE 8有217个package，4240个方法，而java SE 7有209个package，4024个方法。

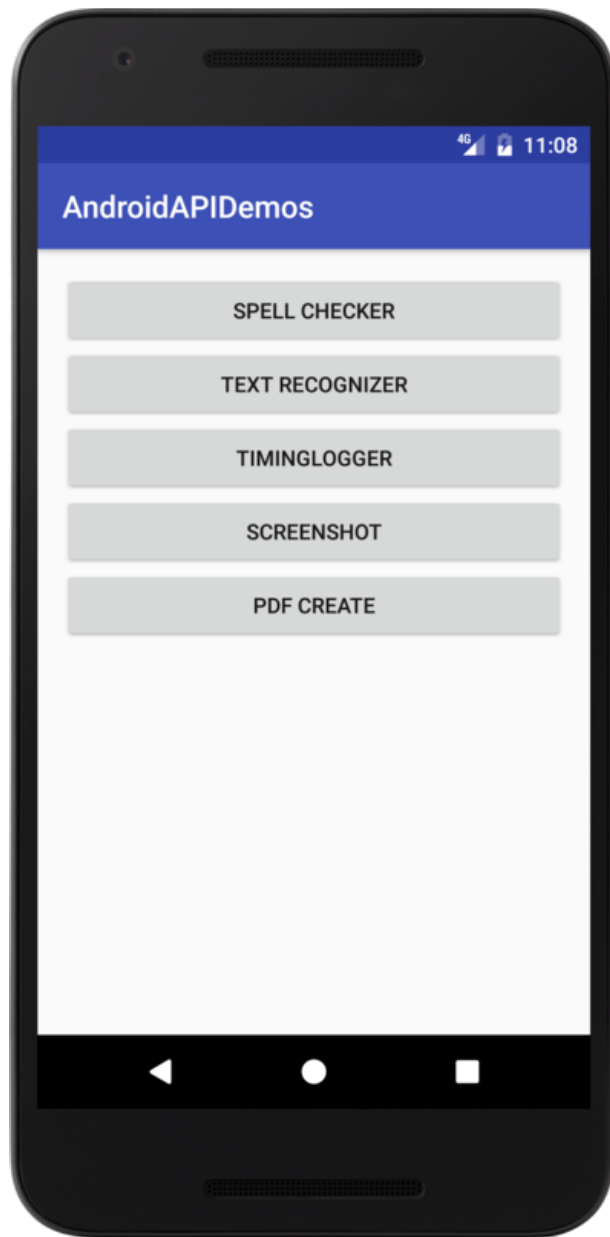
Java Development Kits	Codename	Release	Packages	Classes
Java SE 8 with JDK 1.8.0	---	2014	217	4,240
Java SE 7 with JDK 1.7.0	Dolphin	2011	209	4,024
Java SE 6 with JDK 1.6.0	Mustang	2006	203	3,793
Java 2 SE 5.0 with JDK 1.5.0	Tiger	2004	166	3,279
Java 2 SE with SDK 1.4.0	Merlin	2002	135	2,991
Java 2 SE with SDK 1.3	Kestrel	2000	76	1,842
Java 2 with SDK 1.2	Playground	1998	59	1,520
Development Kit 1.1	---	1997	23	504
Development Kit 1.0	Oak	1996	8	212

source: Java 8 Pocket Guide book by Robert Liguori, Patricia Liguori

在这个系列文章中，我们将从不同角度展示一些鲜为人知的Android API，并使用这些API写了demo，开源地址：<https://github.com/peekler/GDG> (<https://github.com/peekler/GDG>)

demo App中给出的每个API的使用都是在不同的Activity中，从App首页可以进入到不同的API demoActivity。





拼写检查

Android从level 14开始有一个检查拼写的API，可以通过 `TextServicesManager` (<https://developer.android.com/reference/android/view/textservice/TextServicesManager.html>)使用，从level16开始已经可以甚至可以检查一个完整的句子了。

使用方法非常简单，通过 `TextServicesManager` (<https://developer.android.com/reference/android/view/textservice/TextServicesManager.html>)可以创建`SpellCheckerSession` (<https://developer.android.com/reference/android/view/textservice/SpellCheckerSession.html>)：



```
TextServicesManager tsm = (TextServicesManager) getSystemService(Context.TEXT_SERVICE);  
SpellCheckerSession spellCheckerSession = tsm.newSpellCheckerSession(null, null, this);
```

可以通过实现SpellCheckerSessionListener接口得到检查结果：

```
onGetSuggestions(SentenceSuggestionsInfo[] sentenceSuggestionsInfos)  
onGetSentenceSuggestions(SentenceSuggestionsInfo[] sentenceSuggestionsInfos));
```

SentenceSuggestionsInfo (<https://developer.android.com/reference/android/view/textservice/SentenceSuggestionsInfo.html>)数据中保存了正确的文字、偏移量以及所有相关的信息。





demo地址SpellCheckerActivity (<https://github.com/peekler/GDG/blob/master/AndroidAPIDemos/app/src/main/java/hu/autsoft/androidapidemos/SpellCheckerActivity.java>)

文字识别

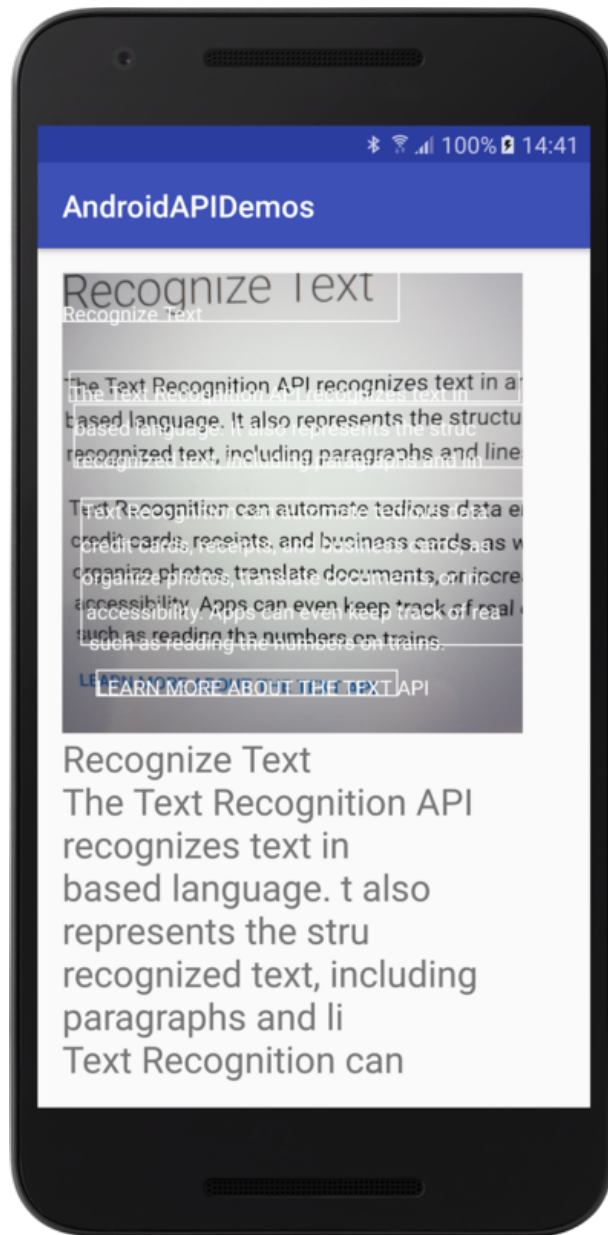
这是Google Play Services Vision API中提供的功能，可以通过gradle dependency非常的引入到project中，需要注意的是不要引入整个Play Services，因为Play Services非常大，而我们需要的只是其中的一小部分，<https://developers.google.com/android/guides/setup> (<https://developers.google.com/android/guides/setup>)中可以找到相关的帮助。



Vision API (<https://developers.google.com/vision/>)中包含的服务有：

- 人脸识别
- 条形码扫描
- 文字识别

使用 Text Recognizer API (<https://developers.google.com/vision/text-overview>)非常简单：



首先，在build.gradle中引入依赖：

```
compile 'com.google.android.gms:play-services-vision:10.0.1'
```



然后创建TextRecognizer对象：

```
TextRecognizer textRecognizer = new TextRecognizer.Builder(context).build();
```

之后实现 Detector.Processor (<https://developers.google.com/android/reference/com/google/android/gms/vision/Detector.Processor>) 接口接口监听结果，得到的结果是TextBlock (<https://developers.google.com/android/reference/com/google/android/gms/vision/text/TextBlock>) 数组。

```
public class OcrDetectorProcessor implements Detector.Processor<TextBlock> {  
  
    @Override  
    public void receiveDetections(Detector.Detections<TextBlock> detections) {  
        ...  
        SparseArray<TextBlock> items = detections.getDetectedItems();  
        ...  
    }  
  
    @Override  
    public void release() {  
    }  
}
```

合理地使用 TextRecognizer (<https://developers.google.com/android/reference/com/google/android/gms/vision/text/TextRecognizer>)，一般要自定义包含SurfaceView (<https://developer.android.com/reference/android/view/SurfaceView.html>)的View用于在屏幕显示结果。demo地址 OCRActivity (<https://github.com/peekler/GDG/blob/master/AndroidAPIDemos/app/src/main/java/hu/autsoft/androidapidemos/OCRActivity.java>)，ocr (<https://github.com/peekler/GDG/tree/master/AndroidAPIDemos/app/src/main/java/hu/autsoft/androidapidemos/ocr>) 中有一些帮助类。

TimingLogger

TimingLogger (<https://developer.android.com/reference/android/util/TimingLogger.html>)

可以很容易地计算两个log信息之间的时间差，如下所示：

```
D/TAG_MYJOB: MyJob: begin  
D/TAG_MYJOB: MyJob: 2002 ms, Phase 1 ready  
D/TAG_MYJOB: MyJob: 2002 ms, Phase 2 ready  
D/TAG_MYJOB: MyJob: 2001 ms, Phase 3 ready  
D/TAG_MYJOB: MyJob: end, 6005 ms
```



使用TimingLogger (<https://developer.android.com/reference/android/util/TimingLogger.h>)

tml) :

```
TimingLogger timings = new TimingLogger("TAG_MYJOB", "MyJob");
```

然后通过`addSplit(...)` 方法创建一个 log entry :

```
timings.addSplit("Phase 1 ready");
```

当使用`dumpToLog()`后 , log信息就会打印出来 :

```
timings.dumpToLog();
```

注意要使用TimingLogger , 要设置adb命令是Tag可用 :

```
setprop log.tag.TAG_MYJOB VERBOSE
```

demo地址 : TimingLoggerActivity (<https://github.com/peekler/GDG/blob/master/AndroidAPIDemos/app/src/main/java/hu/autsoft/androidapidemos/TimingLoggerActivity.java>).

截屏

在某些情况下 , 截屏非常有用。也有一些第三方库如 Falcon (<https://github.com/jraska/Falcon>)实现这个功能 , 从level 21开始 MediaProjection (<https://developer.android.com/reference/android/media/projection/MediaProjection.html>)可以实时获取屏幕内容和系统声音信息流。

qi , 有时使用标准的Android API通过 `getWindow()` 非常简单地把屏幕内容保存为 Bitmap :

```
View viewRoot = getWindow().getDecorView().getRootView();
viewRoot.setDrawingCacheEnabled(true);
Bitmap screenShotAsBitmap = Bitmap.createBitmap(viewRoot.getDrawingCache());
viewRoot.setDrawingCacheEnabled(false);
// use screenShotAsBitmap as you need
```





demo地址：ScreenCaptureActivity (<https://github.com/peekler/GDG/blob/master/AndroidAPIDemos/app/src/main/java/hu/autsoft/androidapidemos/ScreenCaptureActivity.java>)

PDF创建

从level 19开始Android支持本地内容生成PDF文件。

首先创建一个PageInfo new PdfDocument.PageInfo.Builder(w,h,pageNum).create()
; ,然后使用PDFDocument中的 startPage([pageInfo]) 就可以创建一个PDF文件了。



以下的代码创建了一个demo.pdf文件：

```
public void createPdfFromCurrentScreen() {
    new Thread() {
        public void run() {
            // Get the directory for the app's private pictures directory.
            final File file = new File(
                Environment.getExternalStorageDirectory(), "demo.pdf");

            if (file.exists ()) {
                file.delete ();
            }

            FileOutputStream out = null;
            try {
                out = new FileOutputStream(file);

                PdfDocument document = new PdfDocument();
                Point windowSize = new Point();
                getWindowManager().getDefaultDisplay().getSize(windowSize);
                PdfDocument.PageInfo pageInfo =
                    new PdfDocument.PageInfo.Builder(
                        windowSize.x, windowSize.y, 1).create();
                PdfDocument.Page page = document.startPage(pageInfo);
                View content = getWindow().getDecorView();
                content.draw(page.getCanvas());
                document.finishPage(page);
                document.writeTo(out);
                document.close();
                out.flush();

                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(PDFCreateActivity.this, "File created: "+file
                    }
                });
            } catch (Exception e) {
                Log.d("TAG_PDF", "File was not created: "+e.getMessage());
            } finally {
                try {
                    out.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }.start();
}
```

感谢阅读。

此篇是译文，原文地址 [Discovering the Android API - Part 1 \(https://blog.autsoft.hu/discovering-the-android-api-part-1/\)](https://blog.autsoft.hu/discovering-the-android-api-part-1/)



Other

欢迎关注公众号wutongke，每天推送移动开发前沿技术文章：



wutongke


推荐阅读：

重要-作为Android开发者必须了解的Gradle知识 (<http://www.jianshu.com/p/c31513f5f550>)

编写高效的Android代码（译） (<http://www.jianshu.com/p/d8f2eab43e4a>)

Android中使用gradient的一条建议 (<http://www.jianshu.com/p/508d1cf8fb61>)

寻找卓越的（Android）软件工程师 (<http://www.jianshu.com/p/3615c18539bc>)

 Android (/nb/6315168)

[举报文章](#) © 著作权归作者所有



wutongke (/u/0e0821e94979) ♂

写了 64259 字，被 1638 人关注，获得了 2833 个喜欢
(/u/0e0821e94979)

+ 关注

Hi西木


如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

赞赏支持

 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 3




更多分享



下载简书 App ▶

随时随地发现和创作内容



nshu.io

9b.jpg)

(/apps/download?utm_source=nbc)

被以下专题收入，发现更多相似内容



编程基础 (/c/ac606004dc94?utm_source=desktop&utm_medium=notes-included-collection)



大神带我打代码 (/c/3572965061ef?utm_source=desktop&utm_medium=notes-included-collection)



资源贴 (/c/c5f567d84127?utm_source=desktop&utm_medium=notes-included-collection)



今日看点 (/c/3sT4qY?utm_source=desktop&utm_medium=notes-included-collection)



Android开发 (/c/0dc880a2c73c?utm_source=desktop&utm_medium=notes-included-collection)



程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)



Android... (/c/c0e20adfb3bb?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ∨

