

توسعه برنامه های

موبایل

جلسه یازدهم مجازی

بخش دوم

سحر صادقی

## آموزش ایجاد بارکد خوان در برنامه نویسی اندروید

برنامه های زیادی همانند Api گوگل برای استفاده و ایجاد اسکنر بارکدخوان در اپلیکشن های اندروید وجود دارد. اما هر کدام از این ابزارها در هنگام بارگذاری و اجرا مشکلات خاص خود را دارند. بنابراین استفاده از یک کتابخانه خوب می تواند تمام این مشکلات را حل نماید.

در این آموزش با استفاده از کتابخانه ZXing به شما آموزش خواهیم داد تا بتوانید به راحتی یک اپلیکشن بارکدخوان ایجاد و یا از این کتابخانه در پروژه های خود استفاده نمایید.

توجه: لطفا قبل از استفاده به مجوز های استفاده از این کتابخانه دقت فرمایید

<https://github.com/tarun0/ZXing-Standalone-library>.

ابتدا این کتابخانه را با استفاده از کد زیر در پروژه ی خود فراخوانی کنید:

```
compile 'com.tarun0.zxing-standalone:zxing-standalone:1.0.0'
```

حال هر کجا که نیاز به استفاده از باکدخوان دارید کدهای زیر را فراخوانی کنید:

```
Intent intent = new Intent(getApplicationContext(), CaptureActivity.class);
intent.setAction("com.google.zxing.client.android.SCAN");
intent.putExtra("SAVE_HISTORY", false);
startActivityForResult(intent, RC_BARCODE_CAPTURE);
```

نتیجه اسکن بارکد به اکتیویتی فراخوانی شده برگشت داده می شود و شما به راحتی می توانید نتیجه را با استفاده از تابع زیر دریافت کنید:

```
@Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == RC_BARCODE_CAPTURE) {
            if (resultCode == RESULT_OK) {
                if (data.getStringExtra("SCAN_RESULT_FORMAT").equals("UPC_A") ||
                    data.getStringExtra("SCAN_RESULT_FORMAT").equals("UPC_E")) {
                    String contents = data.getStringExtra("SCAN_RESULT");
```

```

        mBinding.productUpc.setText(contents);
    } else {
        Toast.makeText(this, "The barcode scanned should
be of type UPC", Toast.LENGTH_LONG).show();
    }
} else if (resultCode == RESULT_CANCELED) {
    Log.d("TAG", "RESULT_CANCELED");
    Toast.makeText(this, "Scanning cancelled", Toast.LEN
GTH_LONG).show();
}
}
}
}

```

در تابع بالا می‌توانید نتیجه را دریافت و به صورتی که دوست دارید از آن استفاده کنید.

## آموزش استفاده از بارکد و QR code scanner در اندروید با کمک Google Mobile Vision در

### Android

امروزه بارکدها و QR code ها کاربرد گسترده‌ای در بسیاری از نرم افزارهای موبایل دارند. شما می‌توانید در یک QR code اطلاعاتی مانند: متن، sms، ایمیل، لینک url، عکس، صدا و فرمت‌های دیگر را ذخیره کنید. می‌توان در اندروید با استفاده از کتابخانه Google Mobile Vision اطلاعات ذخیره شده در بارکدها را استخراج کرد. با وجود اینکه کتابخانه‌های زیاد دیگری برای انجام این کار وجود دارند، اما این کتابخانه از جمله بهترین گزینه‌ها است، به این دلیل که نه تنها می‌توان با کمک آن به خواندن بارکدها پرداخت بلکه از ویژگی‌های دیگری مانند شناسایی چهره نیز بهره برد. در این نوشته قصد داریم با ایجاد کردن یک نرم افزار ساده اسکنر بلیت سینما، چگونگی استفاده از کتابخانه google vision را بیاموزیم.

### آموزش Android در Google Mobile Vision API

این API به شما در پیدا کردن اشیاء در تصاویر یا ویدئوها کمک می‌کند Google Mobile Vision API. از قابلیت‌هایی مانند: شناسایی چهره، شناسایی متن و شناسایی بارکد بهره می‌برد. تمام این قابلیت‌ها را می‌توان به صورت جداگانه یا در کنار هم استفاده کرد. هدف این نوشتار توضیح شناسایی بارکد با استفاده از یک مثال موردی real time می‌باشد. تا به حال در مکان‌هایی مانند سوپرمارکت‌ها، تئاتر و هتل نرم افزارهای اسکنر بارکد زیادی دیده‌ایم که می‌توانند اطلاعات مورد نظر کاربر را ارائه کنند. در این مقاله قصد داریم نرم افزار

اسکنر بلیت سینمایی را بسازیم به گونه‌ای که این نرم افزار یک بارکد یا QR Code را اسکن کند و در نهایت برای خریداری بلیت سینما، اطلاعات مربوط به آن بلیت را نمایش دهد. کتابخانه google vision بخشی از خدمات مربوط به google play service است و می‌توانید آن را در build.gradle پروژه خودتان اضافه کنید.

[?](#)

```
1 compile 'com.google.android.gms:play-services-vision:11.0.2'
```

## آموزش Android Barcode Scanner Library

گوگل، آموزش ساده‌ای را با کمک یک تصویر ساده bitmap برای آشناسدن با barcode scanning library فراهم کرده است. اما زمانی که بخواهیم به صورت realtime بازخورد دوربینی را اسکن کنیم، کار یک مقدار سخت می‌شود، زیرا در این صورت به شناسایی بارکد توسط دوربین نیاز خواهیم داشت. من با منشعب کردن (forking) نمونه کار google vision، کتابخانه Barcode Scanner ساده‌ای را توسعه داده‌ام. برخی از باگ‌های نرم افزاری این کتابخانه برطرف شده و برای مواقعی که بارکدی اسکن می‌شود، توابع callback ای در نظر گرفته شده‌اند. همچنین از جمله دیگر قابلیت‌های این کتابخانه می‌توان به وجود یک overlay scanning line indicator اشاره کرد. از این اندیکاتور می‌توانید در نرم افزارهای خودتان استفاده کنید.

برای اینکه روش استفاده از کتابخانه‌ی مربوط به بارکد و QR Code را متوجه شوید، مراحل زیر را دنبال کنید.

۱. androidhive barcode reader و google vision library را به نرم افزار خودتان اضافه کنید.

این مقاله با استفاده از Android Studio 3.0 Canary 9 نوشته شده و implementation کنار گذاشته شده جای خود را به compile داده است.

[?](#)

```
1 build.gradle
2 dependencies {
3     // barcode reader library
4     implementation 'info.androidhive:barcode-reader:1.1.5'
5 }
```

```

6      // google vision library
7      implementation 'com.google.android.gms:play-services-
vision:11.0.2'
8  }

```

۲. جزء مربوط به barcode camera را به جزء یا activity خود اضافه کنید.

[?](#)

```

1  < fragment
2      android:id="@+id/barcode_scanner"
3      android:name="info.androidhive.barcode.BarcodeReader"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      app:auto_focus="true"
7      app:use_flash="false" / >

```

۳. activity خودتان را از BarcodeReader.BarcodeReaderListener پیاده‌سازی کرده و متدهای لازم را override کنید.

4.

```

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;
import android.util.SparseArray; import
com.google.android.gms.vision.barcode.Barcode; import java.util.List; import
info.androidhive.barcode.BarcodeReader; public class MainActivity extends
AppCompatActivity implements BarcodeReader.BarcodeReaderListener {
@Override protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_scan); }
@Override public void onScanned(Barcode barcode) { // single barcode scanned
} @Override public void onScannedMultiple(List< Barcode > list) { // multiple
barcodes scanned } @Override public void onBitmapScanned(SparseArray<
Barcode > sparseArray) { // barcode scanned from bitmap image } @Override
public void onScanError(String s) { // scan error } @Override public void
onCameraPermissionDenied() { // camera permission denied } }

```

۵. پروژه خود را اجرا کرده و سعی کنید بارکد یا یک QR Code را اسکن کنید. نتیجه اسکن شده به شکل منتهای onscanned() یا onScannedMultiple() برگشت خواهد خورد.

آموزش اضافه کردن قابلیت اسکن کردن با استفاده از خط شاخص در overlay دوربین (Overlay Indicator Line در Android)

عموماً تمام نرم افزارهای اسکن بارکد برای اینکه نشان دهند که فرایند اسکن کردن تا چه حد پیشرفت داشته، از یک خط شاخص یا indicator line بهره می برند. من برای اینکه بتوانم به این قابلیت دست پیدا کنم، کلاسی را در همین کتابخانه ایجاد کرده ام به گونه ای که این کلاس قابلیت استفاده مجدد را دارد و می توان آن را به صفحه دوربین اضافه کرد. برای اضافه کردن یک خط اسکن انیمیشنی، info.androidhive.barcode.ScannerOverlay را به همان فعالیتی اضافه کنید که فرگمنت یا جزء camera را همپوشانی کرده است.

[?](#)

```
< info.androidhive.barcode.ScannerOverlay android:layout_width="match_parent"
app:line_color="#7323DC" app:line_speed="6" app:line_width="4" app:square="1"/>
```

این کتابخانه قابلیت های دیگری مانند فلش خودکار دوربین و صدای beep را نیز دارا است. برای اطلاعات دقیق تر در رابطه با این کتاب خانه به صفحه Github آن مراجعه کنید.

آموزش ایجاد پروژه جدید – ساخت نرم افزار اسکن بلیت در Android

با توجه به اینکه Barcode Scanner Library در حال حاضر موجود است، می خواهیم بدانیم که چگونه می توان با در نظر گرفتن یک مثال واقعی، از این کتاب خانه استفاده کرد. این برنامه ای که قرار است بسازیم، نه تنها اسکن بارکد را مشخص می کند، بلکه مباحثی مانند ساختن رابط کاربری پیچیده، فراخوانی REST api برای دریافت فرمت json فیلم و نوشتن کلاس های شخصی view را پوشش می دهد. در مجموع این نرم افزار شامل سه صفحه می شود:

**صفحه splash**

صفحه اسکن بارکد و صفحه ای است که اطلاعات مربوط به بلیت سینما را نمایش می دهد. در زیر می توانید اسکرین شات های مربوط به نرم افزار را مشاهده کنید.



## آموزش REST API در Android

برای ساخت این نرم افزار ، به یک REST api نیاز خواهیم داشت تا بتوانیم توسط بارکد فیلم، در دیتابیس فیلمها به جستجو پردازیم. برای جستجوی فیلمها بر اساس بارکد، من برای نمونه یک rest api را نوشتم. این api مقدار پیش تعریف شده بارکد را به عنوان پارامتر جستجو ( query param ) می گیرد و در db به دنبال آن می گردد. در زیر می توانید برای آزمایش نمونه هایی از بارکدها را مشاهده کنید.

1. dn\_barcode.jpg
2. spiderman\_barcode.jpg
3. wonderwoman\_barcode.jpg
4. dunkirk\_barcode.jpg

### جستجو کردن فیلم:

با استفاده از مقدار read بارکد، درخواست get را ایجاد کنید.

<https://api.androidhive.info/barcodes/search.php?code=dunkirk>

### نتیجه فیلم:

در واکنش به درخواست بالا، json مطابق فیلم ارائه می شود.

[?](#)

```
1  {
2      "name": "Dunkirk",
3      "poster": "https://api.androidhive.info/barcodes/dunkirk.jpg",
4      "duration": "1hr 46min",
5      "rating": 4.6,
6      "released": true,
7      "genre": "Action",
8      "price": "□200",
9      "director": "Christopher Nolan"
10 }
```

حالا به تمام اطلاعاتی که نیاز داریم دسترسی داریم. بیاید کارمان را با ایجاد یک پروژه جدید در اندروید استودیو شروع کنیم.

**نکته:**

این پروژه با استفاده از 9 Android Studio 3.0 Canary نوشته شده است.

۱. برای ایجاد یک پروژه جدید در اندروید استودیو به file رفته و بر روی new project کلیک کنید و بعد از آن جزئیات مربوط به پروژه را پر کنید. من اسم پروژه خودم را MovieTickets و اسم پکیج را info.androidhive.movietickets گذاشته‌ام.

۲. build.gradle 2. نرم افزار را باز کنید و وابستگی‌های مربوط به barcode و google vision را در آن اضافه کنید. برای اینکه بتوانیم http ها را فراخوانی، json را تجزیه (Parse) کنیم و تصاویر را نمایش دهیم؛ باید کتابخانه‌های Glide، Volley و Gson را اضافه کنیم.

[?](#)

```
1 app/build.gradle
```

[?](#)



```

dependencies {
1     implementation 'com.google.android.gms:play-services-
2 vision:11.0.2'
3
4     // vision barcode scanner
5     implementation 'info.androidhive:barcode-reader:1.1.2'
6
7     // glide image library
8     implementation 'com.github.bumptech.glide:glide:4.0.0-
RC1'
9     annotationProcessor
10    'com.github.bumptech.glide:compiler:4.0.0-RC1'
11
12    implementation 'com.android.volley:volley:1.0.0'
13    implementation 'com.google.code.gson:gson:2.6.2'
}

```

۳. 3 منابع color,dimen و string زیر را در دایرکتوری res به فایل‌های مربوط به خودشان اضافه کنید.

[?](#)

```
1 strings.xml
```

[?](#)

```

1 < resources >
2     < string name="app_name">Movie Tickets< /string >
3     < string name="title_activity_ticket">Book Ticket< /string >
4     < string name="lbl_duration">DURATION< /string >
5     < string name="lbl_genre">GENRE< /string >
6     < string name="lbl_rating">RATING< /string >
7     < string name="lbl_price">PRICE< /string >
8     < string name="btn_buy_now">BUY NOW< /string >

```

```
8      < string name="btn_coming_soon">COMING SOON< /string >
9      < string name="msg_no_ticket_found">No ticket found. Try scanning
http://api.androidhive.info/QR Codes/< /string >
10    < /resources >
11
```

[?](#)

```
1  dims.xml
```

[?](#)

```
1  < ?xml version="1.0" encoding="utf-8"? >
2  < resources >
3      < dimen name="dimen_20">20dp< /dimen >
4      < dimen name="dimen_10">10dp< /dimen >
5      < dimen name="activity_margin">16dp< /dimen >
6      < dimen name="lbl_directory">14dp< /dimen >
7      < dimen name="lbl_movie_name">28dp< /dimen >
8      < dimen name="img_poster_height">220dp< /dimen >
9      < dimen name="dimen_40">40dp< /dimen >
10 < /resources >
```

[?](#)

```
1  colors.xml
```

[?](#)

```
1  < ?xml version="1.0" encoding="utf-8"? >
2  < resources >
3      < color name="colorPrimary">#6d0094< /color >
```

```

4      < color name="colorPrimaryDark">#6d0094< /color >
5      < color name="colorAccent">#ff2068< /color >
6      < color name="colorAccentSecondary">#ad1a7f< /color >
7      < color name="viewBg">#f8f8f8< /color >
8      < color name="btn_disabled">#999< /color >
9      < color name="lbl_value">#222222< /color >
10     < /resources >

```

۴. کلاسی با نام MyApplication.java را ایجاد کرده و کد زیر را در آن اضافه کنید. در اینجا ما یک volley تک شیء ( volley singleton instance ) را ایجاد می‌کنیم. volley برای فراخوانی http مناسب نیست، بلکه در این آموزش برای ایجاد نمونه از یکپارچه سازی لحاظ شده است. برای نرم افزارهای تولید شده خود Retrofit را لحاظ کنید.

[?](#)

```

1 MyApplication.java

```

[?](#)

```

1  import android.app.Application;
2  import android.text.TextUtils;
3
4  import com.android.volley.Request;
5  import com.android.volley.RequestQueue;
6  import com.android.volley.toolbox.Volley;
7
8  /**
9   * Created by ravi on 31/07/17.
10   */
11
12  public class MyApplication extends Application {
13

```

```
14     public static final String TAG = MyApplication.class
15         .getSimpleName();
16
17     private RequestQueue mRequestQueue;
18
19     private static MyApplication mInstance;
20
21     @Override
22     public void onCreate() {
23         super.onCreate();
24         mInstance = this;
25     }
26
27     public static synchronized MyApplication getInstance() {
28         return mInstance;
29     }
30
31     public RequestQueue getRequestQueue() {
32         if (mRequestQueue == null) {
33             mRequestQueue =
34             Volley.newRequestQueue(getApplicationContext());
35
36             return mRequestQueue;
37         }
38
39         public< T > void addToRequestQueue(Request< T > req,
40             String tag) {
41             // set the default tag if tag is empty
42             req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
43             getRequestQueue().add(req);
```

```

43     }
44
45     public < T > void addToRequestQueue(Request< T > req) {
46         req.setTag(TAG);
47         getRequestQueue().add(req);
48     }
49
50     public void cancelPendingRequests(Object tag) {
51         if (mRequestQueue != null) {
52             mRequestQueue.cancelAll(tag);
53         }
54     }
55 }

```

۵. AndroidManifest.xml را باز کنید و کلاس MyApplication را به تگ < application > اضافه کنید permission مربوط به اینترنت را هم اضافه کنید چون به فراخوانی http نیاز خواهیم داشت.

[?](#)

1 AndroidManifest.xml

[?](#)

```

1 < ?xml version="1.0" encoding="utf-8"? >
2 < manifest
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     package="info.androidhive.movietickets" >
5
6     < uses-permission
7         android:name="android.permission.INTERNET" / >
8
9     < application
10         android:name=".MyApplication"

```

```

10         android:allowBackup="true"
11         android:icon="@mipmap/ic_launcher"
12         android:label="@string/app_name"
13         android:roundIcon="@mipmap/ic_launcher_round"
14         android:supportsRtl="true"
15         android:theme="@style/AppTheme" >
16         < activity
17             android:name=".MainActivity"
18             android:screenOrientation="portrait"
19             android:theme="@style/AppTheme.NoActionBar" >
20             < intent-filter >
21                 < action
22                     android:name="android.intent.action.MAIN" / >
23                 < category
24                     android:name="android.intent.category.LAUNCHER" / >
25             < /intent-filter >
26         < /activity >
27         < activity
28             android:name=".ScanActivity"
29             android:screenOrientation="portrait" / >
30         < activity
31             android:name=".TicketActivity"
32             android:label="@string/title_activity_ticket"
33             android:screenOrientation="portrait"
34             android:theme="@style/AppTheme.NoActionBar" ><
35     /activity >
36     < /application >
37 < /manifest >

```

۶. 6. در پوشه drawable ⇒ res ، یک drawable xml به نام bg\_gradient.xml را اضافه کنید، این

drawable، پس زمینه متغیری به view می دهد.

[?](#)

1 bg\_gradient.xml

[?](#)

```
1 < ?xml version="1.0" encoding="utf-8"? >
2 < shape
  xmlns:android="http://schemas.android.com/apk/res/android"
3   android:shape="rectangle" >
4     < gradient
5       android:angle="135"
6       android:centerColor="@color/colorAccentSecondary"
7       android:endColor="@color/colorPrimary"
8       android:startColor="@color/colorAccent"
9       android:type="linear" / >
10
11     < corners android:radius="0dp" / >
12 < /shape >
```