



# آموزش برنامه نویسی اندروید در محیط اندروید استودیو

متریال دیزاین

بخش دهم: کار با Bottom Sheet

مدرس : سید مهدی مطهری

[www.android-studio.ir](http://www.android-studio.ir)

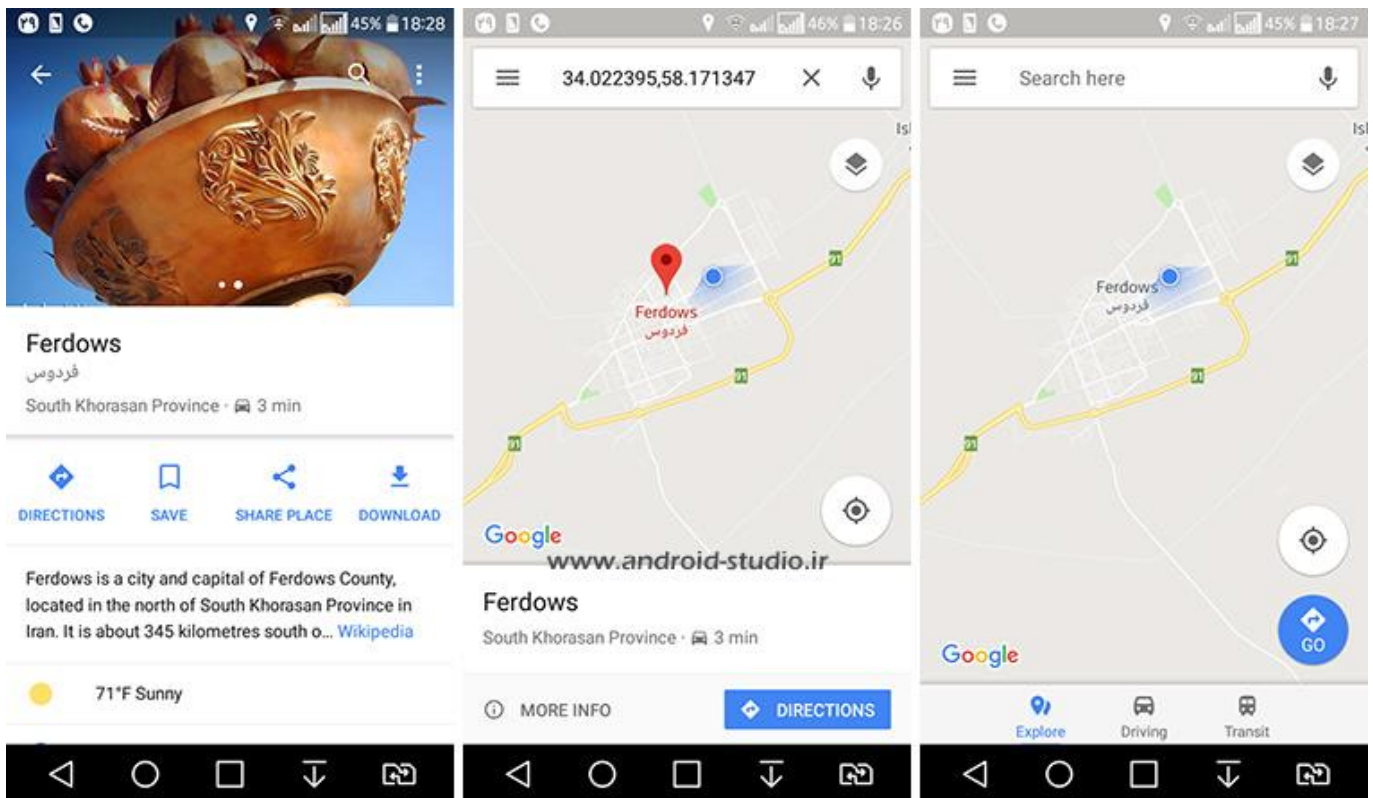


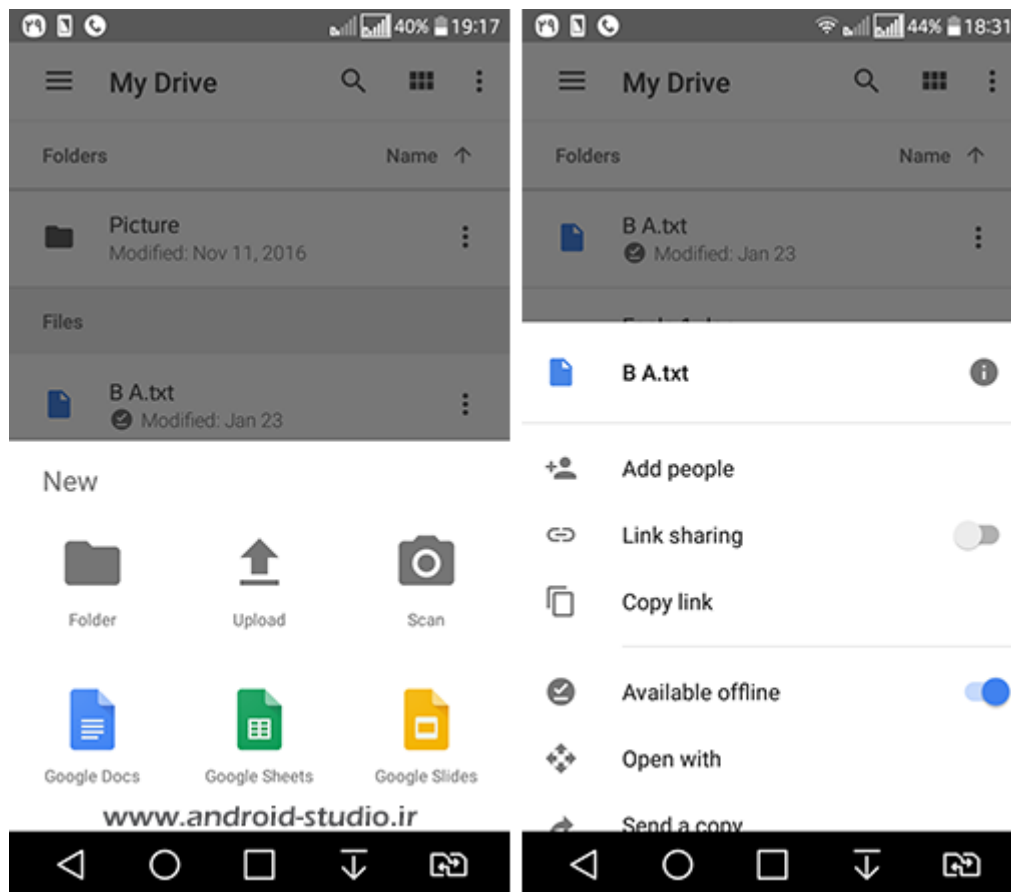
## به نام خدا

### معرفی Bottom Sheet:

Bottom Sheet یکی دیگر از کامپوننت‌های **متریال دیزاین** در اندروید است که برای نمایش اطلاعات، منوها و یا هر چیز دیگری بکار می‌رود. به تعریف ساده، یک صفحه یا نواری است که از پایین صفحه نمایش به سمت بالا باز می‌شود. به طور کلی Bottom Sheet ها را می‌توان به دو دسته Persistent Bottom Sheet و Modal Bottom Sheet تقسیم کرد.

در ادامه دو مثال از کاربرد این کامپوننت را نشان می‌دهم که تصویر اول مربوط به Google Maps و تصویر دوم Google Drive است:





در Google Maps با Pin کردن یک نقطه روی نقشه، یک Bottom Sheet از پایین صفحه باز می‌شود که حاوی اطلاعات مختصری از مکان انتخاب شده است. سپس با لمس دکمه MORE INFO یا کشیدن نوار به سمت بالا، اطلاعات بیشتری به کاربر ارائه می‌شود که در گوگل مپ، در این حالت تمام صفحه را دربر می‌گیرد.

در گوگل درایو هم از این کامپوننت برای نمایش گزینه‌های مرتبط با فایل و همچنین سایر موارد مانند ساخت فولدر جدید یا آپلود فایل استفاده شده است.

من این مبحث را در قالب چند پروژه تهیه می‌کنم تا درک جزئیات و تفاوت‌ها ساده تر شود.

ابتدا یک **پروژه جدید** با نام SimpleBottomSheet و یک Empty Activity ایجاد می‌کنم. برای پیاده سازی Bottom Sheet لازم است کتابخانه Support Design را به پروژه اضافه کنیم:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support:design:27.1.1'
}
```



در این پروژه همانطور که از نام آن پیداست قصد داریم یک Bottom Sheet را در ساده ترین حالت آن معرفی کنیم. بنابراین در این مثال فقط با layout اکتیویتی سروکار داریم و از **java** استفاده نخواهیم کرد. در قدم اول جهت نمایش Bottom Sheet در یک اکتیویتی، باید Root Layout را از نوع CoordinatorLayout انتخاب کنیم:

:activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    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"
    tools:context=".MainActivity">

</android.support.design.widget.CoordinatorLayout>
```

همانطور که در کد بالا ملاحظه می کنید، CoordinatorLayout هم از زیرمجموعه های کتابخانه Design است. از CoordinatorLayout برای کنترل حرکات در متریال استفاده می شود. مانند اجرای animation در حرکت عناصر و المان های رابط کاربری. یعنی در Bottom Sheet استفاده از این layout باعث می شود تا باز و بسته شدن آن با یک حالت انیمیشن همراه باشد که نسبت به یک باز و بسته شدن عادی و ناگهانی، حس و تجربه بهتری را به کاربر منتقل می کند. CoordinatorLayout کاربردهای گسترده ای دارد که در مباحث آتی و در جای مناسب اشاره خواهد شد. به عنوان مثال در برخی از اپ ها با اسکرول به پایین، Toolbar حذف و با اسکرول به بالا، مجدد ظاهر می شود که پیاده سازی این عمل با CoordinatorLayout انجام می پذیرد.



ابتدا یک Bottom Sheet ساده تعریف کرده و در ادامه توضیحات لازم را قید می‌کنم:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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"
xmlns:app="http://schemas.android.com/apk/res-auto"
tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:orientation="vertical"
        android:background="#009688"
        android:padding="8dp"
        app:layout_behavior="android.support.design.widget.BottomSheetBehavior">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#fff"
            android:textSize="20sp"
            android:textStyle="bold"
            android:text="Bottom Sheet Title" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#fff"
            android:textSize="16sp"
            android:text="Lorem ipsum dolor sit amet, te sed vide delicata,
per ex salutandi intellegat temporibus, ei insolens molestiae vis.
Eum ei possim aperiam, fuisset suscipit vim ut. Voluptua repudiare
gubergren id eum, nullam labores an nam. Sed et tota quaerendum,
enim elit democritum quo et.
Lorem ipsum dolor sit amet, te sed vide delicata,
per ex salutandi intellegat temporibus, ei insolens molestiae vis.
Eum ei possim aperiam, fuisset suscipit vim ut. Voluptua repudiare
gubergren id eum, nullam labores an nam. Sed et tota quaerendum,
enim elit democritum quo et." />

    </LinearLayout>
</android.support.design.widget.CoordinatorLayout>
```

یک LinearLayout به layout اضافه کردم که حاوی دو TextView می‌باشد. در انتخاب نوع این Layout محدودیتی نیست و از سایر موارد مانند RelativeLayout، ConstraintLayout و ... هم می‌توان استفاده کرد. کاری که باید انجام دهیم، تعیین این LinearLayout به عنوان یک Bottom Sheet است که به اینصورت انجام شده:

```
app:layout_behavior="android.support.design.widget.BottomSheetBehavior"
```



با افزودن ویژگی فوق به یک layout، اندروید آنرا به عنوان یک Bottom Sheet در نظر می‌گیرد. مقدار این ویژگی را به صورت دیگری هم می‌توانیم تعریف کنیم:

```
app:layout_behavior="@string/bottom_sheet_behavior"
```

کلید Ctrl را نگه داشته، روی مقدار فوق کلیک کنید. به فایل values.xml مربوط به کتابخانه design منتقل می‌شوید که در cache سیستم شما ذخیره شده است. به عنوان فوق برنامه، مسیر محل قرارگیری این فایل که در نوار آدرس اندروید استودیو نمایش داده می‌شود را در درایو مربوطه باز کنید. فولدرهای قبل آن را هم بررسی کنید. تمامی کتابخانه‌هایی که قبلاً یکبار استفاده کردید در اینجا ذخیره شده و برای پروژه‌های بعدی استفاده می‌شود.

پروژه را اجرا می‌کنم:



یک نوار با رنگ پس زمینه و ارتفاع 200dp که قبلاً تعیین کردم در پایین صفحه ظاهر شده که محتوای آن شامل دو TextView می‌باشد. در حال حاضر این یک نوار ثابت است که با کشیدن به سمت پایین

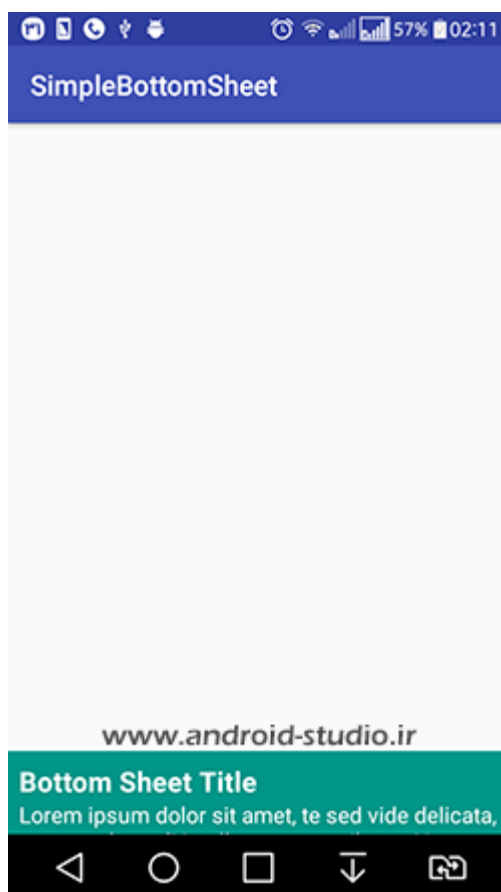




هیچ واکنشی صورت نمی‌پذیرد. خاصیت زیر را با یک مقدار دلخواه به Bottom Sheet (یعنی LinearLayout) اضافه می‌کنم:

```
app:behavior_peekHeight="60dp"
```

مجدد پروژه را اجرا می‌کنم:



(تصویر 1.gif را مشاهده کنید)

**تذکر:** مواردی مانند تصویر بالا جهت مشاهده بهتر خروجی، در قالب تصویر متحرک تهیه شده که در فولدر Gif قرار دارد.

با اضافه شدن خاصیت فوق، نوار ابتدا ارتفاعی برابر مقدار تعریف شده یعنی 60dp دارد که با کشیدن نوار به سمت بالا، تا حداکثر ارتفاع خود یعنی 200dp باز شده و مجدداً با کشیدن آن به سمت پایین، به ارتفاع اولیه یعنی 60dp برمی‌گردد.



واضح است اگر مقدار `behavior_peekHeight` را برابر با 0 قرار بدهیم، در حالت پیش فرض نوار کاملاً مخفی است که در این پروژه قابل استفاده نیست. زیرا برای بالا آوردن نوار مخفی استفاده از جاوا لازم است که در این پروژه قصد کار با جاوا را نداریم.

در قسمت قبل با کشیدن نوار به سمت پایین، در ارتفاع اولیه یعنی 60dp متوقف می‌شود. اما با استفاده از خاصیت `behavior_hideable` و مقدار `true`، نوار به طور کلی به پایین صفحه رفته و محو می‌شود:

```
app:behavior_hideable="true"
```

بدیهی است مقدار `false` برای خاصیت فوق، مانند قسمت قبل عمل کرده و نوار کاملاً مخفی نمی‌شود.

سورس کامل `activity_main.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:orientation="vertical"
        android:background="#009688"
        android:padding="8dp"
        app:behavior_peekHeight="60dp"
        app:behavior_hideable="true"
        app:layout_behavior="@string/bottom_sheet_behavior">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#fff"
            android:textSize="20sp"
            android:textStyle="bold"
            android:text="Bottom Sheet Title" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#fff"
            android:textSize="16sp"
            android:text="Lorem ipsum dolor sit amet, te sed vide delicata,
per ex salutandi intellegat temporibus, ei insolens molestiae vis.
Eum ei possim aperiam, fuisset suscipit vim ut. Voluptua repudiare
gubergren id eum, nullam labores an nam. Sed et tota quaerendum,
enim elit democritum quo et.
Lorem ipsum dolor sit amet, te sed vide delicata,
per ex salutandi intellegat temporibus, ei insolens molestiae vis.
Eum ei possim aperiam, fuisset suscipit vim ut. Voluptua repudiare
gubergren id eum, nullam labores an nam. Sed et tota quaerendum,
enim elit democritum quo et." />

    </LinearLayout>

</CoordinatorLayout>
```





```
</LinearLayout>
</android.support.design.widget.CoordinatorLayout>
```

**تذکر:** سورس فوق با نام activity\_main\_linear.xml در کنار سورس اصلی پروژه قرار داده شده است.

در قسمت قبل، layout نوار Bottom Sheet از نوع LinearLayout بود. من در TextView دوم یک متن نسبتاً طولانی استفاده کردم. اما در حال حاضر فقط قسمتی از متن و بطور کل محتوای نوار برای کاربر نمایش داده می‌شود و مابقی آن مخفی است. یعنی فقط به اندازه 200dp را نشان می‌دهد و قابلیت اسکرول فعال نیست تا کاربر بتواند ادامه محتوا را نیز مشاهده کند. جهت رفع این مساله لازم است یک NestedScrollView را جایگزین LinearLayout کنیم:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".MainActivity">

    <android.support.v4.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:background="#009688"
        android:padding="8dp"
        app:behavior_peekHeight="60dp"
        app:behavior_hideable="true"
        app:layout_behavior="@string/bottom_sheet_behavior">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textColor="#fff"
                android:textSize="20sp"
                android:textStyle="bold"
                android:text="Bottom Sheet Title" />

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textColor="#fff"
                android:textSize="16sp"
                android:text="Lorem ipsum dolor sit amet, te sed vide delicata,
per ex salutandi intellegat temporibus, ei insolens molestiae vis.
Eum ei possim aperiam, fuisset suscipit vim ut. Voluptua repudiare
gubergren id eum, nullam labores an nam. Sed et tota quaerendum,
enim elit democritum quo et.
Lorem ipsum dolor sit amet, te sed vide delicata,
per ex salutandi intellegat temporibus, ei insolens molestiae vis.
Eum ei possim aperiam, fuisset suscipit vim ut. Voluptua repudiare
```



```
gubergren id eum, nullam labores an nam. Sed et tota quaerendum,
enim elit democritum quo et." />

</LinearLayout>

</android.support.v4.widget.NestedScrollView>

</android.support.design.widget.CoordinatorLayout>
```

(دقت کنید داخل NestedScrollView جهت چینش عناصر از یک LinearLayout استفاده شده)

حالا با اجرای مجدد پروژه امکان اسکرول محتوا نیز فراهم شده و بعد از آنکه نوار به سمت بالا کشیده شد و در حداکثر ارتفاع خود یعنی 200dp قرار گرفت، با اسکرول مجدد به سمت بالا، ادامه محتوا نمایش داده می‌شود و به همین ترتیب برای اسکرول به سمت پایین.

(تصویر 2.gif را مشاهده کنید)

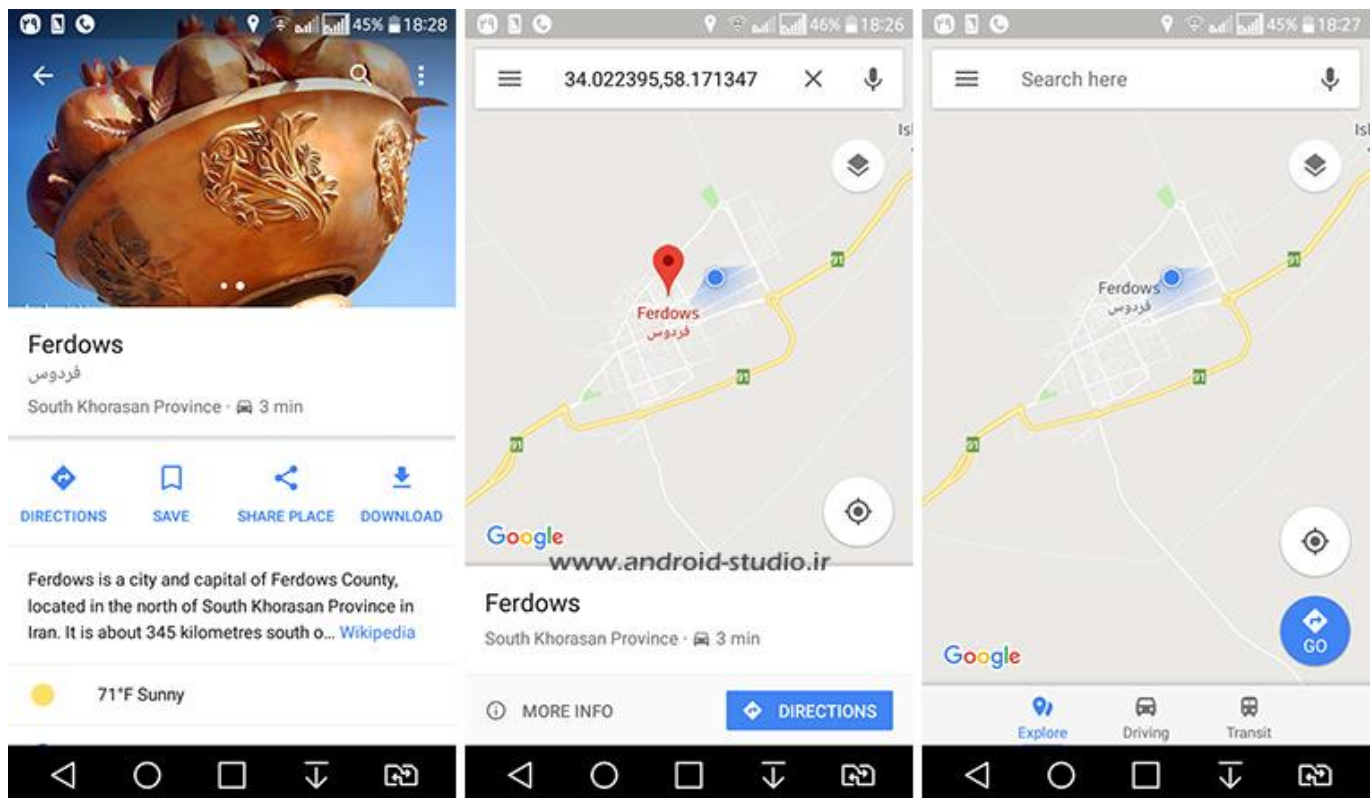
**تذکر:** پروژه فوق، با نام SimpleBottomSheet در پوشه Exercises قرار دارد.

در پروژه قبل با کلیت این کامپوننت متریالی آشنا شدیم. در ادامه قصد داریم دو نوع Persistent و Modal را در دو پروژه جداگانه بررسی کنیم. البته شاید نتوان از نظر تفاوت کاربرد این دو، یک مرز مشخص تعیین کرد اما بعد از مطالعه ادامه مبحث، تا حدود زیادی نوع کاربرد هریک از این دو مورد روشن می‌شود.

## :Persistent Bottom Sheet

از این حالت عموماً جهت نمایش محتوا استفاده می‌شود. مواردی مانند "جزئیات بیشتر"، "اطلاعات تکمیلی" و سایر موارد مشابه که بین محتوای اصلی موجود در صفحه و محتوای Bottom Sheet یک ارتباط وجود دارد و در واقع به نوعی این نوار نقش تکمیل کننده صفحه اصلی را دارد.

مجدد به مثال اولی که در ابتدای مبحث اشاره شد دقت کنید:



کاربر با انتخاب یک نقطه بر روی نقشه، ابتدا اطلاعات کلی از جمله نام مکان و مسافت را مشاهده می‌کند که در ادامه با کشیدن نوار به بالا یا انتخاب دکمه MORE INFO، توضیحات تکمیلی را نیز دریافت خواهد کرد. به عبارت دیگر، محتوای این نوار، اطلاعات داخلی برنامه محسوب می‌گردد و ارتباطی با خارج از آن ندارد.

یک پروژه با نام PersistentBottomSheet و یک Empty Activity ایجاد می‌کنم.

مانند پروژه قبل، اینجا هم لازم است کتابخانه design را اضافه کنم.

در این پروژه می‌خواهم با کلیک روی یک دکمه، یک Bottom Sheet باز و بسته شود. بنابراین به دلیل آنچه در پروژه قبل ذکر شد، ابتدا layout ریشه را به CoordinatorLayout تغییر داده، سپس یک Button به آن اضافه می‌کنم:



:activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    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"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:id="@+id/btn_expand"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:text="Expand" />

    </RelativeLayout>

</android.support.design.widget.CoordinatorLayout>
```

CoordinatorLayout امکان مدیریت نحوه چینش عناصر را ندارد بنابراین دکمه را درون یک RelativeLayout قرار دادم. در مرحله بعد یک Bottom Sheet به layout اضافه می‌کنم:

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/bottom_sheet"
    android:background="#94aab4"
    android:padding="8dp"
    app:behavior_hideable="true"
    app:behavior_peekHeight="60dp"
    app:layout_behavior="@string/bottom_sheet_behavior">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:textSize="20sp"
        android:text="Location: Iran"
        android:paddingBottom="10dp"
        android:paddingTop="10dp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#452f2f"
        android:textSize="16sp"
        android:text="Iran is an Islamic republic on the Persian Gulf with historical
sites dating to the Persian Empire. Extensive marble ruins mark
Persepolis, the empire's capital founded by Darius I in the 6th
century B.C. The modern capital, Tehran, is home to opulent
Golestan Palace, seat of the Qajar Dynasty (1794-1925), plus
modern landmarks such as the 435m-high Milad Tower."/>

</LinearLayout>
```



در اینجا از NestedScrollView استفاده نکردم. ارتفاع هم wrap\_content تعیین کردم بنابراین نوار به اندازه محتوای درون آن باز خواهد شد.

بهتر است برای Bottom Sheet یک layout جداگانه ساخته و آنرا درون layout اکتیویتی اصلی include کنم تا نظم بیشتری به کار داده باشم. یک Layout جدید با نام bottom\_sheet.xml ایجاد کرده، کد فوق را درون آن قرار می‌دهم:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/bottom_sheet"
    android:background="#94aab4"
    android:padding="8dp"
    app:behavior_hideable="true"
    app:behavior_peekHeight="60dp"
    app:layout_behavior="@string/bottom_sheet_behavior">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:textSize="20sp"
        android:text="Location: Iran"
        android:paddingBottom="10dp"
        android:paddingTop="10dp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#452f2f"
        android:textSize="16sp"
        android:text="Iran is an Islamic republic on the Persian Gulf with historical
sites dating to the Persian Empire. Extensive marble ruins mark
Persepolis, the empire's capital founded by Darius I in the 6th
century B.C. The modern capital, Tehran, is home to opulent
Golestan Palace, seat of the Qajar Dynasty (1794-1925), plus
modern landmarks such as the 435m-high Milad Tower." />

</LinearLayout>
```



## :activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    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"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:id="@+id/btn_expand"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:text="Expand" />

    </RelativeLayout>

    <include layout="@layout/bottom_sheet" />
</android.support.design.widget.CoordinatorLayout>
```

در حال حاضر با اجرای پروژه، 60dp از ابتدای نوار نمایش داده می‌شود که با کشیدن به سمت بالا، نوار کامل باز خواهد شد. اما قصد داریم این عمل باز شدن توسط دکمه صورت پذیرد.



اکتیویتی را به اینصورت تکمیل می‌کنم:

```
package ir.android_studio.persistentbottomsheet;

import android.support.design.widget.BottomSheetBehavior;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;

public class MainActivity extends AppCompatActivity {

    Button btnSheet;
    LinearLayout sheetLayout;
    BottomSheetBehavior bottomSheet;

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

        btnSheet = findViewById(R.id.btn_expand);
        sheetLayout = findViewById(R.id.bottom_sheet);
        bottomSheet = BottomSheetBehavior.from(sheetLayout);

        btnSheet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                bottomSheet.setState(BottomSheetBehavior.STATE_EXPANDED);

            }
        });
    }
}
```

ابتدا Button و LinearLayout مربوط به BottomSheet را توسط id ای که قبلا به آنها اختصاص دادم تعریف کردم. سپس یک نمونه از متد BottomSheetBehavior با نام دلخواه bottomSheet ساختم که توسط متد from(). لایه مربوطه را به آن معرفی کرده ام.

**نکته:** در تعریف sheetLayout می‌توان از View بجای LinearLayout یا NestedScrollView ... استفاده کرد. زیرا همه اینها از View ارث بری شده اند:

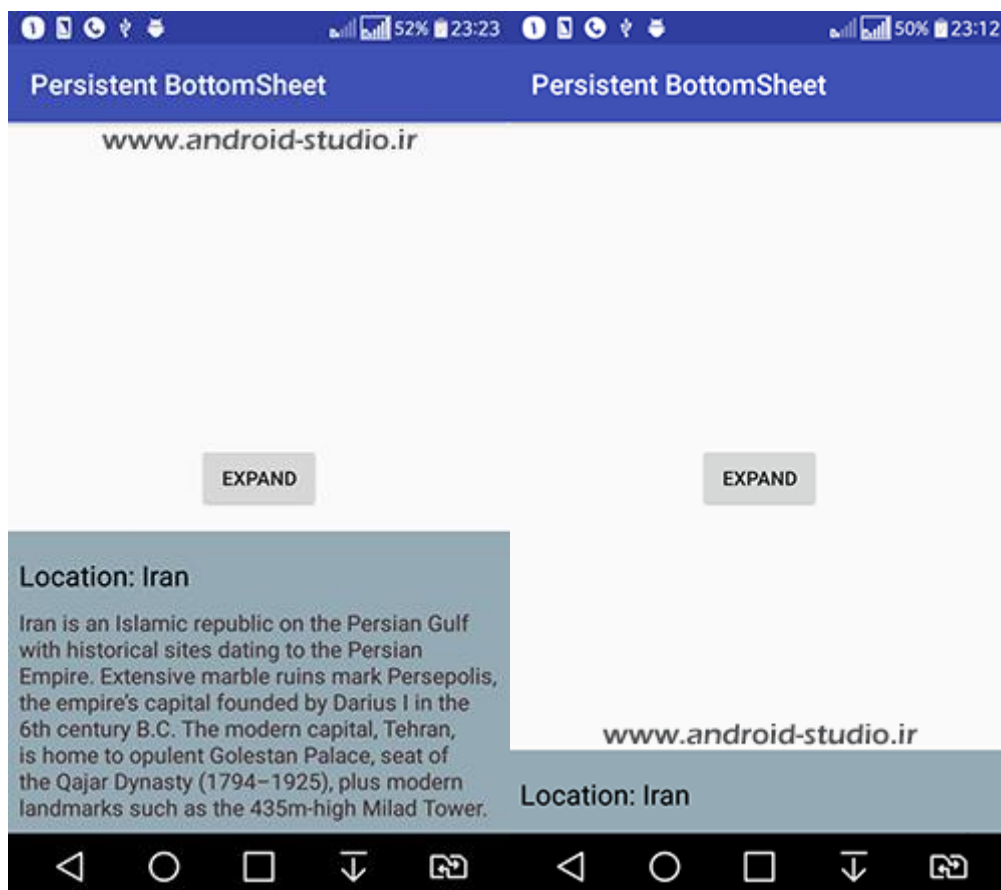
```
View sheetLayout;
```





در ادامه برای دکمه یک Listener ایجاد کردم. از متد `setState()` جهت تعیین حالت Bottom Sheet استفاده می‌شود. من حالت `STATE_EXPANDED` را انتخاب کردم، یعنی با لمس دکمه، نوار در وضعیت Expanded (باز شده) قرار می‌گیرد.

پروژه را اجرا و تست می‌کنم:



عملیات به درستی انجام می‌شود و نوار باز شد. در قدم بعد می‌خواهم با لمس دوباره دکمه، نوار مجدداً بسته شود:



```
btnSheet.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (bottomSheet.getState() == BottomSheetBehavior.STATE_COLLAPSED) {

            bottomSheet.setState(BottomSheetBehavior.STATE_EXPANDED);

        }
        else if (bottomSheet.getState() == BottomSheetBehavior.STATE_EXPANDED) {

            bottomSheet.setState(BottomSheetBehavior.STATE_COLLAPSED);

        }

    }
});
```

با استفاده از if else و متد getState() شرطی تعیین کردم تا اگر نوار در وضعیت Collapsed بود، با لمس دکمه به وضعیت Expanded تغییر کند و بالعکس.

در هریک از دو نقش فوق، دکمه عبارت Expand را نشان می‌دهد که منطقی نیست. هنگامی که نوار در وضعیت Expand قرار دارد، نقش دکمه Collapse کردن آن است بنابراین عنوان دکمه نیز باید تغییر کند که با استفاده از setText() درون شرط این اصلاح را صورت می‌دهم:

```
btnSheet.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (bottomSheet.getState() == BottomSheetBehavior.STATE_COLLAPSED) {

            bottomSheet.setState(BottomSheetBehavior.STATE_EXPANDED);
            btnSheet.setText("Collapse");

        }
        else if (bottomSheet.getState() == BottomSheetBehavior.STATE_EXPANDED) {

            bottomSheet.setState(BottomSheetBehavior.STATE_COLLAPSED);
            btnSheet.setText("Expand");

        }

    }
});
```

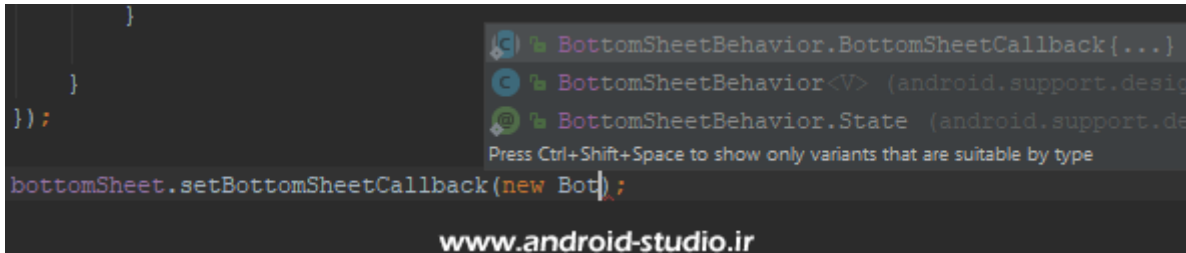
با اجرای مجدد پروژه، نوار در هر وضعیتی که باشد، دکمه عبارت متضاد آنرا نشان می‌دهد.

هنوز یک ایراد دیگر داریم. در حال حاضر اگر وضعیت نوار را فقط با دکمه تغییر دهید همه چیز به خوبی انجام شده و با تغییر حالت نوار، text دکمه هم تغییر می‌کند. اما در صورتی که کاربر نوار را به صورت عادی به طرف بالا یا پایین بکشد، تغییری در text انجام نمی‌شود که یک خطای منطقی است. یعنی اگر



دکمه عبارت Expand را نشان می‌دهد و کاربر نوار را بدون استفاده از دکمه Expand کند (بالا ببرد)، باز هم دکمه کلمه Expand را نشان می‌دهد در صورتی که باید با کلمه Collapse جایگزین شود.

در اینجا متدی داریم با نام `setBottomSheetCallback` که امکان مدیریت Bottom Sheet را فراهم می‌کند:



```
bottomSheet.setBottomSheetCallback(new BottomSheetBehavior.BottomSheetCallback() {
    @Override
    public void onStateChanged(@NonNull View bottomSheet, int newState) {

    }

    @Override
    public void onSlide(@NonNull View bottomSheet, float slideOffset) {

    }
});
```

این متد دو تابع `onSlide` و `onStateChanged` دارد که برای رفع مشکل ایجاد شده از مورد اول استفاده می‌کنیم:

```
public void onStateChanged(@NonNull View bottomSheet, int newState) {

    if (newState == BottomSheetBehavior.STATE_EXPANDED) {
        btnSheet.setText("Collapse");
    } else if (newState == BottomSheetBehavior.STATE_COLLAPSED) {
        btnSheet.setText("Expand");
    }

}
```

در اینجا با استفاده از `newState()` یک شرط تعیین شده. به اینصورت که اگر وضعیت جدید روی Expanded بود، دکمه کلمه Collapse را نشان دهد و بالعکس. حالا با اجرای مجدد پروژه همه چیز به درستی کار می‌کند.

(تصویر 3.gif را مشاهده کنید)



با تعیین این شرط، عملاً `setText` هایی که در `Listener` دکمه تعریف کرده بودم کارایی ندارد و آنها را حذف می‌کنم (در سورس پروژه `comment` شده است).

:MainActivity.java

```
package ir.android_studio.persistentbottomsheet;

import android.support.annotation.NonNull;
import android.support.design.widget.BottomSheetBehavior;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;

public class MainActivity extends AppCompatActivity {

    Button btnSheet;
    LinearLayout sheetLayout;
    BottomSheetBehavior bottomSheet;

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

        btnSheet = findViewById(R.id.btn_expand);
        sheetLayout = findViewById(R.id.bottom_sheet);
        bottomSheet = BottomSheetBehavior.from(sheetLayout);

        btnSheet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (bottomSheet.getState() == BottomSheetBehavior.STATE_COLLAPSED) {

                    bottomSheet.setState(BottomSheetBehavior.STATE_EXPANDED);
                    //btnSheet.setText("Collapse");

                }
                else if (bottomSheet.getState() == BottomSheetBehavior.STATE_EXPANDED) {

                    bottomSheet.setState(BottomSheetBehavior.STATE_COLLAPSED);
                    //btnSheet.setText("Expand");

                }

            }
        });

        bottomSheet.setBottomSheetCallback(new BottomSheetBehavior.BottomSheetCallback() {
            @Override
            public void onStateChanged(@NonNull View bottomSheet, int newState) {

                if (newState == BottomSheetBehavior.STATE_EXPANDED) {
                    btnSheet.setText("Collapse");
                }
                else if (newState == BottomSheetBehavior.STATE_COLLAPSED) {
                    btnSheet.setText("Expand");
                }

            }

        });

    }

    @Override
```



```
        public void onSlide(@NonNull View bottomSheet, float slideOffset) {  
            }  
        }  
    }  
}
```

دو خاصیت behavior\_peekHeight و behavior\_hideable را به جای xml می‌توان در جاوا نیز تعریف و کنترل کرد (در سورس کامنت شده):

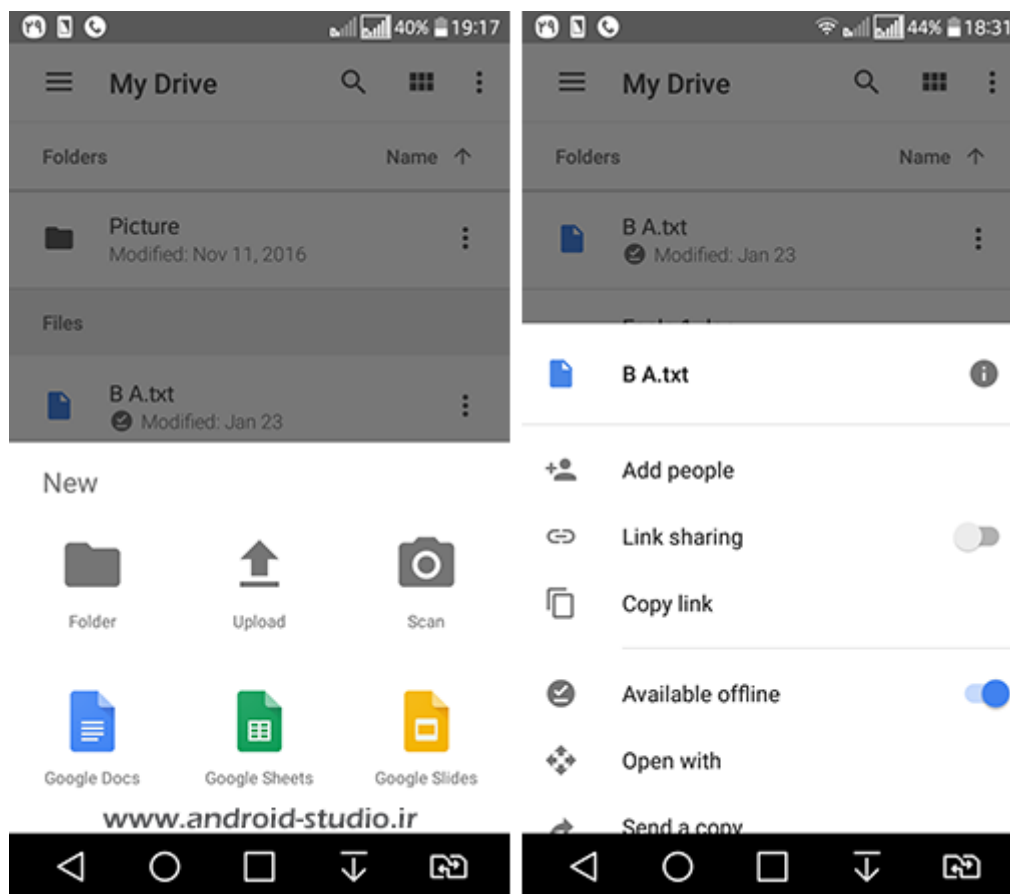
```
bottomSheet.setPeekHeight(0);  
bottomSheet.setHideable(true);
```

**تذکر:** پروژه فوق، با نام PersistentBottomSheet در پوشه Exercises قرار دارد.



## :Modal Bottom Sheet

در این نوع، در حقیقت ما یک Dialog (مشابه AlertDialog) را به عنوان یک Bottom Sheet نمایش می‌دهیم. مانند مثال دوم ابتدای مبحث. یعنی Google Drive:



از Modal عموماً جهت نمایش گزینه‌هایی مانند Upload، Copy، Share و سایر اکشن‌های مشابه استفاده می‌شود.

یک پروژه جدید با نام ModalBottomSheet و یک Empty Activity ایجاد می‌کنم.

مشابه موارد قبل ابتدا کتابخانه Support Design را به پروژه اضافه می‌کنم.

activity\_main.xml را مشابه پروژه قبل می‌سازم با این تفاوت که layout ای که include شده بود حذف می‌شود:



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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"
tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:id="@+id/btn_expand"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:text="Expand" />

    </RelativeLayout>

</android.support.design.widget.CoordinatorLayout>
```

برای نمایش Bottom Sheet در قالب یک Dialog، یک **فرگمنت** می‌سازیم. بعد از ساخته شدن فرگمنت، باید ارث بری آن را از Fragment به BottomSheetDialogFragment تغییر دهیم:

:BottomSheetFragment.java

```
package ir.android_studio.modalbottomsheet;

import android.os.Bundle;
import android.support.design.widget.BottomSheetDialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class BottomSheetFragment extends BottomSheetDialogFragment {

    public BottomSheetFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_bottom_sheet, container, false);
    }
}
```





**نکته:** اگر بعد از ساخت فرگمنت، ارور مربوط به کتابخانه support-v4 گرفتید، نیاز به تغییر آی پی و نصب آن نیست. این کتابخانه برای ایمپورت شدن android.support.v4.app.Fragment اضافه می‌شود که ما در این پروژه به آن نیازی نداریم و بجای Fragment از BottomSheetDialogFragment استفاده می‌کنیم. بنابراین جهت رفع مشکل کافی است com.android.support:support-v4:xx.x.x را از build.gradle حذف کنید.

حذف import android.support.v4.app.Fragment از کلاس فرگمنت فراموش نشود.

حالا باید layout این فرگمنت را تکمیل کنم. همان محتوایی که برای bottom\_sheet.xml پروژه قبل ساخته بودم را در اینجا استفاده می‌کنم:

:fragment\_bottom\_sheet.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/bottom_sheet"
    android:background="#94aab4"
    android:padding="8dp"
    app:behavior_hideable="true"
    app:behavior_peekHeight="60dp"
    app:layout_behavior="@string/bottom_sheet_behavior">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:textSize="20sp"
        android:text="Location: Iran"
        android:paddingBottom="10dp"
        android:paddingTop="10dp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#452f2f"
        android:textSize="16sp"
        android:text="Iran is an Islamic republic on the Persian Gulf with historical
sites dating to the Persian Empire. Extensive marble ruins mark
Persepolis, the empire's capital founded by Darius I in the 6th
century B.C. The modern capital, Tehran, is home to opulent
Golestan Palace, seat of the Qajar Dynasty (1794-1925), plus
modern landmarks such as the 435m-high Milad Tower." />

</LinearLayout>
```



در مرحله نهایی، یک متد Listener برای دکمه Expand می‌نویسم و داخل آنرا به صورت زیر تکمیل می‌کنم:

```
package ir.android_studio.modalbottomsheet;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button btnSheet;

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

        btnSheet = findViewById(R.id.btn_expand);

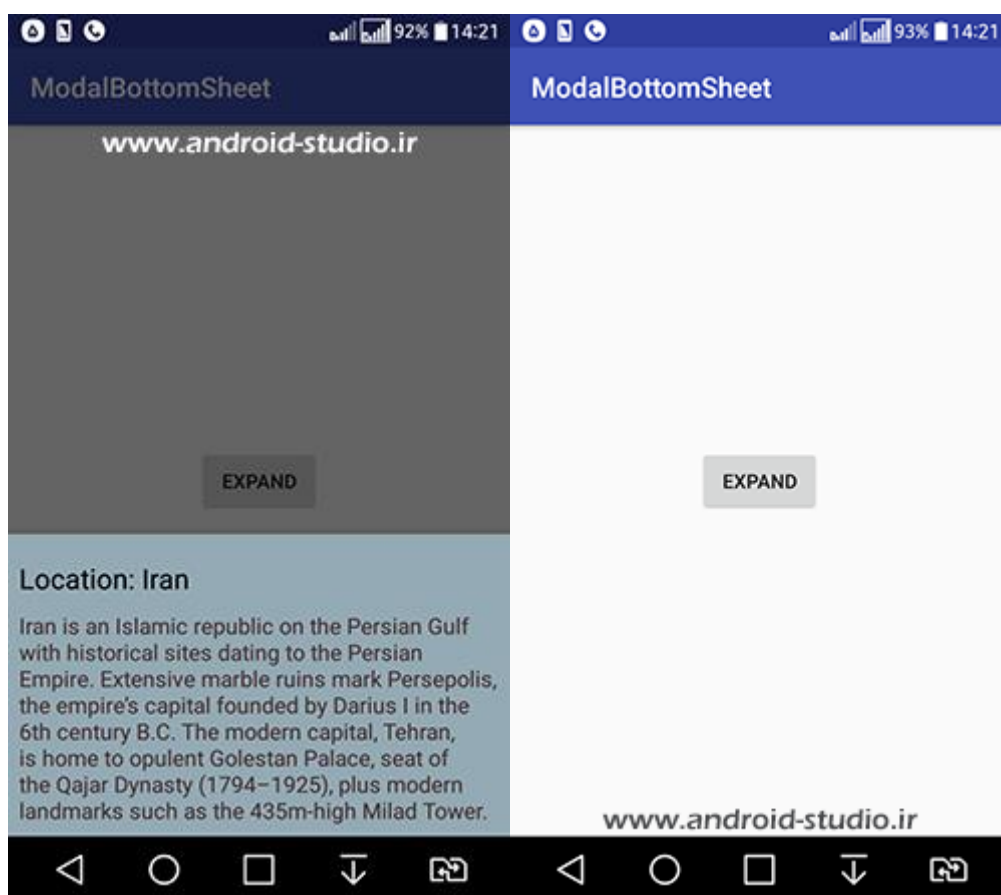
        btnSheet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                BottomSheetFragment bottomFragment = new BottomSheetFragment();
                bottomFragment.show(getSupportFragmentManager(), bottomFragment.getTag());

            }
        });
    }
}
```

ابتدا از فرگمنت یک نمونه با نام bottomFragment ساخته شده. سپس توسط متد show() نمایش داده می‌شود. این متد دو پارامتر ورودی می‌گیرد که اولی FragmentManager و دومی Tag فرگمنت مربوطه است.

پروژه را اجرا می‌کنم:



مانند AlertDialog، این نوار نیز دارای یک elevation (ارتفاع) زیاد است و با دکمه back دیوایس و یا کشیدن نوار به سمت پایین حذف می‌شود.

**تذکر:** پروژه فوق، با نام ModalBottomSheet در پوشه Exercises قرار دارد.

### منابع تکمیلی:

<https://developer.android.com/reference/android/support/design/widget/BottomSheetBehavior>

<https://developer.android.com/reference/android/support/design/widget/BottomSheetDialog>

<https://developer.android.com/reference/android/support/design/widget/BottomSheetDialogFragment>

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش های بهتر یاری فرمائید.

[www.android-studio.ir](http://www.android-studio.ir)