



آموزش برنامه نویسی اندروید در محیط اندروید استودیو

کار با نوتیفیکیشن (Notification)

بخش چهارم: ارسال پوش نوتیفیکیشن توسط سرویس ایرانی Pushe

مدرس : سیدمهدی مطهری

www.android-studio.ir



به نام خدا

در مبحث قبل با قابلیت ارسال پوش نوتیفیکیشن در اندروید آشنا شدیم. در این قسمت از سری مباحث آموزش برنامه نویسی اندروید به نحوه ارسال پوش نوتیفیکیشن توسط سرویس ایرانی پوشه (Pusher) و همچنین معرفی امکانات این سرویس از جمله ارسال اعلان بر اساس موقعیت مکانی، نوع اتصال کاربر، زمان ارسال و... پرداخته می‌شود.

معرفی سرویس ایرانی پوشه



پوشه یکی از سرویس‌های مطرح داخلی جهت ارسال پوش نوتیفیکیشن در پلتفرم‌های iOS، Android و Web است (البته نسخه iOS هنوز راه اندازی نشده). این سرویس قابلیت‌های متعددی را در اختیار توسعه دهنده و مدیران اپلیکیشن یا وب سایت‌ها قرار می‌دهد که با مراجعه به صفحه اصلی وب سایت آن به نشانی pushe.co می‌توانید توضیحات کامل را مطالعه بفرمایید.

قابلیت‌های کلیدی سرویس را در تصویر زیر مشاهده می‌کنید:



مقیاس‌پذیر و آماده

نگران رشد ناگهانی تعداد کاربران‌تان نباشید. پوشه برای توسعه سریع کسب‌وکار شما آماده است.



شنا و پاسخ‌گو

دغدغه دسترسی به پشتیبانی را نداشته باشید. پوشه امروز حاصل بازخوردهای دیروز شماست.



رایگان و قابل اعتماد

پوشه با امکانات اولیه کاملاً رایگان است اما در بسته‌های پولی خدمات بیشتری به شما داده خواهد شد.



در همه پلتفرم‌ها

هیچ فرصتی را برای ارتباط با کاربر از دست ندهید. پوشه در پلتفرم‌های اندروید، iOS و وب کنار شماست.



فیلترهای کاربردی

برای کاربران ایرانی خود با تفکیک شهر، اپراتور تلفن همراه و ... اعلان بفرستید.



آنالیتیکس اپلیکیشن‌ها

با تکیه بر گزارش‌های متنوع و آمارهایی از فعالیت کاربران قدم بعدی توسعه کسب‌وکارتان را آگاهانه بردارید.



امکان کار با API

ارتباطات سرور و کلاینت را با راهکارهای برنامه‌نویسانی که کاملاً متناسب با کسب‌وکار شماست مدیریت کنید.



پیاده‌سازی ۱۰ دقیقه‌ای

سهولت استفاده از SDK پوشه کار نصب کتابخانه را برای توسعه‌دهندگان تسریع کرده است.

تقریباً تمامی امکانات اولیه مورد نیاز در پلن رایگان پوشه در دسترس است و بجز موارد خاص، نیازی به پرداخت هزینه‌ای نخواهیم داشت.

در ادامه مبحث و در قالب یک پروژه آزمایشی، ابتدا سرویس Pushe را روی یک اپلیکیشن پیاده سازی و رجیستر کرده و سپس با نحوه ارسال پوش نوتیفیکیشن و امکانات جانبی آن آشنا می‌شویم.

ارسال پوش نوتیفیکیشن توسط پوشه

برای استفاده از سرویس ابتدا باید ثبت نام کنیم. بعد از ثبت نام پنل کاربری در اختیار ما قرار می‌گیرد:



گزینه اندروید به صورت پیش فرض انتخاب شده. روی دکمه‌ی "اپلیکیشن جدید" کلیک می‌کنم:

در مرحله نخست نام اپلیکیشن، نام بسته (Package name) و ابزار مورد استفاده را باید ثبت کنم. نام اپلیکیشن صرفاً برای ثبت در پل کاربری است و لزوماً نباید حتماً با نام حقیقی اپلیکیشن یکسان باشد.



اما مورد دوم یعنی نام پکیج باید عینا همان چیزی باشد که در هنگام ساخت پروژه جدید در اندروید استودیو تعیین می‌کنیم.

من یک پروژه با نام Pushe و نام بسته ir.android_studio.pushe ساختم که فقط یک Empty Activity دارد. فقط امیدوارم در حین مطالعه آموزش، نام پروژه را با کلاس مربوط به کتابخانه اشتباه نگیرید. بهتر بود نام دیگری برای پروژه انتخاب می‌کردم اما در اواسط تهیه آموزش این نکته به ذهنم رسید و فرصت اصلاح هم نبود. بنده را خواهید بخشید.

پروژه در محیط اندروید استودیو ساخته شده بنابراین گزینه ابزار مورد استفاده، همین آیتم را انتخاب می‌کنم. با کلیک روی دکمه ایجاد اپلیکیشن، اپلیکیشن ثبت شده و مراحل رجیستر آغاز می‌شود. در مرحله دوم از ما خواسته تا کتابخانه پوشه را به پروژه اضافه کنیم:

توصیه می‌کنم قبل از طی مراحل، راهنمای مربوط به اندروید استودیو پوشه را مطالعه کنید (این لینک). برای مثال در این داکيومنت تاکید شده در build.gradle پروژه حتما هردو مخزن google() و jcenter() تعریف شده باشند.

پس از اطمینان از مورد فوق، کتابخانه پوشه را به build.gradle (Module:app) اضافه کرده، سینک می‌کنم:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'co.ronash.android:pushe-base:1.6.3'
}
```



اگر در دریافت کتابخانه و سینک شدن پروژه مشکل دارید، **مبحث تغییر آی پی** را مطالعه بفرمائید.
پس از sync شدن پروژه، در پنل پوشه روی دکمه "بعدی" کلیک می‌کنم:

مانیفست

متن زیر را کپی کنید و در تگ application مانیفست خود درج نمایید.

```
<meta-data android:name="co.ronash.pushe.token" android:value="PUSHE_354401661489"/>
```

متن زیر را درون تگ manifest در ابتدای فایل مانیفست خود قرار دهید.

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

در صورتی که مایل نیستید از امکانات موقعیت‌محور پوشه استفاده کنید می‌توانید دسترسی بالا را در پروژه قرار ندهید.

www.android-studio.ir

← بعدی بستن → قبلی

در این قسمت دو تگ معرفی شده که می‌بایست در AndroidManifest.xml اضافه شود. با تعریف این تگ‌ها، هنگام نصب اپ توسط کاربر، مجوز دسترسی به موقعیت مکانی گرفته می‌شود. همانطور که در انتهای کادر بیان شده، این دسترسی الزامی نیست و تنها در صورتی باید به مانیفست اضافه شود که بخواهیم از امکانات موقعیت محور پوشه (ارسال پوش نوتیفیکیشن بر اساس موقعیت جغرافیایی و همچنین مشاهده موقعیت جغرافیایی کاربران) استفاده کنیم. تگ‌ها به اینصورت به مانیفست اضافه شد:



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ir.android_studio.pushe">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <meta-data android:name="co.ronash.pushe.token" android:value="PUSHE_354401661489"/>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

در مرحله بعد باید متد Pushe.initialize در onCreate اکتیویتی تعریف شود:



تغییرات در کد برنامه

- در MainActivity برنامه خود در قسمت import کتابخانه پوشه را اضافه کنید.

کپی

```
import co.ronash.pushe.Pushe;
```

- در onCreate کد روبرو را وارد کنید.

کپی

```
Pushe.initialize(this,true);
```

اگر در تابع بالا مقدار true گذاشته شود و دستگاه کاربر شما گوگل پلی سرویس نداشته باشد و یا آن را غیرفعال کرده باشد دیالوگی باز می شود و به کاربر پیشنهاد نصب گوگل پلی سرویس داده می شود تا قابلیت دریافت و ارسال پوش نوتیفیکیشن فعال شود. اگر در تابع بالا false را قرار دهید، این دیالوگ برای کاربر باز نمی شود.

www.android-studio.ir

← بعدی

بستن

→ قبلی



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

    Pushe.initialize(this, true);
}
```

در مرحله بعد که آخرین مرحله است، از ما خواسته شده تا پروژه را اجرا کرده و روی دکمه "بررسی و دریافت نصب" کلیک کنیم تا نصب پوشه بر روی اپلیکیشن بررسی و تایید گردد.

۴

تست و عیب یابی

- اپلیکیشن خود را (که کتابخانه‌ی پوشه به آن اضافه شده‌است) روی یک گوشی موبایل یا ایمولاتور نصب کنید
- مطمئن شوید google play services روی دستگاه شما نصب است
- از اتصال اینترنت دستگاه اطمینان حاصل فرمایید
- اپلیکیشن را روی گوشی خود باز کنید

بعد با فشردن دکمه‌ی «بررسی و دریافت نصب» در پایین این صفحه منتظر شوید تا نصب شما در سیستم پوشه شناخته شود.

به طور معمول ۳ دقیقه زمان لازم است تا نصب شما دریافت شود. در صورتی که بیش از این طول کشید روی دستگاه‌های دیگر نصب کنید و در صورتی که همچنان موفق نشدید مراحل را مجدداً بررسی کنید. همواره می‌توانید با مراجعه به صفحه‌ی نصب‌ها از رسیدن نصب جدید مطلع شوید.

بررسی و دریافت نصب

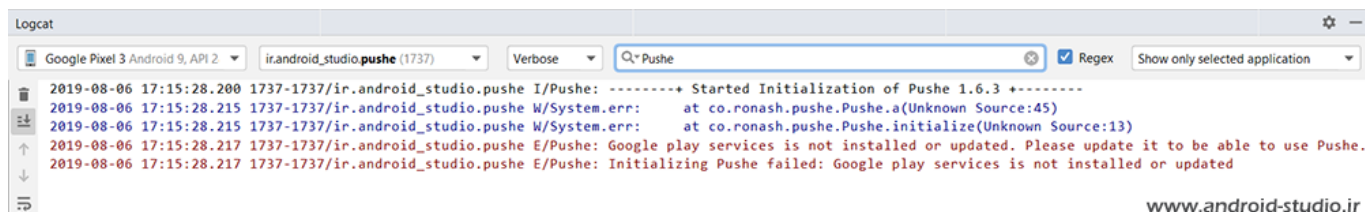


www.android-studio.ir

بستن

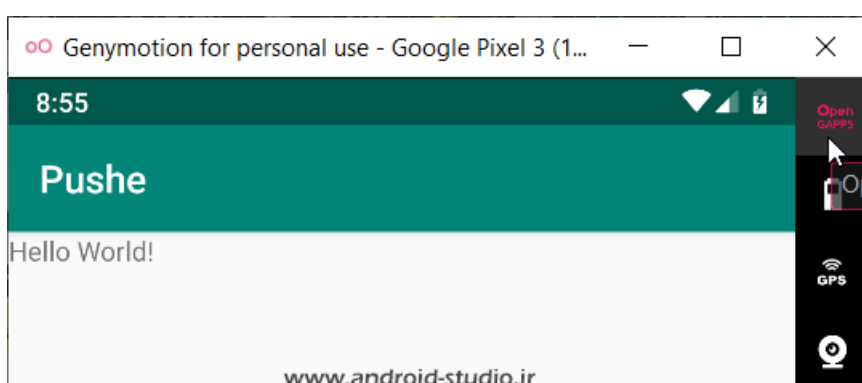
→ قبلی

به نکات فوق دقت کنید. اتصال به اینترنت و همچنین نصب بودن Google Play Services روی دیوایس ضروری است. پروژه را روی **شبیه ساز اندروید** اجرا می‌کنم. پس از اجرای کامل پروژه در قسمت Logcat اندروید استودیو پیغام‌های مربوط به سرویس Pushe را بررسی می‌کنم:

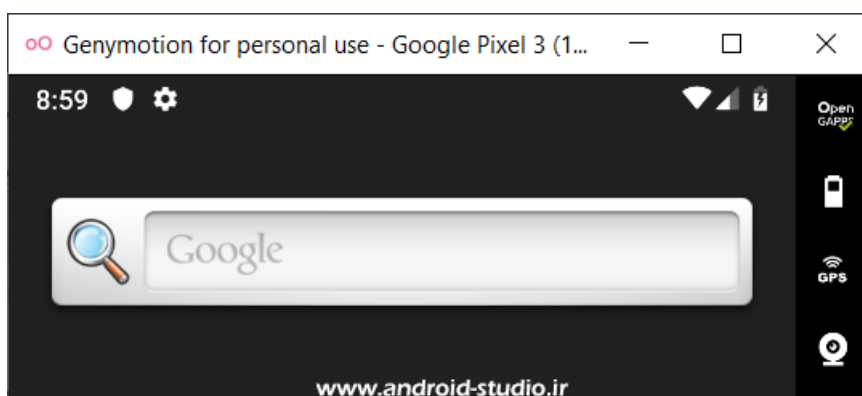


ملاحظه می‌کنید به دلیل نصب نبودن (یا بروز نبودن) Google play services روی دیوایس مجازی من، نصب پوشه انجام نشد.

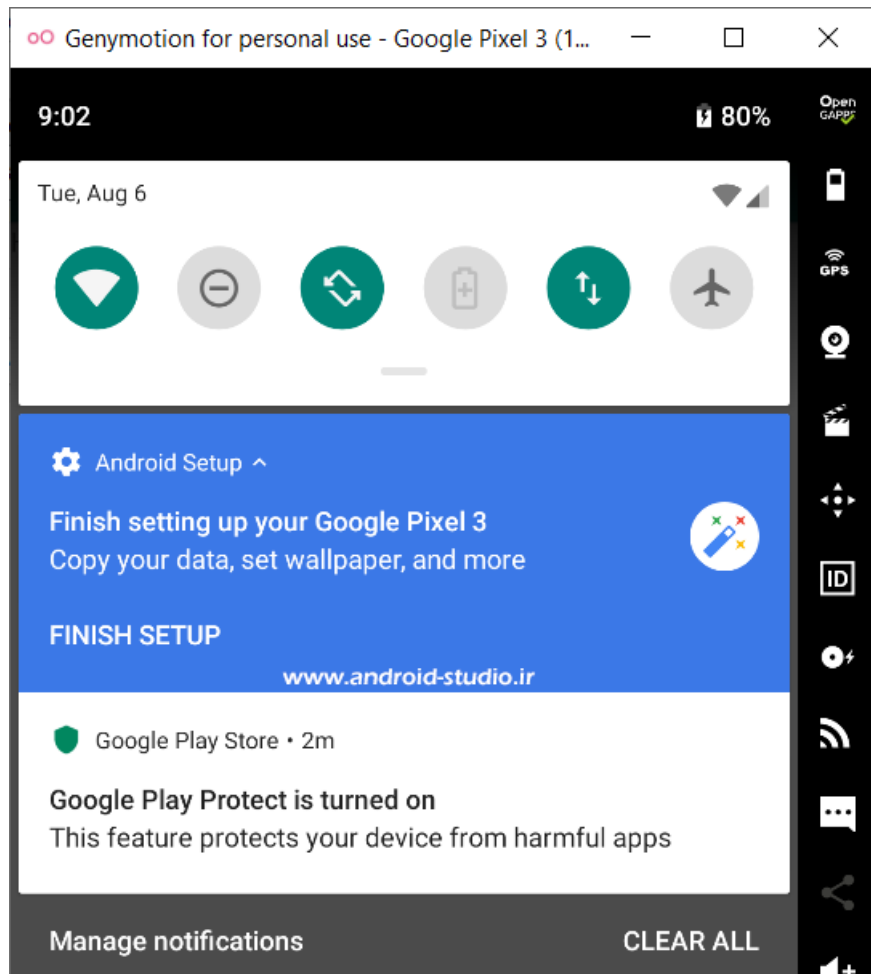
در شبیه ساز Genymotion با نصب Open GAPPs گوگل پلی سرویس روی دیوایس فعال می‌شود. روی دکمه Open GAPPs کلیک کرده و قوانین را Accept می‌کنم تا عملیات دانلود و نصب آغاز شود:



در صورتی که نصب با موفقیت صورت پذیرد، یک تیک سبز رنگ در کنار آیکون آن قرار می‌گیرد:



در صورتی که اعلانی مشابه تصویر زیر دریافت کردید روی FINISH SETUP کلیک کنید تا مراحل لازم طی شود. برای سادگی کار در تمامی مراحل Skip را انتخاب کنید.



مجدد پروژه را اجرا کرده و سپس روی دکمه "بررسی و دریافت نصب" پنل پوشه کلیک می‌کنم:



تست و عیب یابی

- اپلیکیشن خود را (که کتابخانه‌ی پوشه به آن اضافه شده‌است) روی یک گوشی موبایل یا ایمولاتور نصب کنید
- مطمئن شوید google play services روی دستگاه شما نصب است
- از اتصال اینترنت دستگاه اطمینان حاصل فرمایید
- اپلیکیشن را روی گوشی خود باز کنید

بعد با فشردن دکمه‌ی «بررسی و دریافت نصب» در پایین این صفحه منتظر شوید تا نصب شما در سیستم پوشه شناخته شود.

به طور معمول ۳ دقیقه زمان لازم است تا نصب شما دریافت شود. در صورتی که بیش از این طول کشید روی دستگاه‌های دیگر نصب کنید و در صورتی که همچنان موفق نشدید مراحل را مجدداً بررسی کنید.

همواره می‌توانید با مراجعه به صفحه‌ی نصب‌ها از رسیدن نصب جدید مطلع شوید.

مدل دستگاه	شناسه دستگاه	تاریخ نصب	نام برنامه	مورد علاقه
google pixel 3	pid_756a-a7c9-4a	۱۳۹۸/۵/۱۵ ۱۷:۳۰	کویر پلوند فردوس	★

www.android-studio.ir

بستن
→ قبلی

دیوایسی که پروژه روی آن اجرا شده به درستی شناسایی شد. به این ترتیب این اپلیکیشن با پکیج نیم `ir.android_studio.pushe` برای سرویس پوشه شناسایی و تایید شده و حالا می‌توانم برای آن پوش نوتیفیکیشن ارسال کنم.


تذکر: اگر از شبیه ساز AVD اندروید استودیو استفاده می‌کنید، باید دیوایسی را انتخاب کنید که در عنوان آن (with Google APIs) قید شده باشد. سایر نسخه‌ها شامل Google Play Services نیستند.


با کلیک روی دکمه "بستن" از ویزارد خارج شده و برای ارسال پوش نوتیفیکیشن، گزینه "ارسال اعلان" را انتخاب می‌کنم.




۱

نوع اعلان


 نمایش قالب HTML


 ارسال اطلاعات JSON


 نمایش اعلان

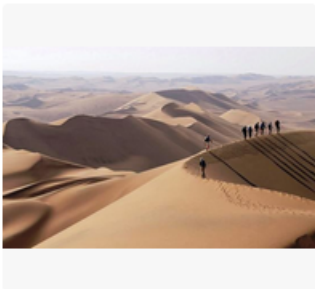
اپلیکیشن‌های من

همه اپلیکیشن‌ها
☐
انتخاب اپلیکیشن
☒

×

کویر پلوند فردوس ×

محتوای اعلان



حذف عکس

تیتراژ

تور دو روزه پلوند

متن اعلان

لذت کویر نوردی در تور دو روزه کویر زیبای پلوند فردوس

عملکرد مورد نظر را انتخاب کنید

باز کردن نرم‌افزار

دکمه‌های اعلان

+ دکمه جدید

www.android-studio.ir

← بعدی

ارسال نهایی

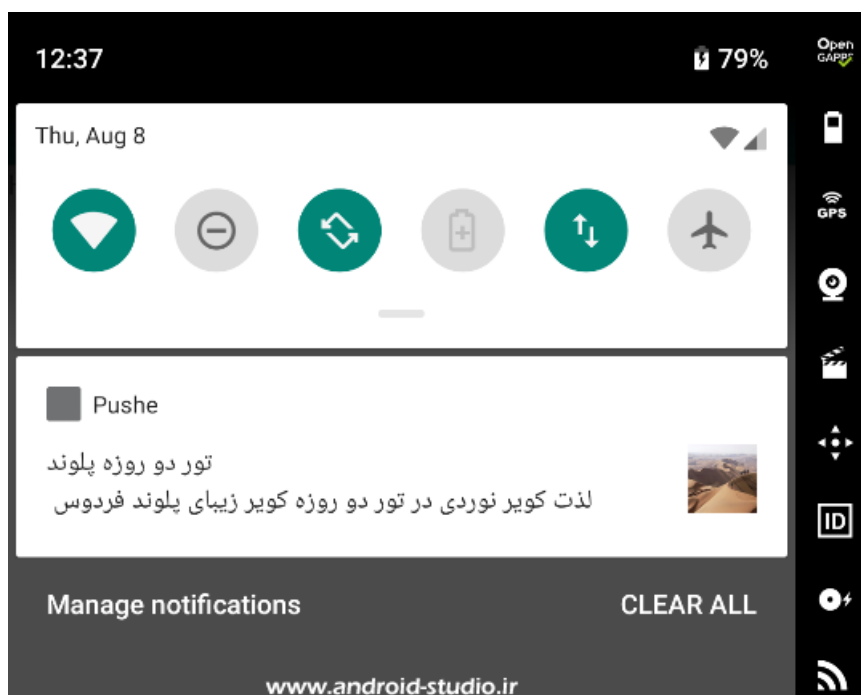
ارسال آزمایشی

در مرحله اول آیتم‌های اصلی اعلان را تنظیم می‌کنیم. نوع اعلان را همان نمایش اعلان انتخاب می‌کنم. در قسمت بعد اپلیکیشنی را انتخاب کرده‌ام که باید اعلان را دریافت کند. قسمت محتوای اعلان را نیز طبق میل و سلیقه خودمان تکمیل می‌کنیم. فعلا همین اعلان ساده را آزمایش می‌کنم. هردو گزینه "ارسال آزمایشی" و "ارسال نهایی" اعلان را به دیوایس(های) هدف ارسال می‌کنند. اما به جهت سادگی در اعمال تغییرات در مراحل بعد، ارسال آزمایشی را انتخاب می‌کنم تا ویزارد بسته نشود. با کلیک روی این دکمه، پنجره‌ای با عنوان "لیست دستگاه‌های مورد علاقه‌ی شما" باز شده که دیوایس من را نشان می‌دهد. در



واقع این همان دیوایسی است که در پروسه فعال شدن اعلان پوشه از آن استفاده کردم و سیستم پوشه به صورت پیش فرض آن را جزء دیوایس‌های مورد علاقه من انتخاب کرده بود.

با انتخاب دیوایس، اعلان ارسال می‌گردد. پس از مدت زمان کوتاهی پوش نوتیفیکیشن روی دیوایس مجازی من دریافت می‌شود:



تذکر: عدم دریافت اعلان دلایل متعددی می‌تواند داشته باشد. ابتدا از برقراری ارتباط اینترنت روی دیوایس مطمئن شوید. برای اطمینان بیشتر روی مرورگر دیوایس یک وبسایت را لود کنید. در دیوایس‌های مجازی ممکن است ارتباط مستمر مانند دیوایس‌های حقیقی پایدار نبوده و نیاز به بستن برنامه و باز کردن مجدد آن باشد. یا حتی بستن و راه اندازی دوباره خود دیوایس.

با لمس اعلان، صفحه اصلی برنامه باز می‌شود. این وظیفه‌ای است که به طور پیش فرض در تنظیمات اعلان در پوشه تعیین شده. اما عملکردهای دیگری هم می‌توان برای اعلان تعریف کرد:



محتوای اعلان



حذف عکس

تیترا

تور دو روزه پلوند

متن اعلان

لذت کویر نوردی در تور دو روزه کویر زیبای
پلوند فردوس

باز کردن لینک

باز کردن نرم افزار

صفحه اپ در مارکت

باز کردن دیالوگ

سایر اینتننت ها

باز کردن صفحه پیامک

باز کردن وب ویو

www.android-studio.ir

عملکردهایی شامل باز کردن لینک، صفحه یک اپ در مارکت (گوگل پلی، بازار، مایکت و ایران اپس)، باز کردن دیالوگ، استفاده از intent ها، باز کردن صفحه پیامک و تعیین یک متن پیش فرض، باز کردن شماره گیر و تعریف یک شماره تماس، باز کردن یک WebView، نمایش یک اکتیویتی خاص در اپلیکیشن اندروید، دعوت به کانال تلگرامی، باز کردن ایمیل و بستن اعلان.

به عنوان مثال با انتخاب گزینه "باز کردن لینک"، می توانم لینک ثبت نام تور کویرنوردی در وب سایت را وارد کنم که کاربر با لمس نوتیفیکیشن، به این لینک هدایت خواهد شد. سایر آیتم ها هرکدام کاربرد خاص خود را دارد که حتما با اکثر موارد آشنا هستید و تجربه آن را در اپ های موجود روی دیوایس خود داشته اید.

همچنین امکان تعریف دکمه (Action) تا نهایتا ۳ مورد وجود دارد:



دکمه‌های اعلان

باز کردن لینک مشاهده تور ✕

آدرس لینک *

<http://domain.com/polond-desert>

باز کردن نرم افزار ورود به برنامه ✕

+ دکمه جدید

www.android-studio.ir

با اعمال هر تغییر روی اعلان، پیش نمایش آن فوراً بروز می‌شود و قبل از ارسال، شکل کلی اعلان را مشاهده می‌کنیم. هرچند جزئیات ممکن است در نسخه‌های مختلف اندروید تفاوت‌هایی داشته باشند که در مبحث **آموزش ساخت نوتیفیکیشن در اندروید** به تعدادی از آنها اشاره کردم.



برای دیدن سایر تنظیمات اعلان روی دکمه "بعدی" کلیک می‌کنم:



عکس بزرگ ☐

محتوای تکمیلی اعلان
متن کامل ☒

تیتراژ کامل
تور دو روزه پلوند

متن کامل

فقط تا آخر ماه با 30 درصد تخفیف در تور استثنائی کویرگردی پلوند شرکت کنید

متن نوار اعلان
تخفیف تور کویرنوردی

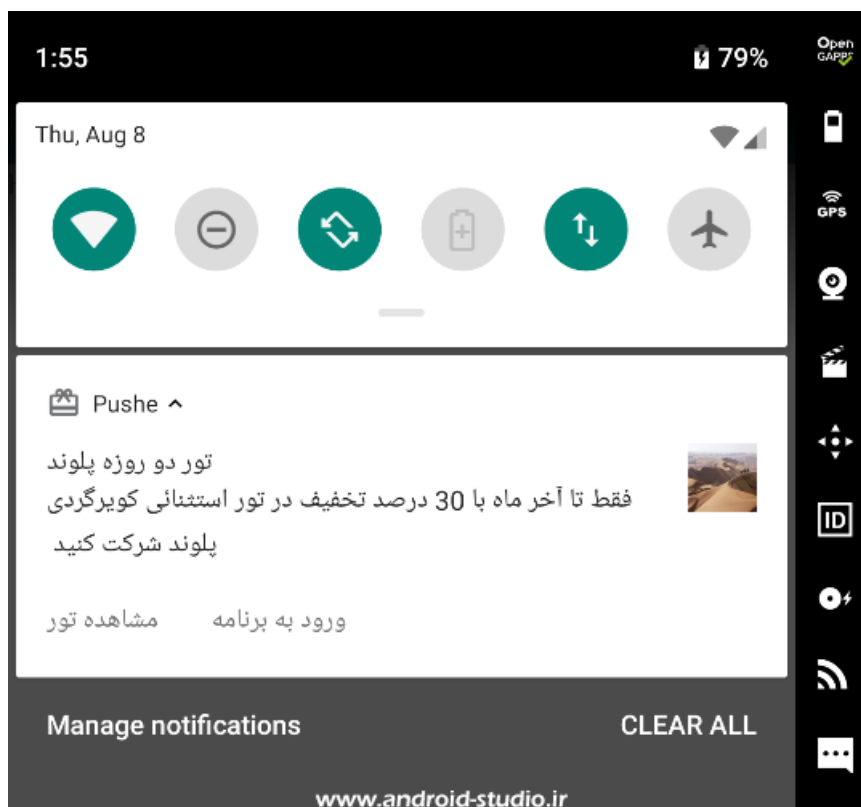
خلاصه

آیکون نوار اعلان ☒

www.android-studio.ir

← بعدی
ارسال نهایی
ارسال آزمایشی
→ قبلی

در این مرحله محتوای مربوط به نوتیفیکیشن در حالت باز شده (Expanded) را می‌توان تعیین کرد. علاوه بر آن، آیتم‌های متن نوار اعلان، خلاصه متن و همچنین آیکون اعلان نیز قابل تغییر هستند. یک ارسال آزمایشی جدید انجام می‌دهم:



قابلیت‌های فیلتر و تاپیک

در مرحله بعد دو قابلیت با عناوین "فیلتر" و "تاپیک" گنجانده شده است.

فیلتر: توسط پارامترهای متعدد می‌توانیم فیلترهایی اعمال کنیم تا اعلان فقط برای گروه خاصی از کاربران ارسال گردد:



تاپیک
فیلتر

فیلتر بر اساس

اتصال به اینترنت

ایراتور

موقعیت مکانی

نسخه اپلیکیشن

برند گوشی

تنظیمات بیشتر

☐ ارسال به کاربران کانال خاص

☒ ارسال به کاربران یکتا

www.android-studio.ir

← بعدی

ارسال نهایی

ارسال آزمایشی

→ قبلی

این فیلترها شامل موقعیت مکانی (انتخاب یک یا چند استان)، ایراتور (موبایل)، نوع اتصال کاربر به اینترنت (اینترنت ثابت یا موبایل)، برند گوشی، نسخه اپلیکیشن نصب شده روی دیوایس و ارسال به یک کانال خاص را شامل می‌شوند.

به عنوان مثال اگر بخواهم اعلان مربوط به تور کویرنوردی را عمدتاً کاربران استان‌های خراسان و همچنین استان‌های مجاور مانند یزد و سمنان دریافت کنند، با انتخاب این استان‌ها، تا حدود زیادی به این هدف دست خواهم یافت.

تاپیک: این قابلیت مشابه عضویت در خبرنامه‌های ایمیلی است. پوشه قابلیت را در اختیار ما قرار داده تا کاربران بتوانند در موضوعات متفاوتی عضو شده و صرفاً پوش‌های مرتبط با همان موضوع را دریافت کنند. به عنوان مثال من دکمه‌ای به اکتیویتی اضافه می‌کنم که با کلیک روی آن، این دیوایس در موضوعی



با نام desert (به معنی کویر) رجیستر شود. از این پس هرگاه در پنل پوشه، پوش نوتیفیکیشنی در تاپیک desert ارسال گردد، صرفا این دیوایس و سایر دیوایس‌هایی که در این موضوع ثبت نام کرده‌اند، پوش‌ها را دریافت می‌کنند.

مزیت این قابلیت در این است که می‌توان اعلان‌های تخصصی را صرفا برای افرادی ارسال کرد که مایل به دریافت آن هستند. در نتیجه هم تعامل افراد با اپ بیشتر شده و هم با اعلان‌های پراکنده و غیر ضروری برای کاربران مزاحمت ایجاد نمی‌شود. اگر شخصی در تاپیک "اخبار ورزشی" عضو شده باشد، شاید با دریافت روزانه ۵ پوش ورزشی هم مشکلی نداشته و حتی استقبال کند. اما قطعاً برای من که هیچ میانه‌ای با مسابقات ورزشی و اخبار آن ندارم، حتی دریافت یک اعلان در هفته می‌تواند آزاردهنده باشد.

یک Button به اکتیویتی اضافه نموده و توسط متد subscribe کلاس Pushe، نام تاپیک(ها) را تعیین می‌کنم:

:MainActivity.java

```
package ir.android_studio.pushe;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import co.ronash.pushe.Pushe;

public class MainActivity extends AppCompatActivity {

    Button newsButton;

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

        Pushe.initialize(this, true);

        newsButton = findViewById(R.id.btn_newsLetter);

        newsButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Pushe.subscribe(getApplicationContext(), "desert");
            }
        });
    }
}
```

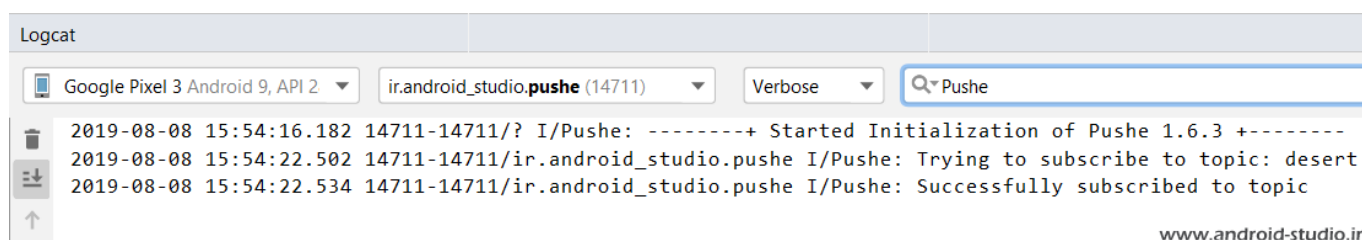


این متد دو ورودی می‌گیرد. اولی context برنامه و دومی نام تاپیک که در اینجا من desert وارد کردم. پروژه را اجرا کرده و روی دکمه کلیک می‌کنم:



نکته: در نامگذاری تاپیک از کاراکترهای زبان فارسی استفاده نکنید.

بعد از کلیک روی دکمه، لاگ کت اندروید استودیو را بررسی می‌کنم:



www.android-studio.ir



اپ با موفقیت در تاپیک desert عضو شد. حالا نام این تاپیک در پنل پوشه نمایش داده می‌شود که می‌توانیم اعلان را فقط برای اعضای این تاپیک ارسال کنیم:



پس تاپیک‌ها در خود برنامه باید تعریف شوند نه پنل پوشه. بعد از اولین عضویت در یک تاپیک، نام آن در پنل ما قابل نمایش خواهد بود.

نکته: اضافه شدن نام تاپیک به پنل پوشه ممکن است تا ۲۴ ساعت زمان ببرد. یا شاید لازم باشد یکبار از حساب کاربری خارج شده و مجدد وارد شوید تا تاپیک در لیست نمایش داده شود.

فقط دیوایسی می‌تواند در تاپیک‌ها عضو شود که قبلاً روی سرویس پوشه رجیستر شده باشد. بنابراین بهتر است کد مربوط به عضویت در تاپیک، درون این شرط قرار گیرد. به این ترتیب درخواست عضویت از جانب دیوایسی که به هر دلیلی هنوز روی پوشه شناسایی نشده، ارسال نمی‌گردد. اینکار توسط متد `isPusherInitialized` کلاس `Pusher` صورت می‌گیرد:

```
newsButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (Pusher.isPusherInitialized(getApplicationContext())) {
            Pusher.subscribe(getApplicationContext(), "desert");
        }
    }
});
```

همچنین توسط متد `Pusher.unsubscribe` کاربر می‌تواند لغو عضویت در تاپیک را درخواست کند:



```
Pushe.unsubscribe(getApplicationContext(), "desert");
```

غیرفعال کردن و فعال کردن دریافت پوش نوتیفیکیشن پوشه

با استفاده از متد `Pushe.setNotificationOff` دریافت پوش نوتیفیکیشن غیر فعال می‌شود. یک دکمه دیگر با عنوان "عدم دریافت اعلانات" به اکتیویتی اضافه می‌کنم:

```
disableBtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Pushe.setNotificationOff(getApplicationContext());  
    }  
});
```

همچنین کاربری که قبلاً دریافت پوش را غیر فعال کرده مجدداً توسط متد `Pushe.setNotificationOn` می‌تواند دریافت اعلان‌ها را فعال نماید.

در مرحله آخر، سایر تنظیمات قرار داده شده:



۴

تعیین زمان ارسال

ارسال زمان بندی

☐ زمان بندی
 ☒ در اسرع وقت

الویت ارسال ?

متوسط

تنظیمات نمایش

☐ روشن شدن صفحه نمایش
 ☐ صدای اعلان

☐ صفحه نمایش کاربر هنگام دریافت اعلان روشن می شود.

☐ LED

[← بعدی](#)
[ارسال نهایی](#)
[ارسال آزمایشی](#)
[→ قبلی](#)

تنظیماتی شامل زمان ارسال اعلان، کلید جایگزینی، زمان نمایش اعلان، زمان زنده ماندن اعلان و تنظیمات مربوط به نمایش آن.

در قسمت زمان بندی می توانیم تعیین کنیم اعلان دقیقا در چه روز و چه ساعتی ارسال گردد.

بهترین توضیح برای قابلیت کلید جایگزینی، توضیحات خود پوشه است:

"پیام هایی که تا به حال با این کلید ارسال شده باشند، در صورتی که کاربر آنها را ندیده باشد، با این پیام جدید شما جایگزین خواهند شد. با استفاده از این کلید می توانید جلوی ارسال چندباره ی پیام های مشابه به کاربران را بگیرید."



در قسمت زمان نمایش اعلان تعیین می‌کنیم اعلان به محض دریافت روی دیوایس ظاهر شود یا به زمانی که کاربر برنامه را باز می‌کند مોકول گردد.

تنظیمات مربوط به نمایش Push Notification واضح بوده و نیاز به توضیحی ندارد.

تذکر: ارسال پوش نوتیفیکیشن با محتوای تکراری و پشت سرهم قطعا تاثیر منفی زیادی بر اعتماد کاربر به اپ شما می‌گذارد. بنابراین قبل از ارسال نهایی حتما محتوا و تنظیمات اعلان را با دقت بررسی کرده و تک تک آیتم‌ها را از نگاه و دید مخاطب (و نه نگاه خودتان) بسنجید. محتوای جذاب، زمان مناسب ارسال، عدم دریافت اعلان‌های غیر ضرور و... همگی مواردی هستند که عدم رعایت آن تاثیر بدی بر برنامه شما خواهد گذاشت. اگر مخاطبین اعلان فعلی، ساکنین تهران هستند، نیازی به درگیر کردن سایر شهرها و استان‌ها نیست.

توجه: سورس پروژه اندروید درون پوشه Exercises قرار دارد

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش‌های بهتر یاری فرمائید.
این فایل رایگان بوده و انتشار آن (بدون دخل و تصرف) مانعی ندارد.

www.android-studio.ir