Send Notification

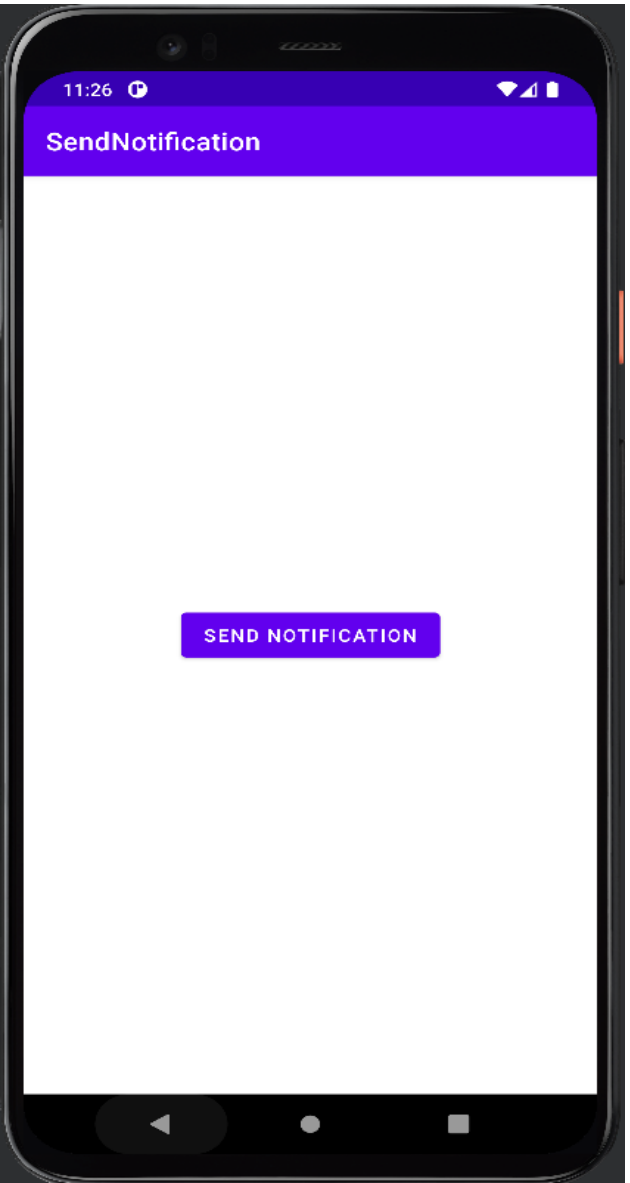
Dr. Charles Yu

- Send a notification to phone user
- Here are the frequently seen use cases
  - For the perspective on the system’s level
    - When we get the system Broadcast intents from the Broadcast Receiver
      - Receiving a phone call
      - Receiving a SMS/MMS
  - For the perspective on end user’s application’s level
    - When something is going wrong
      - Data collection
      - Data computation

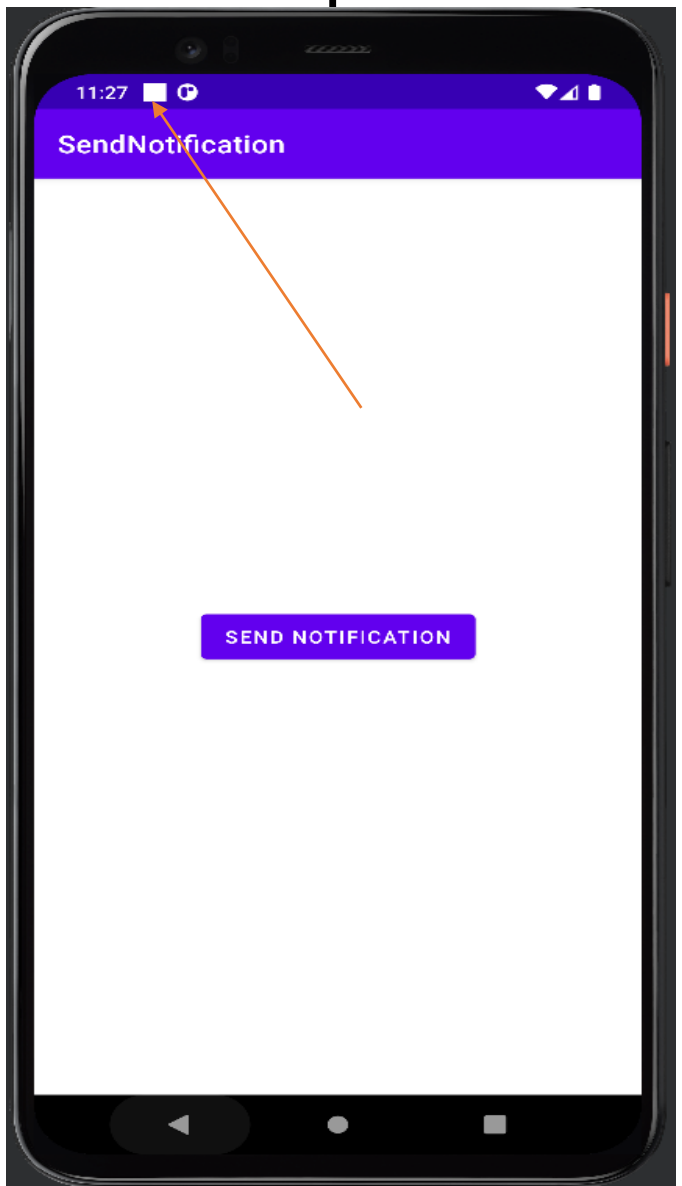
Send a notification to phone user

- [Demo1] SendNotification
  - In this application, we only have one Java file --- MainActivity
    - And one button
    - Click the button and send the Notification
    - Here Is an example (See the next page)

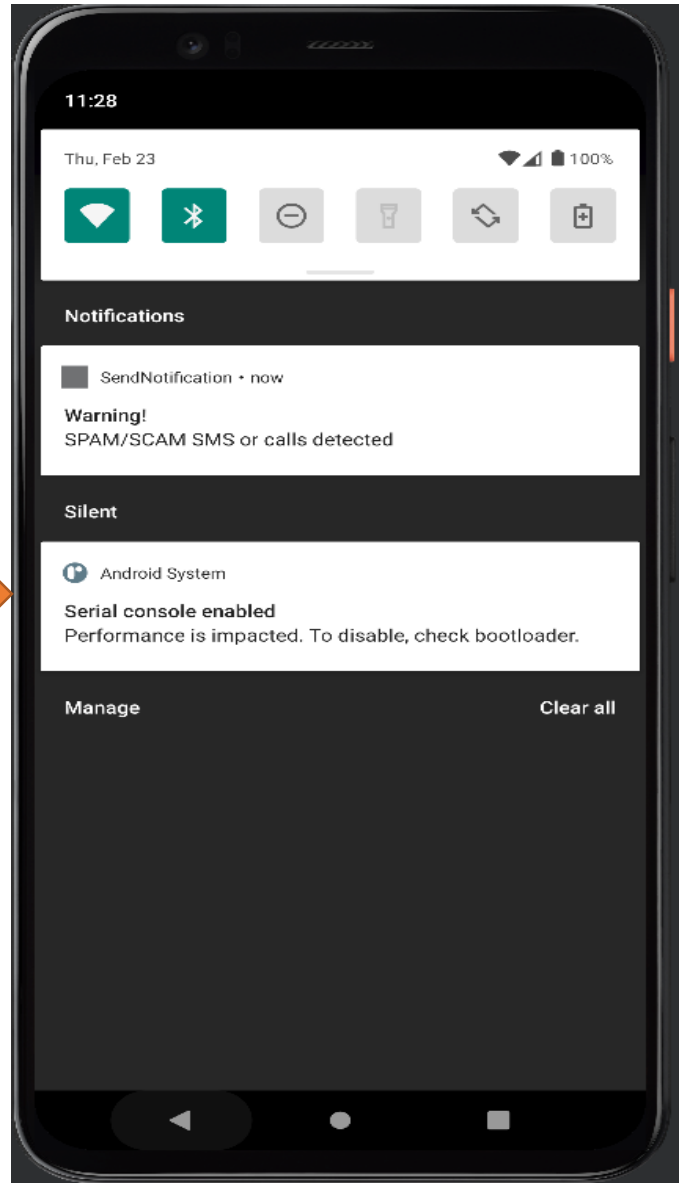
Send a notification to phone user



Click the Button



Swipe it Down



Send a notification to phone user

- Our AndroidMenifest.xml
  - The installation access right
    - This is required
    - POST_NOTIFICATION

activity_main.xml | MainActivity.java | AndroidManifest.xml

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

    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.SendNotification"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Send a notification to phone user

- activity_main.xml
- I comment out the default “TextView” and put a “Button” over there.
- This is the id
- This is the displaying text

activity_main.xml | MainActivity.java | AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!--
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    -->
    <!-- <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/notify_button"
        android:text="Send Notification"
        tools:ignore="MissingConstraints" />
    -->
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/notify_button"
        android:text="Send Notification"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Send a notification to phone user

- We still need to use the request code to send runtime permission
- The Context object is Instantiated
- The button, which we will use it later to send the notification

```java
public class MainActivity extends AppCompatActivity {

    2 usages
    Button notifyBtn;
    4 usages
    Context mContext;

    2 usages
    private static final int  MY_PERMISSIONS_POST_NOTIFICATIONS = 0; // This is the request code

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

        mContext = this.getApplicationContext();

        notifyBtn = findViewById(R.id.notify_button);

        /*
        * This means if the version is newer
        * O: Oreo, API level 26, Android v8.0
        * https://en.wikipedia.org/wiki/Android_version_history
        */

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel( id: "MyNotificationChannelId",
                    name: "MyNotificationChannelName",
                    NotificationManager.IMPORTANCE_DEFAULT);

            NotificationManager manager = getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }
```

Send a notification to phone user

- If the version of Android OS (emulator) is greater than v8.0 (Oreo), we need to setup the notification channel by passing channel id, channel name, and importance level into the constructor

```java
public class MainActivity extends AppCompatActivity {

    2 usages
    Button notifyBtn;
    4 usages
    Context mContext;

    2 usages
    private static final int  MY_PERMISSIONS_POST_NOTIFICATIONS = 0; // This is the request code

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

        mContext = this.getApplicationContext();

        notifyBtn = findViewById(R.id.notify_button);

        /*
        * This means if the version is newer
        * O: Oreo, API level 26, Android v8.0
        * https://en.wikipedia.org/wiki/Android_version_history
        */

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel( id: "MyNotificationChannelId",
                    name: "MyNotificationChannelName",
                    NotificationManager.IMPORTANCE_DEFAULT);

            NotificationManager manager = getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }
```

Send a notification to phone user

- They notification manager is instantiated by calling getSystemService()
- Then, we use the notification manager to create the notification channel by passing the channel we just build.

```java
public class MainActivity extends AppCompatActivity {

    2 usages
    Button notifyBtn;
    4 usages
    Context mContext;

    2 usages
    private static final int  MY_PERMISSIONS_POST_NOTIFICATIONS = 0; // This is the request code

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

        mContext = this.getApplicationContext();

        notifyBtn = findViewById(R.id.notify_button);

        /*
        * This means if the version is newer
        * O: Oreo, API level 26, Android v8.0
        * https://en.wikipedia.org/wiki/Android_version_history
        */

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel( id: "MyNotificationChannelId",
                    name: "MyNotificationChannelName",
                    NotificationManager.IMPORTANCE_DEFAULT);

            NotificationManager manager = getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }
```

Send a notification to phone user

- Now, it’s about the time to deal with the button
- Get the notification builder instance first

```java
notifyBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Notification code goes here

        NotificationCompat.Builder builder = new NotificationCompat.Builder(mContext, channelId: "MyNotificationChannelId");
        builder.setContentTitle("Warning!");
        builder.setContentText("SPAM/SCAM SMS or calls detected");
        builder.setSmallIcon(R.drawable.ic_launcher_background);
        builder.setAutoCancel(true);

        NotificationManagerCompat managerCompat = NotificationManagerCompat.from(mContext);

        if (ActivityCompat.checkSelfPermission(mContext, android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.

            ActivityCompat.requestPermissions( activity: MainActivity.this,
                    new String[]{ android.Manifest.permission.RECEIVE_SMS},
                    MY_PERMISSIONS_POST_NOTIFICATIONS);
        }

        // As for the id, you can pass any unique id.
        // In this way, I'm just saying the notification id starts from zero
        managerCompat.notify( id: 0, builder.build());
    }
});
}
```

Send a notification to phone user

- Setup the notification, its title and text content
- Setup the representing icon and AutoCancel properties

```
notifyBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Notification code goes here

        NotificationCompat.Builder builder = new NotificationCompat.Builder(mContext, channelId: "MyNotificationChannelId");
        builder.setContentTitle("Warning!");
        builder.setContentText("SPAM/SCAM SMS or calls detected");
        builder.setSmallIcon(R.drawable.ic_launcher_background);
        builder.setAutoCancel(true);

        NotificationManagerCompat managerCompat = NotificationManagerCompat.from(mContext);

        if (ActivityCompat.checkSelfPermission(mContext, android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.

            ActivityCompat.requestPermissions( activity: MainActivity.this,
                    new String[]{ android.Manifest.permission.RECEIVE_SMS},
                    MY_PERMISSIONS_POST_NOTIFICATIONS);
        }

        // As for the id, you can pass any unique id.
        // In this way, I'm just saying the notification id starts from zero
        managerCompat.notify( id: 0, builder.build());
    }
});
}
```

Send a notification to phone user

- Get the NotificationManagerCompat Object
- We are going to use this object to send out the notification
- We still need to ask for **run-time permissions** from user
- Check the slide #5, it is the **only one permission** we need

```java
notifyBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Notification code goes here

        NotificationCompat.Builder builder = new NotificationCompat.Builder(mContext, channelId: "MyNotificationChannelId");
        builder.setContentTitle("Warning!");
        builder.setContentText("SPAM/SCAM SMS or calls detected");
        builder.setSmallIcon(R.drawable.ic_launcher_background);
        builder.setAutoCancel(true);

        NotificationManagerCompat managerCompat = NotificationManagerCompat.from(mContext);

        if (ActivityCompat.checkSelfPermission(mContext, android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.

            ActivityCompat.requestPermissions( activity: MainActivity.this,
                    new String[]{ android.Manifest.permission.RECEIVE_SMS},
                    MY_PERMISSIONS_POST_NOTIFICATIONS);
        }

        // As for the id, you can pass any unique id.
        // In this way, I'm just saying the notification id starts from zero
        managerCompat.notify( id: 0, builder.build());
    }
});
}
```

Send a notification to phone user

- There is a thing. (weird) **If I don't put this "check the permission"**, I will get a compilation error.
- However**, if I put this, it doesn't show the run-time requesting dialog**, like we have seen earlier

```java
notifyBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Notification code goes here

        NotificationCompat.Builder builder = new NotificationCompat.Builder(mContext, channelId: "MyNotificationChannelId");
        builder.setContentTitle("Warning!");
        builder.setContentText("SPAM/SCAM SMS or calls detected");
        builder.setSmallIcon(R.drawable.ic_launcher_background);
        builder.setAutoCancel(true);

        NotificationManagerCompat managerCompat = NotificationManagerCompat.from(mContext);

        if (ActivityCompat.checkSelfPermission(mContext, android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.

            ActivityCompat.requestPermissions( activity: MainActivity.this,
                    new String[]{ android.Manifest.permission.RECEIVE_SMS},
                    MY_PERMISSIONS_POST_NOTIFICATIONS);
        }

        // As for the id, you can pass any unique id.
        // In this way, I'm just saying the notification id starts from zero
        managerCompat.notify( id: 0, builder.build());
    }
});
}
```

Send a notification to phone user

- The last piece in MainActivity. We put a call-back function to handle the permission status
- However, it seems that **it is NEVER get executed**! **Because I cannot see the Toast at all!**

```java
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        switch (requestCode) {
            case MY_PERMISSIONS_POST_NOTIFICATIONS:
                if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    Toast.makeText( context: this,  text: "Permission Granted", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText( context: this,  text: "Permission Not Granted!", Toast.LENGTH_LONG).show();
                    this.finish();
                }
        }
    }
}
```

Send a notification to phone user

- In conclusion. It seems like we only need to check the access-rights to **send the notification.** And, this is a **MUST.**
- However, we don’t really need to request run-time permission “**officially**” for sending the notification
- It seems like the access rights for “sending the notification” **is not the security critical jobs?**