

## פרק 23 - ContentProvider

### מה הוא ContentProvider?

הוא רכיב המשמש להעברת מידע בין אפליקציה ContentProvider אחת לשנייה.

### מתי נשתמש ב ContentProvider?

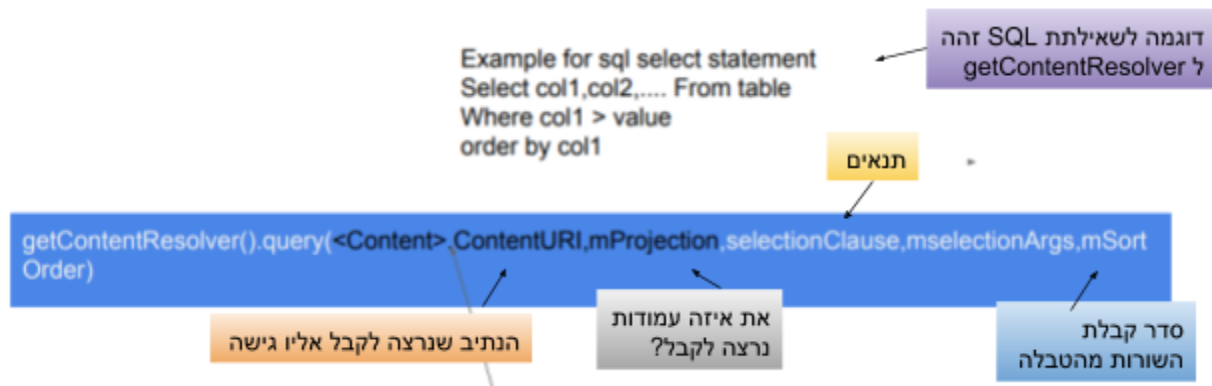
במידה ונרצה שהאפליקציה שלנו תקבל מידע מאפליקציה אחרת. דוגמה: שליפת אנשי הקשר מהאפליקציה שבטלפון ולאחר מכן הצגתם באפליקציה שלנו.

כשנרצה לקבל מידע מאפליקציית אנשי הקשר נשלח בקשה ( request ) באמצעות ContentProvider , ובחזרה נקבל תשובה ( response ) מהאפליקציה באמצעות ContentResolver . התשובה תהיה מצביע ( cursor ) ל database עם נתונים.

אם נרצה שאפליקציות אחרות יוכלו לקבל מהאפליקציה שלנו מידע נוכל לבנות גם Content Resolver לאפליקציה שלנו, אך לא נלמד זאת בקורס שלנו.

בשביל ליצור את ה query ה ContentResolver יקבל חמישה פרמטרים:

## Content Provider



סיכום:

:ContentProvider

- 1.content uri - הנתיב שנרצה לקבל אליו גישה
- 2.insert()
- 3.update()
- 4.delete()
- 5.query()

ספקי התוכן מאפשרים שיתוף נתונים בין אפליקציות  
 ספקי התוכן מספקים מנגנונים להבטחת המידע

ספקי התוכן נותנים ממשק שמחבר תהליך אחד עם קוד מתהליך  
אחר

contentResolver משמש כ - client לספקי התוכן. כאשר נרצה  
לגשת לנתונים של ספק תוכן אז נשתמש בו כדי לתקשר עם ספק  
התוכן.

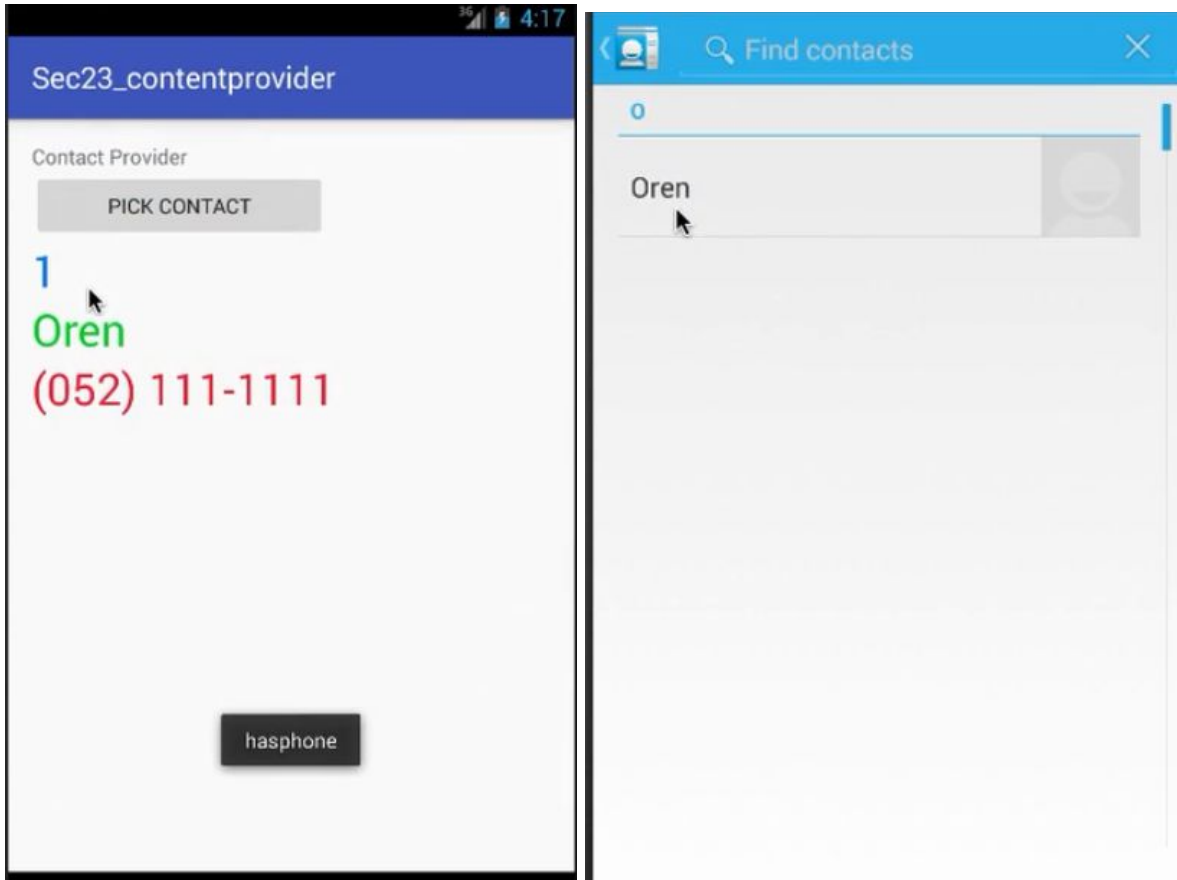
ספק התוכן יודע לקבל בקשות מה- client ומחזיר תשובה או תוצאה.

במידה ואתם מעוניינים שהאפליקציה שלכם תשתף נתונים עם  
אפליקציות אחרות אזי נפתח ספק תוכן.

באנדרואיד ישנם ספקי תוכן כגון אודיו, תמונות, וידאו, לוח שנה ועוד.

דוגמה לשימוש ב ContentProvider:

שליפת אנשי הקשר מאפליקציית אנשי הקשר בטלפון ולאחר מכן הצגתם באפליקציה שלנו.



שימו לב! בשביל להריץ את האפליקציה על האימולטור בתור התחלה נוסיף אנשי קשר לאימולטור.

שלב 1 - קובץ המניפסט

בתור התחלה נוסיף למניפסט הרשאה לקרוא את אנשי הקשר:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="iil.co.appschool.sec23_contentprovider">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/contentprovider"
        android:label="Sec23_contentprovider"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <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>
```

## שלב 2 - עיצוב MainActivity:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"
    android:orientation="vertical"
    tools:context="iil.co.appschool.sec23_contentprovider.MainActivity">

    <TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Contact Provider" />
```

```
<Button
```

```
android:layout_width="200dp"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/btnShowAll"
```

```
android:text="Pick Contact"
```

```
/>
```

```
<TextView
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/tvId"
```

```
android:text=""
```

```
android:textSize="30sp"
```

```
android:textColor="#0775e4"
```

```
/>
```

```
<TextView
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/tvName"
```

```
android:text=""
```

```
android:textSize="30sp"
```

```
android:textColor="#07e421"
```

```
/>
```

```
<TextView
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/tvPhone"
```

```
android:text=""
```

```
android:textSize="30sp"
```

```
android:textColor="#e40733"
```

```
/>
```

שלב 3 - עריכת MainActivity:

```
public class MainActivity extends AppCompatActivity {
```

```
    Button btnShowAll;
```

הצהרות

```
    TextView tvName, tvPhone, tvId;
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

ניתן הפניה ל btnShowAll ונאזין לו

```
    btnShowAll = (Button) findViewById(R.id.btnShowAll);
```

```
    btnShowAll.setOnClickListener(new View.OnClickListener() {
```

```
@Override
```

```
public void onClick(View v) {
```

בהמשך נממש את showContactsPicker

```
    showContactsPicker();
```

```
}
```

```
});
```

```
    tvName=(TextView)findViewById(R.id.tvName);
```

ניתן הפניות לכל האובייקטים שלנו

```
    tvId=(TextView)findViewById(R.id.tvId);
```

```
    tvPhone=(TextView)findViewById(R.id.tvPhone);
```

```
}
```

```
public void showContactsPicker() {
```

```
    Intent intent = new Intent(Intent.ACTION_PICK, ContactsContract.Contacts.CONTENT_URI);
```

```
    startActivityForResult(intent, 0);
```

נעבור לאפליקציית אנשי הקשר עם requestCode 0

```
}
```



@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
```

בחזרה מאנשי הקשר

```
    super.onActivityResult(requestCode, resultCode, data);
```

```
    if (requestCode == 0) {
```

```
        if (resultCode == RESULT_OK) {
```

```
            Cursor cursor;
```

נצהיר על מצביע

נבדוק מה גרסת ה SDK, כיוון שהקוד יהיה שונה עבור גרסאות SDK מסוימות

```
            if (android.os.Build.VERSION.SDK_INT < 11) {
```

```
                cursor = managedQuery(data.getData(), null, null, null, null);
```

הנתונים שבחרנו באנשי קשר חוזרים ב data של האינטנט

```
            } else {
```

```
                ///---Honeycomb and later use the cursor loader class to retrieve managed cursor---
```

```
                CursorLoader cursorLoader = new CursorLoader(
```

```
                    this,
```

```
                    data.getData(),
```

```
                    null,
```

```
                    null,
```

```
                    null,
```

```
                    null);
```

נעשה אותו הדבר אך בגרסת קוד שונה עבור גרסאות SDK אחרות

```
                cursor = cursorLoader.loadInBackground();
```

```
            }
```

```

String name = cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
String id = cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID));
String hasPhone = cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER));

tvId.setText(id);
tvName.setText(name);

if (hasPhone.equalsIgnoreCase("1")){//check if contact has phone
    String phoneNum = "";

    Cursor phones = getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
        ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " = " + id, null, null);

    if (phones.moveToFirst())
        phoneNum=phones.getString(phones.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));

    phones.close();
    cursor.close();

    tvPhone.setText(phoneNum);
    Toast.makeText(this, "hasphone", Toast.LENGTH_LONG).show();
}
else tvPhone.setText("This record doesn't have phone number");
}
}
}
}

```

נשלוח את השם וה id של איש הקשר שאנחנו מצביעים אליו

נשנה את ה TextView בהתאם לאיש הקשר

1 אומר שיש לאיש הקשר טלפון. 0 אומר שאין

נתיב לתיקית אנשי הקשר

תנאי: נקבל את איש הקשר שה id שלו שווה לזה ששלפנו

נגיד למצביע להצביע לאיבר הראשון שעונה על התנאי

נשלוף את מספר הטלפון של איש הקשר

נסגור את הנתונים שפתחנו

תוכלו להריץ את האפליקציה:

