

Mobil megoldások

Kocsis Gergely
2019.09.30.

Többablakos alkalmazás

Ahhoz, hogy egy alkalmazás több ablakból állhasson, egyszerűen több Activity-t kell készíteni.

Az Activity-k közötti átjárást az Intent-ek segítik:

“An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed.”



Shoppinglist alkalmazás

A főablakban csak egy lista található és egy “Add” gomb.

A gomb megnyomására megjelenik egy új ablak, ahonnan előre beírt tételek közül lehet kiválasztani, hogy mit adunk hozzá a listához.

Első körben készítsünk olyan alkalmazást, mely egy gomb megnyomására megnyit egy másik Activity-t.



Shoppinglist alkalmazás

A főablakban csak egy lista található és egy “Add” gomb.
A gomb megnyomására megjelenik egy új ablak, ahonnan előre beírt tételek közül lehet kiválasztani, hogy mit adunk hozzá a listához.

Első körben készítsünk olyan alkalmazást, mely egy gomb megnyomására megnyit egy másik Activity-t.

Egy új Activity indítása:

```
Intent intent = new Intent(this, ItemsActivity.class);  
startActivity(intent);
```



Shoppinglist alkalmazás

A főablakban csak egy lista található és egy “Add” gomb.
A gomb megnyomására megjelenik egy új ablak, ahonnan előre beírt tételek közül lehet kiválasztani, hogy mit adunk hozzá a listához.

Első körben készítsünk olyan alkalmazást, mely egy gomb megnyomására megnyit egy másik Activity-t.

Egy új Activity indítása:

```
Intent intent = new Intent(this, ItemsActivity.class);  
startActivity(intent);
```

Szülő-gyermek kapcsolat beállítása az AndroidManifest.xml-ben:

```
<activity android:name=".ItemsActivity"  
    android:parentActivityName=".MainActivity"></activity>
```



Shoppinglist alkalmazás

Most módosítsuk úgy a kódot, hogy a második Activity-n szereplő gombok felirata hozzáadódjon a listánkhoz.

Ehhez egyrészt az indításkor jeleznünk kell, hogy az új Activity-t valamilyen céllal indítjuk.

```
public void addButtonClicked(View view) {  
    Intent intent = new Intent(this, ItemsActivity.class);  
    startActivityForResult(intent, 1);  
}
```

Request Code: Ezzel tudjuk majd beazonosítani a válsz érkezésekor, hogy pontosan mire is kaptuk azt



Shoppinglist alkalmazás

Most módosítsuk úgy a kódot, hogy a második Activity-n szereplő gombok felirata hozzáadódjon a listánkhoz.

Ehhez egyrészt az indításkor jeleznünk kell, hogy az új Activity-t valamilyen céllal indítjuk.

```
public void addButtonClicked(View view) {  
    Intent intent = new Intent(this, ItemsActivity.class);  
    startActivityForResult(intent, 1);  
}
```

Másrészt az indított Activity-ből vissza kell adnunk egy Intentet

```
public void addItem(View view) {  
    Intent replyIntent = new Intent();  
    setResult(RESULT_OK, replyIntent);  
    finish();  
}
```

Request Code: Ezzel tudjuk majd beazonosítani a válsz érkezésekor, hogy pontosan mire is kaptuk azt



Shoppinglist alkalmazás

Adatokat küldeni az Intent "Extra" tulajdonságában tudunk név érték párok megadásával.

Például a lenti kódban az "ITEM" névhez a gomb címkejét rendeljük értéként

```
public void addItem(View view) {  
    Button btn = (Button)view;  
    Intent replyIntent = new Intent();  
    replyIntent.putExtra("ITEM", btn.getText().toString());  
    setResult(RESULT_OK, replyIntent);  
    finish();  
}
```



Shoppinglist alkalmazás

Az adatok kinyeréséhez felül kell definiálni az onActivityResult metódust:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode==1){
        if (resultCode==RESULT_OK){
            if (listIsEmpty) {listTextView.setText("");listIsEmpty=false;}
            listTextView.append("\n" +data.getStringExtra("ITEM"));
        }
    }
}
```



Shoppinglist alkalmazás

Az adatok kinyeréséhez felül kell definiálni az onActivityResult metódust:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode==1){
        if (resultCode==RESULT_OK){
            if (listIsEmpty) {listTextView.setText("");listIsEmpty=false;}
            listTextView.append("\n" +data.getStringExtra("ITEM"));
        }
    }
}
```

Vesd össze:

```
startActivityForResult(intent, 1);
```



Shoppinglist alkalmazás

Az adatok kinyeréséhez felül kell definiálni az onActivityResult metódust:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode==1){
        if (resultCode==RESULT_OK) {
            if (listIsEmpty) {listTextView.setText("");listIsEmpty=false;}
            listTextView.append("\n" +data.getStringExtra("ITEM"));
        }
    }
}
```

Vesd össze:

```
startActivityForResult(intent, 1);
```

Vesd össze:

```
replyIntent.putExtra("ITEM", btn.getText().toString());
```



Shoppinglist alkalmazás

Az alkalmazás ezzel kész, viszont lehattunk volna elegánsabbak is.

- Emeljük az Intentben átvitt név-érték pár nevét nevesített konstanssá, hogy ne lehessen elgépelni
- Használjunk String erőforrásokat statikusan beégetett Stringek helyett (ALT+ENTER → Extract String resource)
- Adjunk felhasználó rákeressen google-ben a termékre

```
if (requestCode==2) {  
    if (resultCode==RESULT_OK) {  
        Intent intent = new Intent(Intent.ACTION_VIEW,  
            Uri.parse("https://www.google.com/search?q=" + data.getStringExtra("ITEM")));  
        if (intent.resolveActivity(getPackageManager()) != null) {  
            startActivity(intent);  
        }  
    }  
}
```

- Bővítsük az alkalmazást úgy, hogy lehetőség legyen darabszámot megadni.
- További fejlesztés lehet, hogy termékcsoportokat definiálunk, amiből választani lehet.

