

안드로이드 코틀린으로 만든 리사이클러뷰(RecyclerView)에 클릭 기능 추가하기 (Add ClickListener) 스터디/Android+Kotlin

❖ 지난 게시글에 이어지는 게시글입니다.

<https://dalgonakit.tistory.com/138>

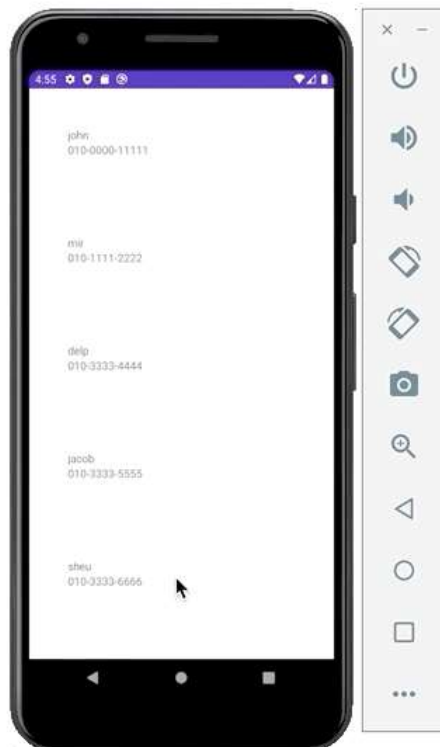
안드로이드 코틀린으로 리스트 만들기 (RecyclerView)

dalgonakit.tistory.com

1.

2. 안녕하세요 !

오늘은 지난번 만든 리스트 앱에 클릭 기능을 추가해보도록 하겠습니다.



지난 앱에서는 클릭 기능을 추가하지 않아서 방향하는 마우스를 볼 수 있습니다 ㅠㅠ

2.1.1. 참고.기존 자료

지난 게시글에서 제작한 파일들 입니다.

참고해주세요!

2.1.2. Contacts.kt

```
KOTLIN package com.example.mylisapplication
```

```
class Contacts(var name: String, var tel: String) {  
}
```

2.1.3. ContactsListAdapter.kt

```
KOTLIN package com.example.mylisapplication
```

```
import android.view.LayoutInflater
```

```
import android.view.View
```

```
import android.view.ViewGroup
```

```
import androidx.recyclerview.widget.RecyclerView
```

```
class ContactsListAdapter(private val itemList : List<Contacts>) : RecyclerView.Adapter<ContactsViewHolder>() {  
  
    override fun getItemCount() : Int {  
        return itemList.size  
    }  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ContactsViewHolder {  
        val inflatedView = LayoutInflater.from(parent.context).inflate(R.layout.item_contacts, parent, false)  
        return ContactsViewHolder(inflatedView)  
    }  
  
    override fun onBindViewHolder(holder: ContactsViewHolder, position: Int) {  
        val item = itemList[position]  
        holder.apply {  
            bind(item)  
        }  
    }  
}
```

2.1.4. ContactsViewHolder.kt

```
KOTLIN package com.example.mylisapplication
```

```
import android.view.View
```

```

import androidx.recyclerview.widget.RecyclerView

import kotlinx.android.synthetic.main.item_contacts.view.*

class ContactsViewHolder(v: View) : RecyclerView.ViewHolder(v) {

    var view : View = v

    fun bind(item: Contacts) {

        view mName.text = item.name

        view mTel.text = item.tel

    }
}

```

2.1.5. ListActivity.kt

```

KOTLIN
package com.example.mylistapplication

```

```

import android.app.Activity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_layout.*

class ListActivity : Activity() {

    val contactsList : List<Contacts> = listOf(

        Contacts("john", "010-0000-1111"),
        Contacts("mir", "010-1111-2222"),
        Contacts("delp", "010-3333-4444"),
        Contacts("jacob", "010-3333-5555"),
        Contacts("sheu", "010-3333-6666"),
        Contacts("ma", "010-3333-7777"),
        Contacts("ham", "010-3333-8889")

    )

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_layout)

        val adapter = ContactsListAdapter(contactsList)

        mRecyclerView.adapter = adapter

    }
}

```

3. 자! 본격적으로 클릭을 달아보겠습니다!

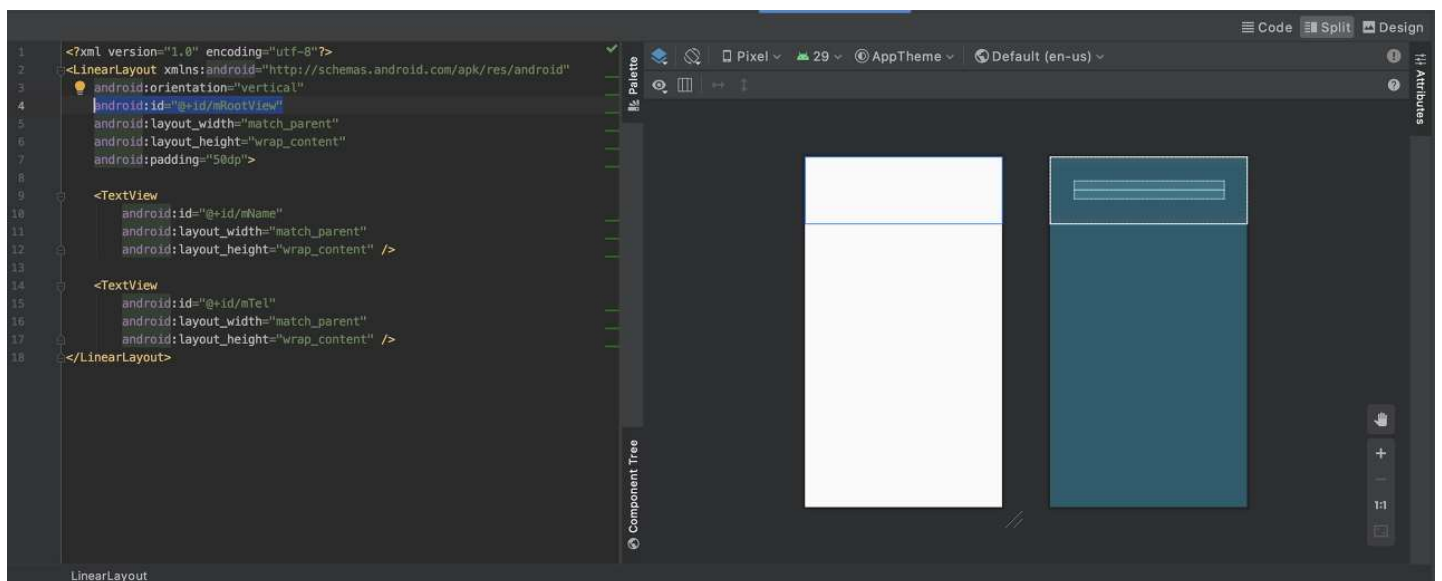
각 칸을 클릭을 하게되면, 칸에 들어간 정보를 Toast메세지를 이용해서 화면에 출력해보겠습니다.

거기다가 이름 뒤에 '1'을 점점 추가해 보겠습니다.

4. XML 수정

클릭리스너를 추가하기 전에, 리스너가 연결될 영역에 id를 추가해야합니다

최상위에 있는 LinearLayout태그에 android:id="@+id/mRootView"를 추가합니다



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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:id="@+id/mRootView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="50dp">
```

```
<TextView
    android:id="@+id/mName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```

<TextView
    android:id="@+id/mTel"

    android:layout_width="match_parent"

    android:layout_height="wrap_content" />
</LinearLayout>

```

5. 클릭 리스너 추가

클릭 리스너를 추가하는 방법은 정말 많습니다.

그 중 방법 3가지로 추려 보았습니다.

1. viewHolder에 추가하는 방법
2. Adapter로 전달하는 방법
3. 3.Adapter에 클릭리스너 추가하는 방법

5.1.1. 1. ViewHolder에 추가하는 방법

ViewHolder에 setOnClickListener를 호출해서 Toast 메시지를 띄웠습니다.

```

package com.example.mylisapplication

```

```

import android.view.View

```

```

import android.widget.Toast

```

```

import androidx.recyclerview.widget.RecyclerView

```

```

import kotlinx.android.synthetic.main.item_contacts.view.*

```

```

class ContactsViewHolder(v: View) : RecyclerView.ViewHolder(v) {

    var view : View = v

    fun bind(item: Contacts) {

        view.mName.text = item.name

        view.mTel.text = item.tel

        view.mRootView.setOnClickListener {

            Toast.makeText(view.context, "${item.name}\n${item.tel}", Toast.LENGTH_SHORT).show()

        }
    }
}

```

```
}  
}
```

ContactsViewHolder.kt

이 방법은 간단한 작업은 좋지만, ViewHolder에서 구현했기 때문에,
bind 함수에서 item에 접근할 수는 있지만, RecyclerView 또는 Adapter, 인스턴스 등에 접근할 수 없어서
ViewHolder안에 클릭리스너를 구현한 상태로 복잡한 기능을 구현하려고 하면 할 수록 ViewHolder가 아닌 다른 괴물이 되어버립니다.

그리고 이 방법은 '뷰가 재사용되는 RecyclerView'를 재사용을 안 사용하는 것처럼 만들어 줍니다.
Adapter를 이용해서 초기 생성시 화면에 보여줄 최소한의 ViewHolder만 생성하고,
생성된 ViewHolder를 재활용하여 화면에 출력된 부분에만 onBindViewHolder함수를 호출하는데 그때
ViewHolder의 bind() 함수가 호출되는 것이다보니, 매번 ViewHolder가 화면에 보일때 마다 setOnClickListener가 호출되는것이니 엄청 비효율적일 수 밖에
없습니다.

한 번 뷰가 만들어지면 ClickListener를 한 번 할당하는 것이 이상적 입니다.

5.1.2. 2. Adapter로 전달하는 방법

Adapter에서 bind할때 setOnClickListener를 이용해서 clickListener를 전달합니다

```
package com.example.mylisapplication
```

```
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Toast  
import androidx.recyclerview.widget.RecyclerView
```

```
class ContactsListAdapter(private val itemList : List<Contacts>) : RecyclerView.Adapter<ContactsViewHolder>() {  
  
    override fun getItemCount(): Int {  
        return itemList.size  
    }  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ContactsViewHolder {  
        val inflatedView = LayoutInflater.from(parent.context).inflate(R.layout.item_contacts, parent, false)  
        return ContactsViewHolder(inflatedView)  
    }  
  
    override fun onBindViewHolder(holder: ContactsViewHolder, position: Int) {  
        val item = itemList[position]  

```

```

        val adapter = this

        holder.apply {
            bind(item, View.OnClickListener{
                Toast.makeText(view.context, "Adapter\n${item.name}\n${item.tel}", Toast.LENGTH_SHORT).show()

                item.name = item.name + "1"

                adapter.notifyDataSetChanged()
            })
        }
    }
}

```

ContactsListAdapter.kt

```

KOTLIN
package com.example.mylisapplication

```

```

import android.view.View
import androidx.recyclerview.widget.RecyclerView
import kotlinx.android.synthetic.main.item_contacts.view.*

```

```

class ContactsViewHolder(v: View) : RecyclerView.ViewHolder(v) {

    var view : View = v

    fun bind(item: Contacts, onClickListener: View.OnClickListener) {
        view.mName.text = item.name
        view.mTel.text = item.tel
        view.mRootView.setOnClickListener(onClickListener)
    }
}

```

ContactsViewHolder.kt

ClickListener 구현만 Adapter에서 했다 뿐이지, 첫번째 방법인 ViewHolder에서 구현하는 방법과 별다를게 없지만 Adapter에 접근할 수 있는 장점이 있습니다.

위 예제에서는 클릭할때마다 item의 이름에 "1"을 추가하고, adapter.notifyDataSetChanged() 함수를 호출해서 리스트를 갱신할 수 있습니다.

5.1.3. 3.Adapter에 클릭리스너 추가하는 방법

Adapter에 ClickListener를 구현하고, onBindViewHolder 부분에 holder.itemView에 클릭리스너를 연결합니다

```

KOTLIN
override fun onBindViewHolder(holder: ContactsViewHolder, position: Int) {
    val item = itemList[position]

```

```

        holder.itemView.setOnClickListener {
            itemClickListener.onClick(it, position)
        }
        holder.apply {
            bind(item)
        }
    }
}

//ClickListener
interface OnItemClickListener {
    fun onClick(v: View, position: Int)
}

private lateinit var itemClickListener : OnItemClickListener

fun setItemClickListener(itemClickListener: OnItemClickListener) {
    this.itemClickListener = itemClickListener
}

```

ListActivity에선 Adapter에 추가된 setItemClickListener 함수를 이용해서 클릭리스너를 구현하면 됩니다

```

package com.example.mylistapplication

import android.app.Activity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_layout.*

class ListActivity : Activity() {

    val contactsList : List<Contacts> = listOf(
        Contacts("john", "010-0000-1111"),
        Contacts("mir", "010-1111-2222"),
        Contacts("delp", "010-3333-4444"),
        Contacts("jacob", "010-3333-5555"),
        Contacts("sheu", "010-3333-6666"),
        Contacts("ma", "010-3333-7777"),
        Contacts("ham", "010-3333-8889")
    )

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_layout)
    }
}

```



```

val adapter = ContactsListAdapter(contactsList)
adapter.setItemClickListener(object : ContactsListAdapter.OnItemClickListener{
    override fun onClick(v: View, position: Int) {
        val item = contactsList[position]

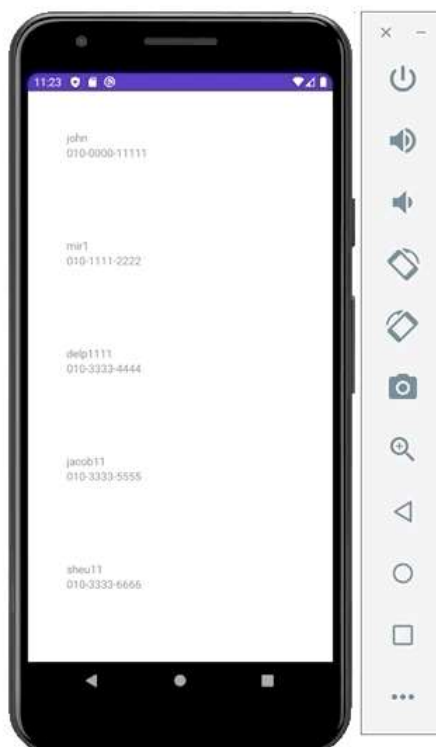
        Toast.makeText(v.context, "Activity\n${item.name}\n${item.tel}", Toast.LENGTH_SHORT).show()
        item.name = item.name + "1"
        adapter.notifyDataSetChanged()
    }
})
mRecyclerView.adapter = adapter
}
}

```

6. 동작모습

3번을 이용해서 완성된 모습입니다.

리스트의 칸을 클릭하면, 토스트 메시지가 출력되면서 이름 뒤에는 '1'이 추가됩니다.



[Adapter에 클릭리스너 추가하기] 실행한 모습

7. 프로젝트 파일

MvListApplication.zip
0.42MB

8. 마무리

RecyclerView에는 많은 요소들이 있는 만큼 더 다양한 구현방법이 있지만,
일반적으로 사용하는 방법에 대해서 소개해드렸습니다

다음은 아이템 갯수를 가변적으로 만드는 방법에 대해서 소개해 드리겠습니다
또 봐요~