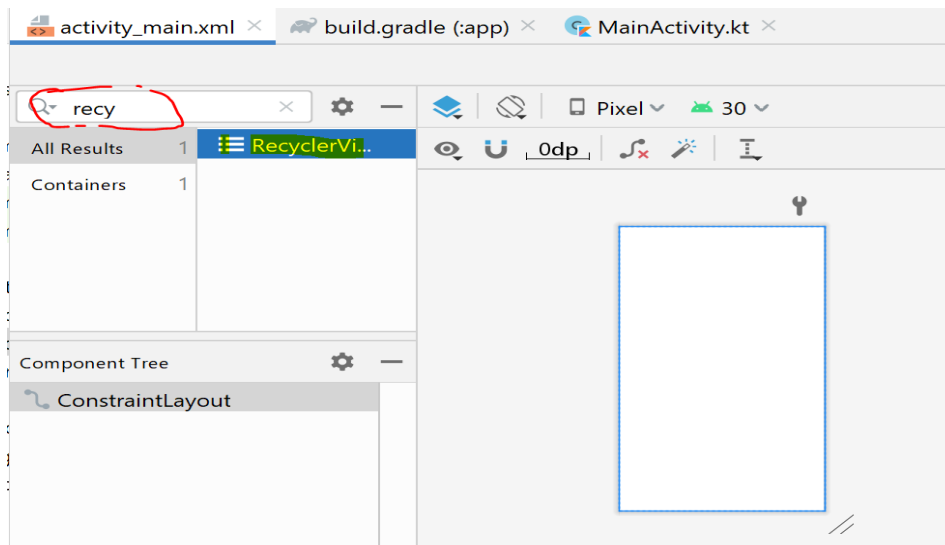


## RecyclerView Implementation for the Book class – Step by Step Manual

Add under plugins build.gradle

```
id 'kotlin-android-extensions'
```

**Step 1:** Create a new Project and open activity\_main.xml, remove the TextView. Go to your XML Design editor and drag RecyclerView by searching.



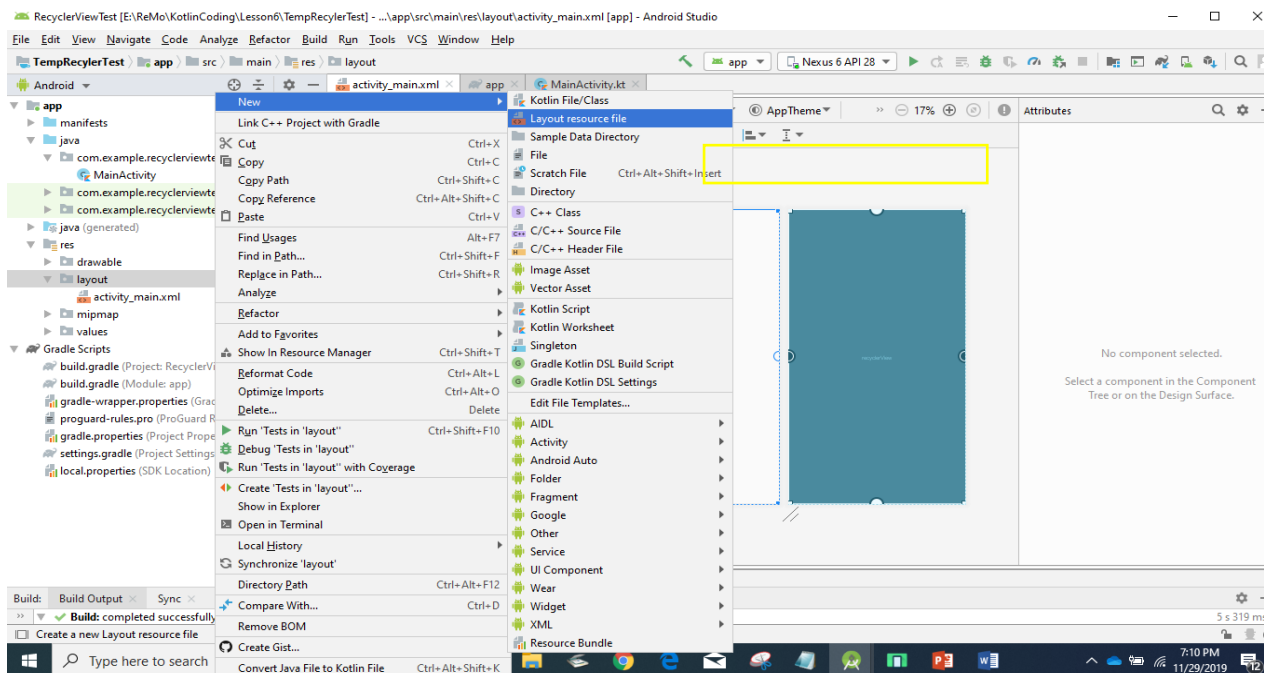
**Step 2:** Set constraints for all four edges, then you activity\_main.xml has the below code.

```
<?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">
```

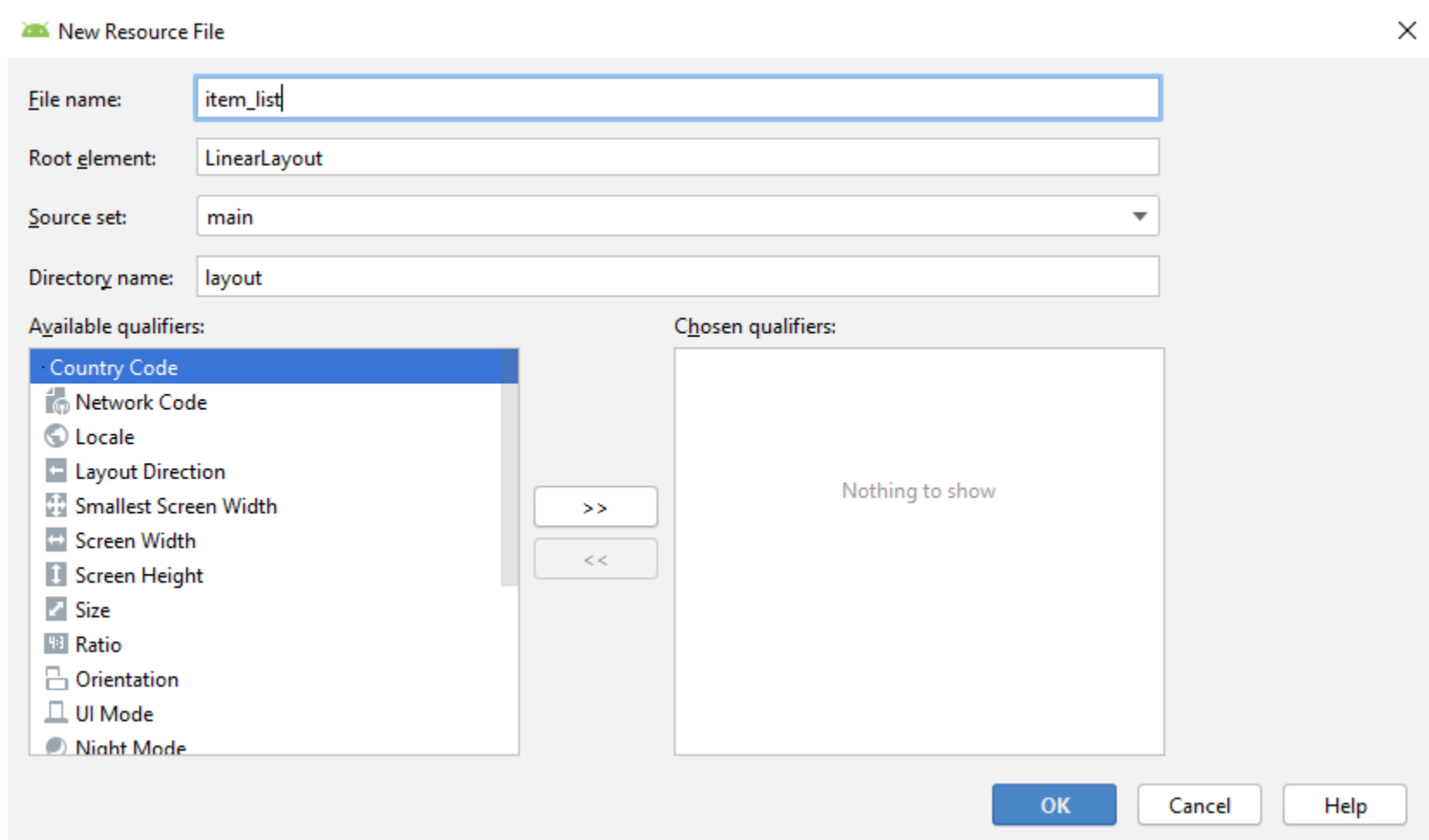
```
tools:context=".MainActivity">
```

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerView1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

**Step 3:** Create Layout Resource File according to show the view of single item in the list. So, create a new Layout resource file as specified below



**Step 4:** Provide the name of your Layout as item\_list



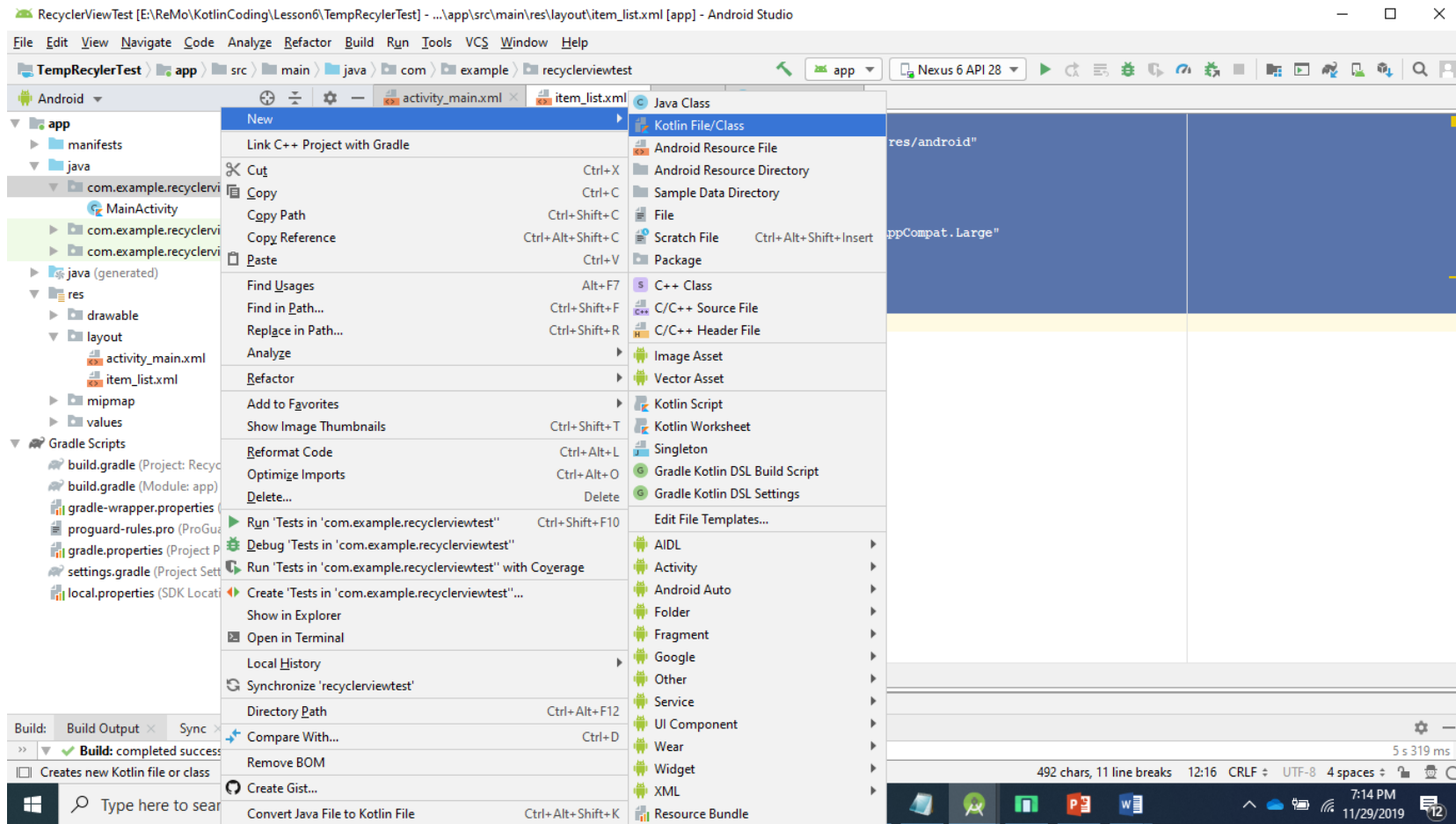
**Step 5:** Type the following code in your item\_list.xml. Only one TextView is used for this example. You can customize according to your needs.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp">
    <TextView
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"
        android:id="@+id/name"
        android:text="Book Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"
        android:text="Author"
        android:id="@+id/author"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

**Step 6:** Create a Data Class Book

```
data class Book(val name: String, val author: String)
```

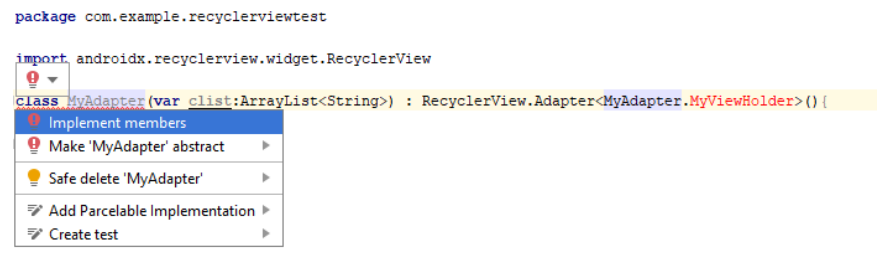
**Step 7:** Create Kotlin class as per the screenshot below and name it as MyAdapter.



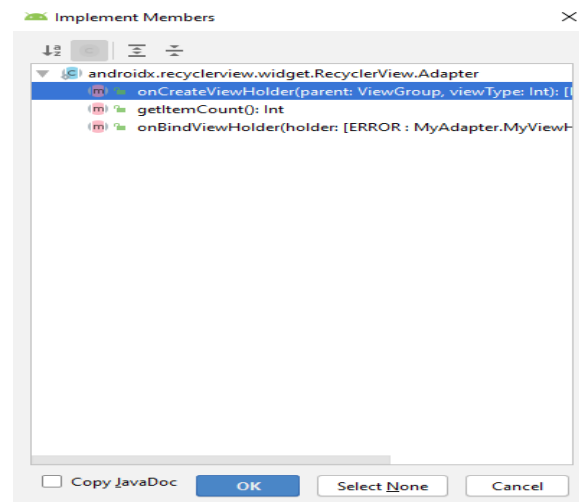
**Step 8:** Create a Class as Mentioned below. Your class should have a constructor to accept the input list and should inherit from RecyclerView.Adapter<Type>. The type be your own inner class Which is here MyViewHolder from MyAdapter class. MyViewHolder inherit from RecyclerView.ViewHolder(itemView)

```
class MyAdapter(var blist:ArrayList<Book>) : RecyclerView.Adapter<MyAdapter.MyViewHolder>(){  
}
```

**Step 9:** You will get errors as below



**Step 10:** Click the Red Error, you will get the methods needs to override from RecyclerView.Adapter, select all the methods as per the screen shot and click OK.



**Step 11:** Your code will look like this, **Remove the line:** `TODO("Not yet implemented")`, from each method. You will get error on `MyAdapter.MyViewHolder`, red color spot.

```
class MyAdapter(var blist: ArrayList<Book>) : RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyAdapter.MyViewHolder {
        TODO("Not yet implemented")
    }

    override fun onBindViewHolder(holder: MyAdapter.MyViewHolder, position: Int) {
        TODO("Not yet implemented")
    }

    override fun getItemCount(): Int {
        TODO("Not yet implemented")
    }
}
```

**Step 12:** Create an inner class `MyAdapter` as mentioned below.

```
class MyAdapter(var blist: ArrayList<Book>) : RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyAdapter.MyViewHolder {

    }

    override fun onBindViewHolder(holder: MyAdapter.MyViewHolder, position: Int) {

    }

    override fun getItemCount(): Int {

    }
    inner class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)
}
```

**Step 13:** Implement all the methods, as mentioned below

```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import kotlinx.android.synthetic.main.item_list.view.*

// Inflate the Layout to set in the RecyclerView and return the ViewHolder object
class MyAdapter(var blist: ArrayList<Book>) : RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyAdapter.MyViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.item_list, parent, false)
        return MyViewHolder(view)
    }
    /* Sets the contents of an item at a given position in the RecyclerView.
    Called by RecyclerView to display the data at a specified position over and over.
    */
    override fun onBindViewHolder(holder: MyAdapter.MyViewHolder, position: Int) {
        holder.itemView.name.text = blist[position].name
        holder.itemView.author.text = blist[position].author
    }
    // return Size of the list of data.

    override fun getItemCount(): Int {
        return blist.size
    }

    /*RecyclerView.Adapter accepts the generic type of your Adapter inner class ViewHolder type.
    In this example Adapter class name is MyAdapter and the MyViewHolder is the inner class */

    inner class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)
}
```



**Step 14:** Do the implementation in the MainActivity.kt

MainActivity.kt

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.LinearLayoutManager
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val books = ArrayList<Book>()
        books.add(Book("Java", "Horstman"))
        books.add(Book("Kotlin", "Joshua Bloch"))
        books.add(Book("JavaFX", "Herbert"))
        books.add(Book("Android Essentials", "Kathy"))
        books.add(Book("Android Development", "Bruce"))
        books.add(Book("Kotlin Coding", "Brain Goetz"))
        books.add(Book("Java", "Horstman"))
        books.add(Book("Kotlin", "Joshua Bloch"))
        books.add(Book("JavaFX", "Herbert"))
        books.add(Book("Android Essentials", "Kathy"))
        books.add(Book("Android Development", "Bruce"))
        books.add(Book("Kotlin Coding", "Brain Goetz"))
        books.add(Book("Java", "Horstman"))
        books.add(Book("Kotlin", "Joshua Bloch"))
    }
}
```

```

books.add(Book("JavaFX", "Herbert"))
books.add(Book("Android Essentials", "Kathy"))
books.add(Book("Android Development", "Bruce"))
books.add(Book("Kotlin Coding", "Brain Goetz"))
books.add(Book("Java", "Horstman"))
books.add(Book("Kotlin", "Joshua Bloch"))
books.add(Book("JavaFX", "Herbert"))
books.add(Book("Android Essentials", "Kathy"))
books.add(Book("Android Development", "Bruce"))
books.add(Book("Kotlin Coding", "Brain Goetz"))
// Set the Layout Manager
recyclerView1.LayoutManager =LinearLayoutManager(this)
// Create an object for the MyAdapter
val adapter = MyAdapter(books)
// Set adapter to your RecyclerView
recyclerView1.adapter = adapter
    }
}

```

Step 15: If you do everything as per the screen the instructions, after running will get the output as below

