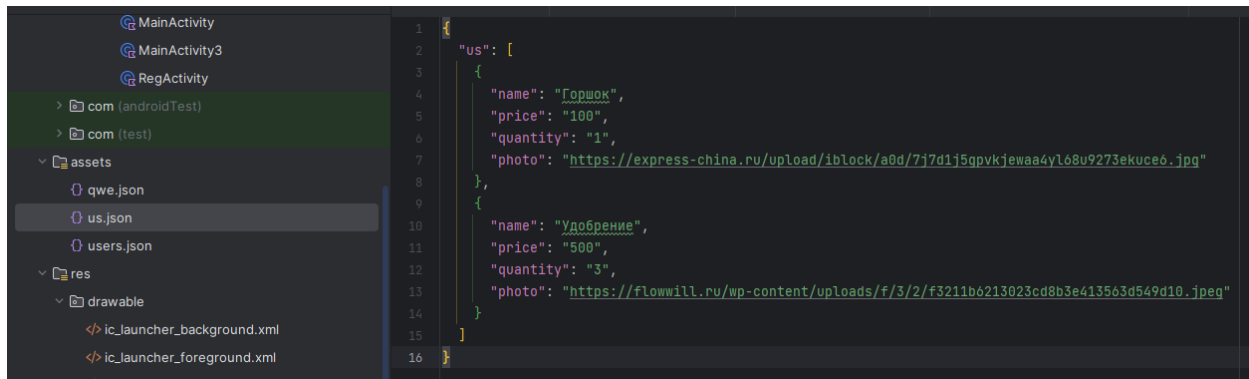


Лабораторная работа 4

Вывод данных из JSON

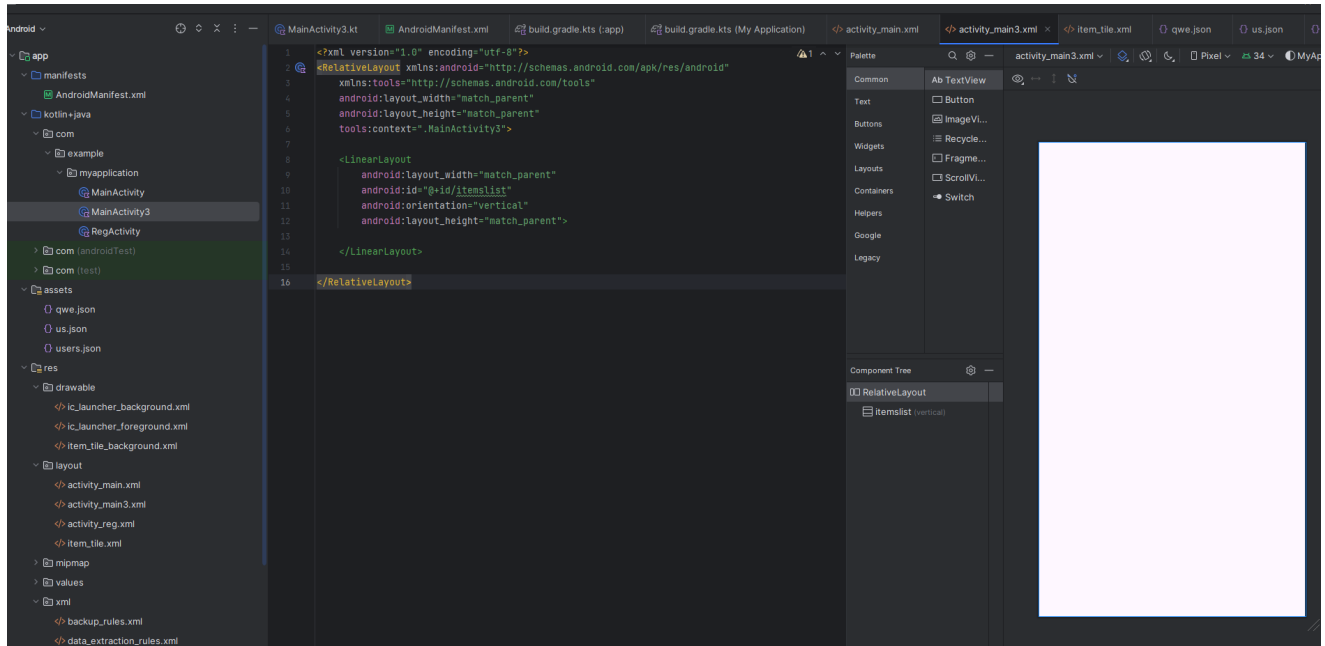
Первое что нам нужно сделать, это добавить JSON с данными наших товаров. Для этого:

1. В папке assets создаем файл с расширением .json
2. Вводим необходимые тестовые данные



```
{
  "us": [
    {
      "name": "Горшок",
      "price": "100",
      "quantity": "1",
      "photo": "https://express-
china.ru/upload/iblock/a0d/7j7d1j5gpvkjewaa4yl68u9273ekuce6.jpg"
    },
    {
      "name": "Удобрение",
      "price": "500",
      "quantity": "3",
      "photo": "https://flowwill.ru/wp-
content/uploads/f/3/2/f3211b6213023cd8b3e413563d549d10.jpeg"
    }
  ]
}
```

Теперь подготавливаем Activity со списком товаров

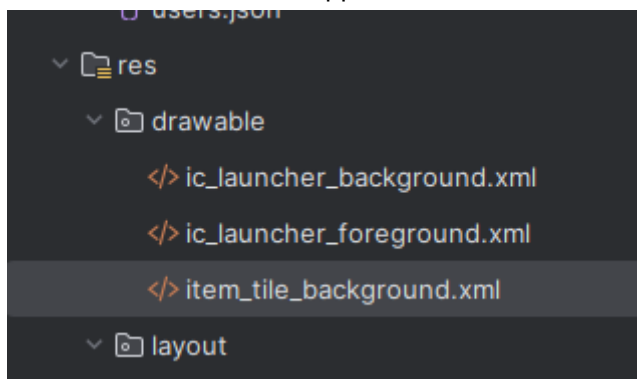


На окне будет только вертикальный LinearLayout для отображения товаров

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    tools:context=".MainActivity3">
    <LinearLayout
        android:layout_width="match_parent"
        android:id="@+id/itemslist"
        android:orientation="vertical"
        android:layout_height="match_parent">
    </LinearLayout>
</RelativeLayout>
```

Теперь создаем xml для блока нашего товара, но сначала добавим белую подложку с тенью.

В папке **drawable** создаем xml с именем item_tile_background



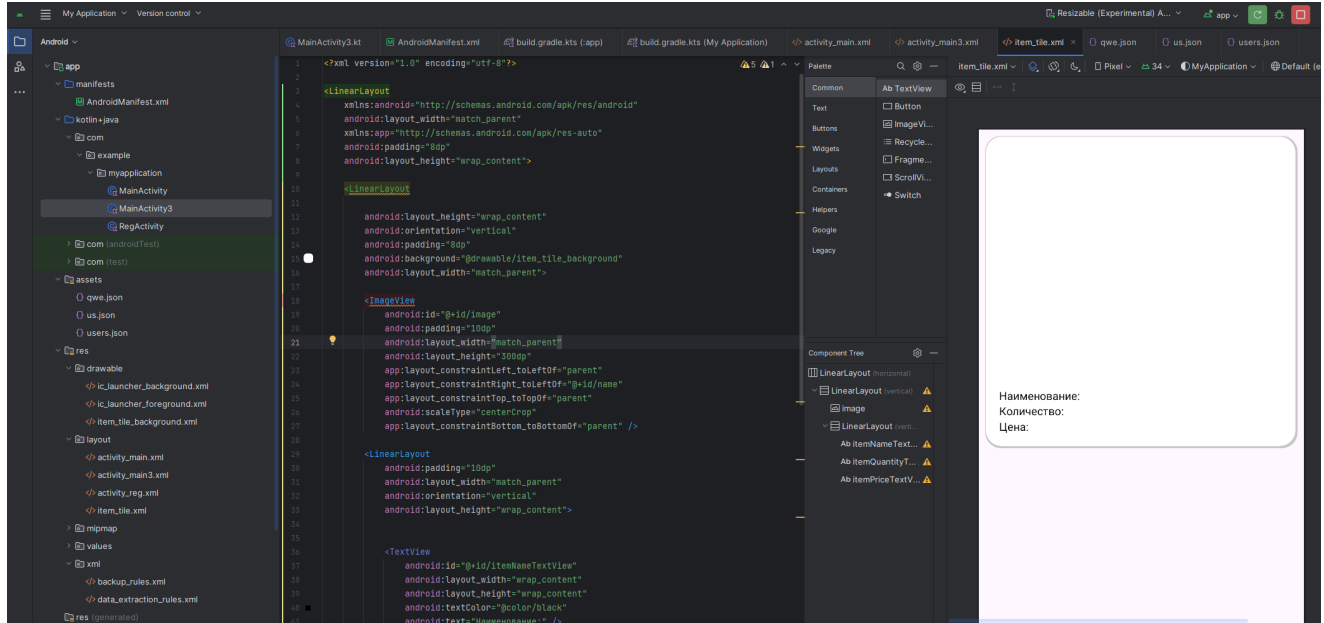
Прописываем:

```

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <layer-list>
            <!-- SHADOW LAYER -->
            <item>
                <shape>
                    <solid android:color="#35000000" />
                    <corners android:radius="20dp" />
                </shape>
            </item>
            <!-- CONTENT LAYER -->
            <item android:bottom="3dp" android:left="1dp"
                android:right="3dp" android:top="1dp">
                <shape>
                    <solid android:color="#ffffff" />
                    <corners android:radius="20dp" />
                </shape>
            </item>
        </layer-list>
    </item>
</selector>

```

Создаем также в папке **layout** xml с именем **item_tile**



В блоке будут поля для отображения имени, количества и цены товара. Изображение будет выводиться в **ImageView**.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:padding="8dp"

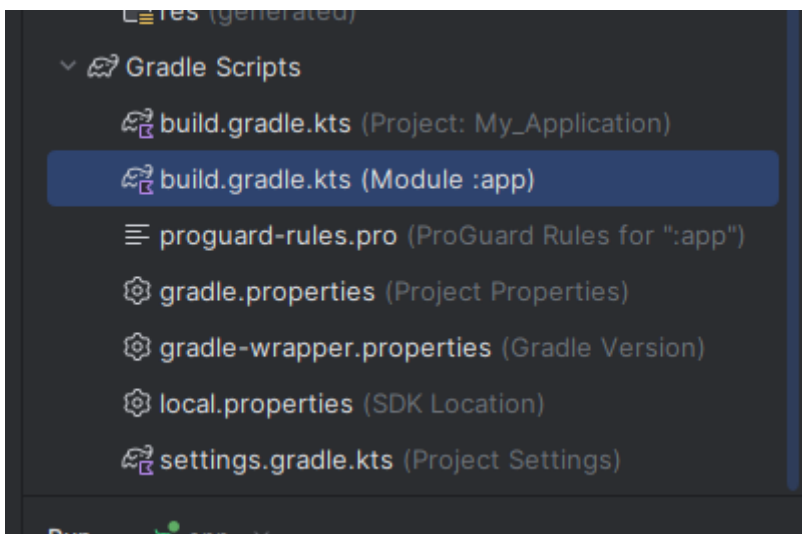
```

```

android:layout_height="wrap_content">
<LinearLayout
android:layout_height="wrap_content"
android:orientation="vertical"
android:padding="8dp"
android:background="@drawable/item_tile_background"
android:layout_width="match_parent">
<ImageView
android:id="@+id/image"
android:padding="10dp"
android:layout_width="match_parent"
android:layout_height="300dp"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toLeftOf="@+id/name"
app:layout_constraintTop_toTopOf="parent"
android:scaleType="centerCrop"
app:layout_constraintBottom_toBottomOf="parent" />
<LinearLayout
android:padding="10dp"
android:layout_width="match_parent"
android:orientation="vertical"
android:layout_height="wrap_content">
<TextView
android:id="@+id/itemNameTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textColor="@color/black"
android:text="Наименование: " />
<TextView
android:id="@+id/itemQuantityTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textColor="@color/black"
android:text="Количество: " />
<TextView
android:id="@+id/itemPriceTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textColor="@color/black"
android:text="Цена: " />
</LinearLayout>
</LinearLayout>
</LinearLayout>

```

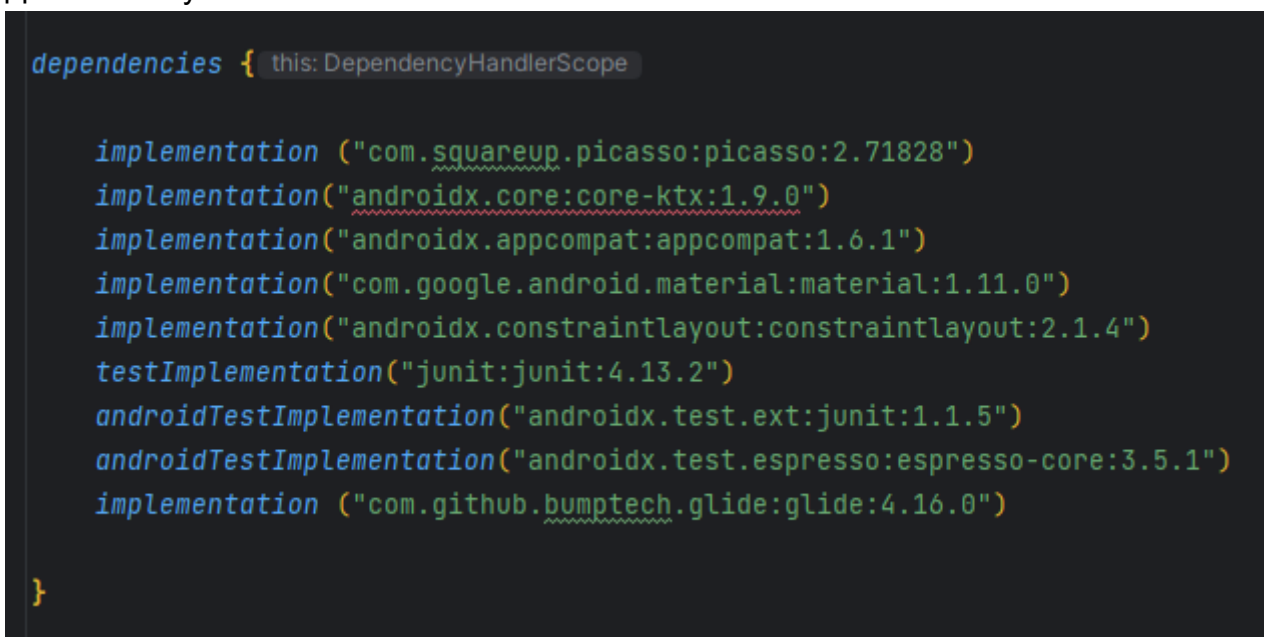
Готово, теперь установим библиотеку Glide для удобной установки изображения в ImageView. Перейдем в build.gradle.kts.



Находим список наших зависимостей, добавляем:

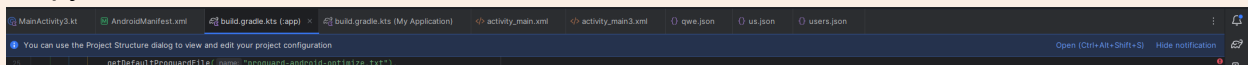
```
implementation ("com.github.bumptech.glide:glide:4.16.0")
```

Должно получиться так

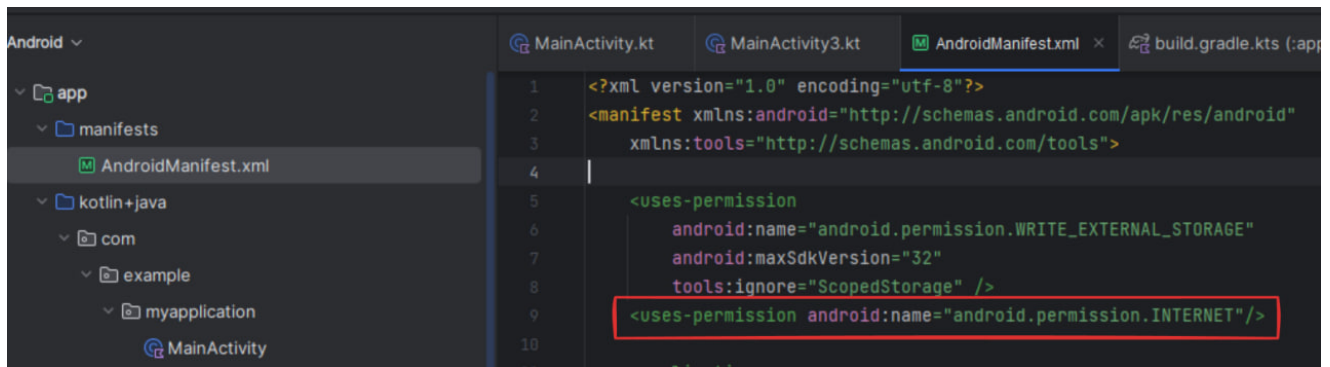


⚠ Не забудьте Sync Now

Сверху может появиться предложение о синхронизации проекта с его конфигурационными файлами, обязательно жмем **Sync Now** и ожидаем загрузки зависимостей



Так же добавляем разрешение в манифесте к интернету. Мы же хотим чтобы наши фотографии загружались?)



Наконец прописываем код нашему Activity

```
package com.example.myapplication
import android.annotation.SuppressLint
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ImageView
import android.widget.LinearLayout
import android.widget.TextView
import com.bumptech.glide.Glide
import com.example.myapplication.R
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch
import org.json.JSONArray
import org.json.JSONObject
import java.io.BufferedReader
import java.io.InputStreamReader

class MainActivity3 : AppCompatActivity() {
    @SuppressLint("MissingInflatedId", "SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main3)

        val itemslist : LinearLayout = findViewById(R.id.itemslist)

        // Прочитать JSON из файла
        val jsonString = readJSONFromAsset()

        // Создать JSON объект из строки
        val jsonObject = JSONObject(jsonString)

        // Получить массив товаров из JSON объекта
        val usArray = jsonObject.getJSONArray("us")

        // Отобразить JSON данные
        for(i in 0 until usArray.length()){
            //Создаем тайл товара
```

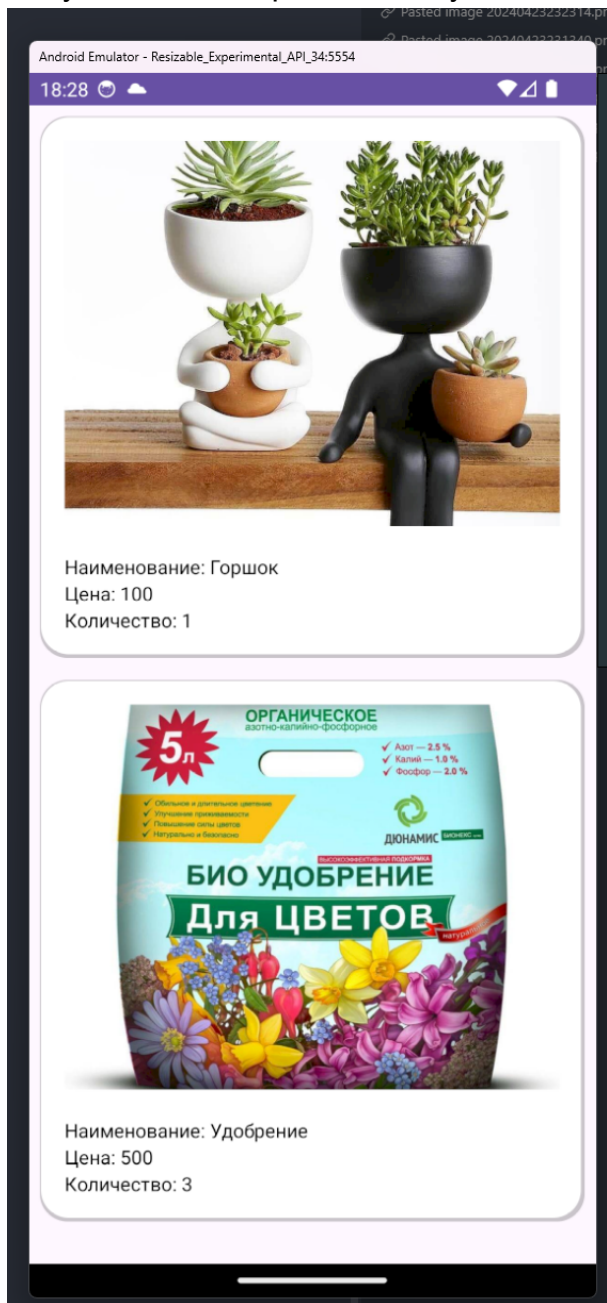
```

val tile = inflater.inflate(R.layout.item_tile, null)
//Устанавливаем значения
tile.findViewById<TextView>(R.id.itemNameTextView).text = "Наименование:
${usArray.getJSONObject(i).getString("name")}"
tile.findViewById<TextView>(R.id.itemQuantityTextView).text = "Цена:
${usArray.getJSONObject(i).getInt("price")}"
tile.findViewById<TextView>(R.id.itemPriceTextView).text = "Количество:
${usArray.getJSONObject(i).getInt("quantity")}"
//Загружаем изображение
GlobalScope.launch(Dispatchers.Main) {
Glide.with(this@MainActivity3)
.load(usArray.getJSONObject(i).getString("photo"))
.into(tile.findViewById<ImageView>(R.id.image))
}
//Добавляем в лист
itemslist.addView(tile)
}
}

private fun readJSONFromAsset(): String {
val inputStream = assets.open("us.json")
val bufferedReader = BufferedReader(InputStreamReader(inputStream))
val stringBuilder = StringBuilder()
var line: String?
while (bufferedReader.readLine().also { line = it } != null) {
stringBuilder.append(line)
}
return stringBuilder.toString()
}
}
}

```

Запускаем и смотрим что получилось



Самостоятельно реализовать:

- Поиск товаров по наименованию
- Сортировку товаров по цене