(Build.gradle(Module))

defaultConfig **{<…>}**viewBinding **{**enabled true**}**

<...>

dependencies **{**androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
 implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
 implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
 implementation 'com.squareup.picasso:picasso:2.71828'  
 implementation 'com.github.bumptech.glide:glide:4.12.0'  
 implementation 'com.android.databinding:viewbinding:7.1.3'  
 implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.4.1'  
 implementation "org.jetbrains.kotlinx:kotlinx-coroutines-android:1.5.0"**}**

**Interface ApiService.kt**

import retrofit2.http.GET  
interface ApiService {@GET("api/recipes")  
suspend fun getRecipe() : Responce}

Object ApiConfig.kt

import retrofit2.Retrofit  
import retrofit2.converter.gson.GsonConverterFactory  
object ApiConfig {  
 fun getService() : ApiService {  
 val client = Retrofit.Builder().  
 baseUrl("<…>/")  
 .addConverterFactory(GsonConverterFactory.create()).build()  
 return client.create(ApiService::class.java)}}

Class ListAdapter.kt  
import android.view.LayoutInflater  
import android.view.ViewGroup  
import androidx.recyclerview.widget.RecyclerView  
import com.bumptech.glide.Glide  
import com.example.podgotovochka.databinding.ItemListBinding  
class ListAdapter (private val responce: ArrayList<Responces>) :  
 RecyclerView.Adapter<ListAdapter.ViewHolder>() {  
 class ViewHolder(private val binding: ItemListBinding) :  
 RecyclerView.ViewHolder(binding.root) {  
 fun bind(responce: Responces) {  
 binding.title.text = responce.title.toString()  
 Glide.with(itemView).load(responce.thumb).into(binding.img)}}  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
val view=ItemListBinding.inflate(LayoutInflater.from(parent.context),  
 parent, false)  
 return ViewHolder(view)}  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 holder.bind(responce[position])}  
 override fun getItemCount(): Int {return responce.size}}

Class MainActivity.kt  
import android.os.Bundle  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.view.isVisible  
import androidx.recyclerview.widget.LinearLayoutManager  
import com.example.podgotovochka.databinding.ActivityMainBinding  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
class MainActivity : AppCompatActivity() {  
 private lateinit var binding : ActivityMainBinding  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 binding = ActivityMainBinding.inflate(layoutInflater)  
 setContentView(binding.root)  
 isLoading(true)  
 CoroutineScope(Dispatchers.IO).launch **{**getRecipe()**}**}  
 private fun isLoading(state: Boolean) {  
 binding.loading.isVisible = state}  
 private suspend fun getRecipe() {  
 val responce = ApiConfig.getService().getRecipe()  
 val results = responce.results  
 runOnUiThread **{**showRecipe(results)**}**}  
 private fun showRecipe(results: List<Responces>) {  
 val adapter = ListAdapter(results as ArrayList)  
 binding.rv.layoutManager = LinearLayoutManager(this)  
 binding.rv.adapter = adapter  
 isLoading(false)}}

(Kotlin Class/File)Responce.kt

package com.example.podgotovochka  
data class Responces(var title : String? = null,  
 var thumb: String? = null)  
data class Responce(var results: List<Responces>)

activity\_main.xml

<?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/rv"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
 <ProgressBar  
 android:id="@+id/loading"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"/>  
</androidx.constraintlayout.widget.ConstraintLayout>

item\_list.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingHorizontal="16dp"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 <ImageView  
 android:id="@+id/img"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
 <TextView  
 android:id="@+id/title"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toEndOf="@+id/img"  
 android:layout\_marginStart="16dp"  
 app:layout\_constraintTop\_toTopOf="@+id/img"  
 app:layout\_constraintBottom\_toBottomOf="@+id/img"/>  
</androidx.constraintlayout.widget.ConstraintLayout>

1) Rest.kt

data class Rest(

var answer:String? = null,

var symb:String? = null

)2)RetrofitService.kt

interface RetrofitService {

@GET("")

fun getList(@Query("input")input:String): Call<Rest>

}3)RetrofitClient.kt

object RetrofitClient {

private var retrofit: Retrofit?=null

fun getClient(baseUrl:String):Retrofit{

if(retrofit==null){

retrofit = Retrofit.Builder().baseUrl(baseUrl).addConverterFactory(GsonConverterFactory.create()).build()

}

return retrofit!!

}

}4)Common.kt

object Common { private val BASE\_URL = "1238123127893127893128937123897"

val retrofitService: RetrofitService

get() = RetrofitClient.getClient(BASE\_URL)

.create(RetrofitService::class.java)

}5)Adapter.kt

class MyAdapter(private val nameList:MutableList<Rest>, val mainActivity: MainActivity):RecyclerView.Adapter<MyAdapter.MyViewHolder>() {

class MyViewHolder(itemView: View):RecyclerView.ViewHolder(itemView)

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder { val list = LayoutInflater.from(parent.context).inflate(R.layout.item\_layout, parent, false )

return MyViewHolder(list)

}

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

val list = nameList[position]

holder.itemView.findViewById<TextView>(R.id.answer).text = list.answer

}

override fun getItemCount() = nameList.size

}6) MainActivity.kt

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main) findViewById<Button>(R.id.button).setOnClickListener {

val editText = findViewById<EditText>(R.id.editText)

val recyclerView = findViewById<RecyclerView>(R.id.recyclerView)

val call = Common.retrofitService.getList(editText.text.toString())

call.enqueue(object : Callback<Rest>{

override fun onResponse(call: Call<Rest>, response: Response<Rest>) {

val list = response.body()!!

recyclerView.adapter = MyAdapter(mutableListOf(list), this@MainActivity) recyclerView.layoutManager = LinearLayoutManager(this@MainActivity)

}

override fun onFailure(call: Call<Rest>, t: Throwable) {

Toast.makeText(this@MainActivity, t.localizedMessage, Toast.LENGTH\_LONG).show()

}

})

}

}

}