

# Recipe App

## Code

```
package com.example.recipeapp
import android.content.Intent
import android.os.Bundle
import android.widget.SearchView
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.room.Room
import com.example.recipeapp.databinding.ActivityMainBinding
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private val recipeAdapter = RecipeAdapter()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.recyclerView.layoutManager = LinearLayoutManager(this)
        binding.recyclerView.adapter = recipeAdapter

        binding.searchView.setOnQueryTextListener(object :
        SearchView.OnQueryTextListener {
            override fun onQueryTextSubmit(query: String?): Boolean {
                query?.let { searchRecipes(it) }
                return true
            }

            override fun onQueryTextChange(newText: String?): Boolean {
                newText?.let { searchRecipes(it) }
                return true
            }
        })

        loadRecipes()
    }

    private fun loadRecipes() {
        val db = Room.databaseBuilder(
            applicationContext,
            RecipeDatabase::class.java, "recipe_database"
        ).build()
    }
}
```

```

        CoroutineScope(Dispatchers.IO).launch {
            db.recipeDao().insert(RecipeEntity("Spaghetti", "Delicious spaghetti recipe"))
        }
    }

    private fun searchRecipes(query: String) {
        val apiService = Retrofit.Builder()
            .baseUrl("https://api.example.com/")
            .addConverterFactory(GsonConverterFactory.create())
            .build()
            .create(RecipeApiService::class.java)

        apiService.searchRecipes(query).enqueue(object :
Callback<RecipeResponse> {
            override fun onResponse(call: Call<RecipeResponse>, response:
Response<RecipeResponse>) {
                if (response.isSuccessful) {
                    recipeAdapter.submitList(response.body()?.results)
                }
            }

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

```

## RecipeAdapter.kt

```

package com.example.recipeapp

import android.content.Intent

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.DiffUtil
import androidx.recyclerview.widget.ListAdapter
import androidx.recyclerview.widget.RecyclerView
import com.example.recipeapp.databinding.ItemRecipeBinding

class RecipeAdapter : ListAdapter<Recipe,
RecipeAdapter.RecipeViewHolder>(RecipeDiffCallback()) {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
RecipeViewHolder {
        val binding =
ItemRecipeBinding.inflate(LayoutInflater.from(parent.context), parent, false)
        return RecipeViewHolder(binding)
    }

    override fun onBindViewHolder(holder: RecipeViewHolder, position: Int) {
        holder.bind(getItem(position))
        holder.itemView.setOnClickListener {
            val context = holder.itemView.context
            val intent = Intent(context, RecipeDetailsActivity::class.java)

```

```

        intent.putExtra("recipe", getItem(position))
        context.startActivity(intent)
    }
}

class RecipeViewHolder(private val binding: ItemRecipeBinding) :
RecyclerView.ViewHolder(binding.root) {
    fun bind(recipe: Recipe) {
        binding.recipeName.text = recipe.name
        binding.recipeDescription.text = recipe.description
    }
}

class RecipeDiffCallback : DiffUtil.ItemCallback<Recipe>() {
    override fun areItemsTheSame(oldItem: Recipe, newItem: Recipe) =
oldItem.name == newItem.name
    override fun areContentsTheSame(oldItem: Recipe, newItem: Recipe) =
oldItem == newItem
}
}

```

## RecipeDetailsActivity.kt

```

package com.example.recipeapp

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.recipeapp.databinding.ActivityRecipeDetailsBinding

class RecipeDetailsActivity : AppCompatActivity() {
    private lateinit var binding: ActivityRecipeDetailsBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityRecipeDetailsBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val recipe = intent.getSerializableExtra("recipe") as Recipe
        binding.recipeName.text = recipe.name
        binding.recipeDescription.text = recipe.description
    }
}

```

## Recipe.kt

```

package com.example.recipeapp

data class Recipe(
    val name: String,
    val description: String
)

```

## RecipeDao.kt

```

package com.example.recipeapp

```

```

import androidx.room.Dao
import androidx.room.Insert
import androidx.room.Query

@Dao
interface RecipeDao {
    @Insert
    suspend fun insert(recipe: RecipeEntity)

    @Query("SELECT * FROM RecipeEntity")
    suspend fun getAll(): List<RecipeEntity>
}

```

## RecipeDatabase.kt

```

package com.example.recipeapp

import androidx.room.Database
import androidx.room.RoomDatabase

@Database(entities = [RecipeEntity::class], version = 1)
abstract class RecipeDatabase : RoomDatabase() {
    abstract fun recipeDao(): RecipeDao
}

```

## RecipeEntity.kt

```

package com.example.recipeapp
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity
data class RecipeEntity(
    @PrimaryKey val name: String,
    val description: String
)

```

## RecipeApiService.kt

```

package com.example.recipeapp
import retrofit2.Call
import retrofit2.http.GET
import retrofit2.http.Query

interface RecipeApiService {
    @GET("recipes/search")
    fun searchRecipes(@Query("query") query: String): Call<RecipeResponse>
}

data class RecipeResponse(
    val results: List<Recipe>
)

```

## build.gradle (Module: app)

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'kotlin-kapt'
}

android {
    compileSdkVersion 33
    defaultConfig {
        applicationId "com.example.recipeapp"
        minSdkVersion 21
        targetSdkVersion 33
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation "androidx.core:core-ktx:1.10.1"
    implementation "androidx.appcompat:appcompat:1.6.1"
    implementation "com.google.android.material:material:1.9.0"
    implementation "androidx.constraintlayout:constraintlayout:2.1.4"
    implementation "androidx.lifecycle:lifecycle-runtime-ktx:2.6.1"
    implementation "org.jetbrains.kotlin:kotlin-stdlib:1.9.0"
    implementation "androidx.room:room-runtime:2.5.2"
    kapt "androidx.room:room-compiler:2.5.2"
    implementation "com.squareup.retrofit2:retrofit:2.9.0"
    implementation "com.squareup.retrofit2:converter-gson:2.9.0"
    implementation "androidx.recyclerview:recyclerview:1.3.1"
    testImplementation "junit:junit:4.13.2"
    androidTestImplementation "androidx.test.ext:junit:1.1.5"
    androidTestImplementation "androidx.test.espresso:espresso-core:3.5.1"
}
```