

Instituto Superior Tecnológico del Azuay

Tecnología Superior en Desarrollo de Software



Instituto Superior
Tecnológico del Azuay

Programación de Aplicaciones Móviles

API Rest con Retrofit

Nombre

Danny Tenemea

M4B

Docente:

Mgtr. Patricio Pacheco

Cuenca - Ecuador

Introducción

Esta guía práctica está centrada en cómo consumir una API o servicios web desde una aplicación Android.

Si por ejemplo tienes una base de datos, pero no tienes una API creada.

Entonces primero deberías definir una API.

Una API es un intermediario entre una base de datos y una aplicación móvil

Objetivo de la Guía

Consumir una API (servicios web) utilizando el IDE de Desarrollo de Android Studio con la librería Retrofit y procesar la respuesta JSON obtenida.

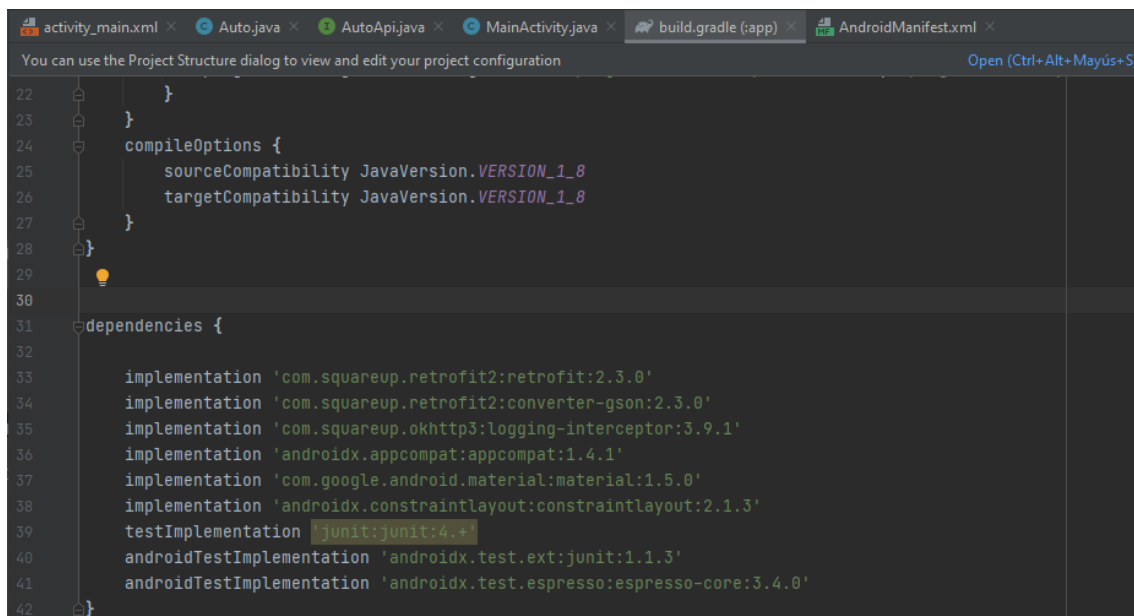
Materiales, herramientas, equipos y software

- Equipos de computación,
- Android Studio
- Internet,
- Material Guía (Talleres, ejercicios prácticos).

Procedimiento

Añadir la librería de Retrofit a nuestro proyecto.

- Existen varias formas de añadir dependencias a nuestro proyecto.
- En este caso usaremos el método más común y recomendado: añadiremos Retrofit vía Gradle.

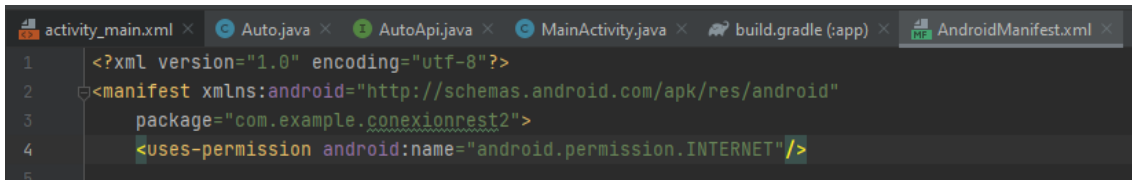


The screenshot shows the 'build.gradle (app)' file in Android Studio. The file contains the following code:

```
22     }
23 }
24 compileOptions {
25     sourceCompatibility JavaVersion.VERSION_1_8
26     targetCompatibility JavaVersion.VERSION_1_8
27 }
28 }
29
30
31 dependencies {
32
33     implementation 'com.squareup.retrofit2:retrofit:2.3.0'
34     implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
35     implementation 'com.squareup.okhttp3:logging-interceptor:3.9.1'
36     implementation 'androidx.appcompat:appcompat:1.4.1'
37     implementation 'com.google.android.material:material:1.5.0'
38     implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
39     testImplementation 'junit:junit:4.+'
40     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
41     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
42 }
```

Solicitar Permisos

- Antes de empezar a configurar Retrofit en nuestro proyecto, es importante que nuestra aplicación se pueda conectar a internet.
- Para solicitar este permiso debemos añadir la siguiente línea a nuestro archivo manifest

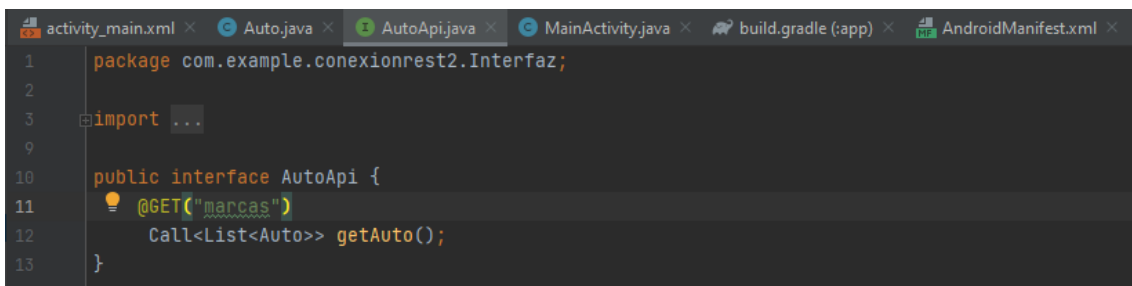


```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.conexionrest2">
4      <uses-permission android:name="android.permission.INTERNET"/>
5  
```

Crea Una clase y una Interfaz

Creamos una interfaz en la cual vamos a insertar el método GET que nos permitirá listar las marcas de los autos que tenemos en nuestra API



```

1  package com.example.conexionrest2.Interfaz;
2
3  import ...
4
5
6
7
8
9
10 public interface AutoApi {
11     @GET("marcas")
12     Call<List<Auto>> getAuto();
13 }

```

Creamos una clase donde vamos a crear los atributos, los cuales tienen que ser nombrados de la misma manera que los campos que existen en nuestra API



```

1  package com.example.conexionrest2.Modelo;
2
3  public class Auto {
4      private String nome;
5      private String codigo;
6
7      public String getNome() {
8          return nome;
9      }
10
11      public void setNome(String nome) { this.nome = nome; }
12
13      public String getCodigo() { return codigo; }
14
15      public void setCodigo(String codigo) { this.codigo = codigo; }
16
17
18
19
20
21
22 }
23

```

Ejemplo de API Adapter

<https://parallelum.com.br/fipe/api/v1/carros/marcas>

```

activity_main.xml x Auto.java x AutoApi.java x MainActivity.java x build.gradle (:app) x AndroidManifest.xml x
1 package com.example.conexionrest2;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 import com.example.conexionrest2.Interfaz.AutoApi;
9 import com.example.conexionrest2.Modelo.Auto;
10
11 import java.util.List;
12
13 import retrofit2.Call;
14 import retrofit2.Callback;
15 import retrofit2.Response;
16 import retrofit2.Retrofit;
17 import retrofit2.converter.gson.GsonConverterFactory;

```

```

activity_main.xml x Auto.java x AutoApi.java x MainActivity.java x build.gradle (:app) x AndroidManifest.xml x
19 public class MainActivity extends AppCompatActivity {
20     private TextView auto_item;
21
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_main);
26         auto_item=findViewById(R.id.auto_item);
27         getAuto();
28     }
29
30     private void getAuto(){
31         Retrofit retrofit = new Retrofit.Builder()
32             .baseUrl("https://parallelum.com.br/fipe/api/v1/carros/")
33             .addConverterFactory(GsonConverterFactory.create())
34             .build();
35         AutoApi autoapi=retrofit.create(AutoApi.class);
36         Call<List<Auto>> call = autoapi.getAuto();
37
38         call.enqueue(new Callback<List<Auto>>() {
39             @Override
40             public void onResponse(Call<List<Auto>> call, Response<List<Auto>> response) {
41                 if(!response.isSuccessful()){
42                     auto_item.setText(response.code());
43                 }
44                 List<Auto> lista_auto = response.body();
45                 for(Auto auto: lista_auto){
46                     String contenido = "";
47                     contenido+="codigo"+auto.getCodigo()+"\n";

```

```

List<Auto> lista_auto = response.body();
for(Auto auto: lista_auto){
    String contenido = "";
    contenido+="codigo"+auto.getCodigo()+"\n";
    contenido+="marca"+auto.getNome()+"\n\n";
    auto_item.append(contenido);
}
}

@Override
public void onFailure(Call<List<Auto>> call, Throwable t) {
    auto_item.setText(t.getMessage());
}

```

Resultados esperados

Al final de la guía, el estudiante está en capacidad de consumir cualquier web service de tipo REST API.