Uso de Retrofit con Java

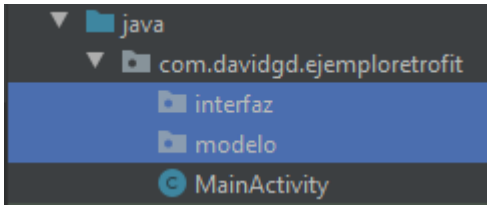
1. Hay que colocar los permisos de internet en el manifest.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.davidgd.ejemploretrofit">

    <!--Permiso de internet para conectar a las url-->
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Ejemplo Retrofit"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

2. En el Gradle de la app colocar dentro de las dependencias el código que está en azul. Uno es para el uso de Retrofit y el otro para el convertidor GSON.

```
dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

    implementation 'com.squareup.retrofit2:retrofit:2.5.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
}
```

3. Crear una carpeta para el modelo y otra para la interfaz, dentro de la carpeta que contiene nuestro MainActivity.java

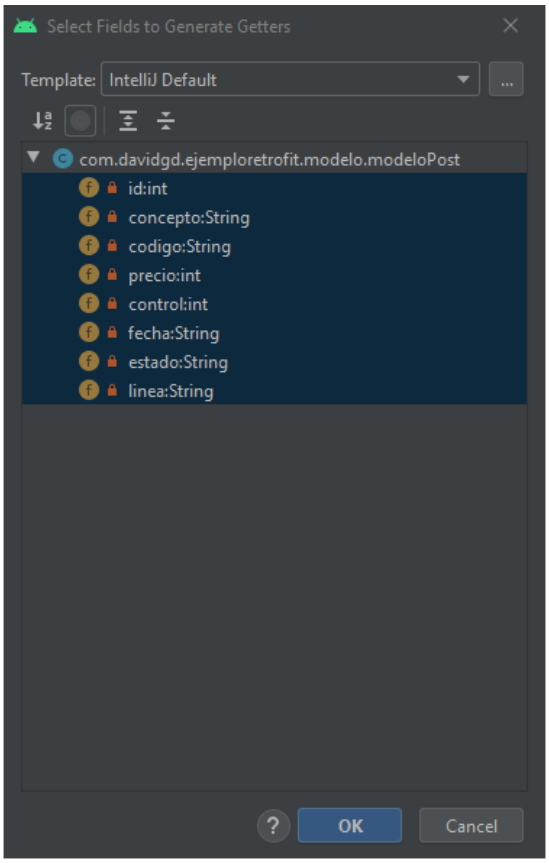


4. Dentro de la carpeta modelo crearemos una clase de java llamada modeloPost y dentro de la clase colocaremos las variables que tendrán el mismo nombre que las keys de nuestro archivo json.

```
{
    id: "5",
    concepto: "APORTACIONES PARA VIGENCIA ACADEMICA",
    codigo: "ISC000000B004001000004",
    precio: "500",
    control: "161070298",
    fecha: "05/06/2021",
    estado: "Cancelada",
    linea: ""
},
```

```java
package com.davidgd.ejemploretrofit.modelo;

public class modeloPost {

    //Nombres de los keys de nuestro archivo Json
    private int id;
    private String concepto;
    private String codigo;
    private int precio;
    private int control;
    private String fecha;
    private String estado;
    private String linea;

}
```

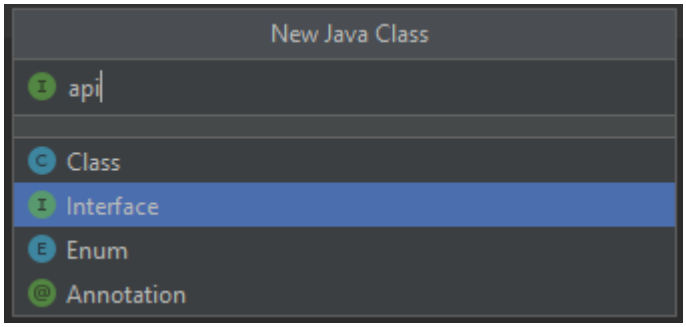
5. Para finalizar crearemos los getters de todas las variables, para lo cual presionaremos las teclas alt+ins, seleccionaremos Getter, seleccionamos todo con control+clic y al final le daremos ok.



Lo cual nos generara lo siguiente:

```
15      public int getId() {
16          return id;
17      }
18
19      public String getConcepto() {
20          return concepto;
21      }
22
23      public String getCodigo() {
24          return codigo;
25      }
26
27      public int getPrecio() {
28          return precio;
29      }
30
31      public int getControl() {
32          return control;
33      }
34
35      public String getFecha() {
36          return fecha;
37      }
```

6. Dentro de la carpeta interfaz crearemos una interfaz de la misma forma que se crea una nueva clase de java pero seleccionaremos interface en este caso la nombrare api.



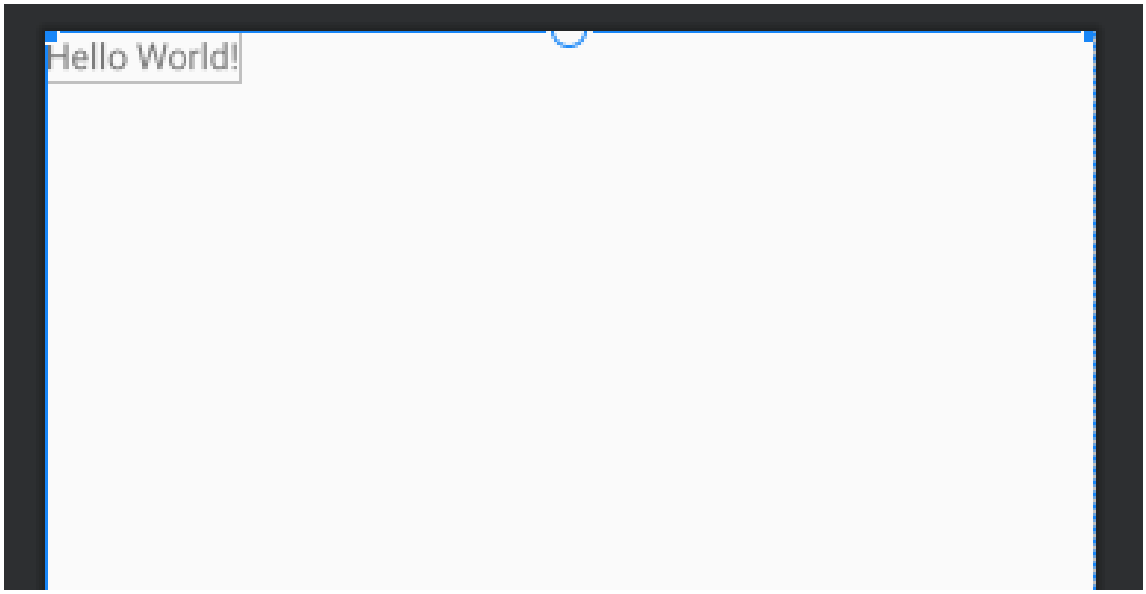
7. Escribiremos el siguiente código, recuerda colocar la última parte de la ruta de la api e importar modeloPost (nombre de nuestro modelo) con alt+enter sobre las letras rojas si es que aparecen.

```
1   package com.davidgd.ejemploretrofit.interfaz;
2
3   import com.davidgd.ejemploretrofit.modelo.modeloPost;
4   import java.util.List;
5   import retrofit2.Call;
6   import retrofit2.http.GET;
7
8   public interface api {
9
10      @GET("rest.php") //Ruta final de nuestra url
11      Call<List<modeloPost>> getPosts(); //Recuerda importar modeloPost
12
13  }
```

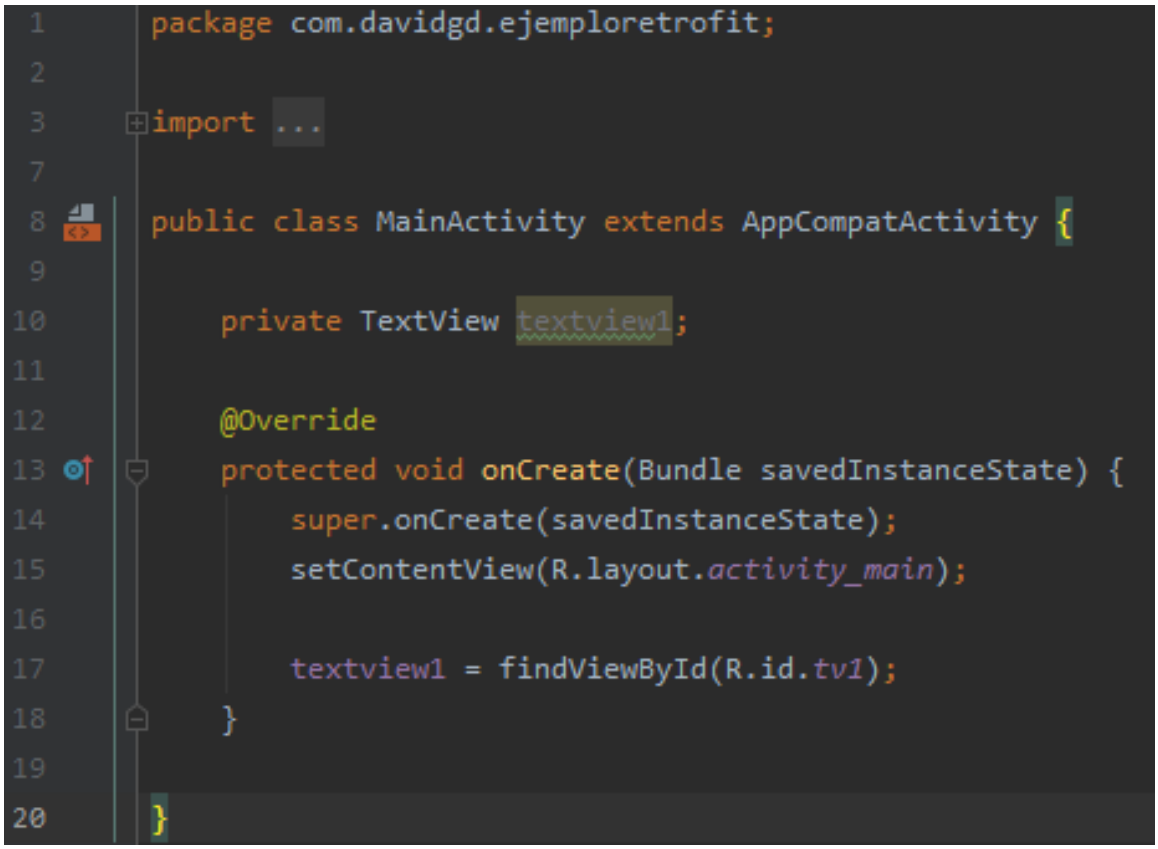
8. Ahora nos dirigiremos a nuestro archivo xml donde se encuentra el diseño de nuestro app colocaremos un scroll view y dentro del scroll colocaremos un text view con el id tv1 de la siguiente forma. Si es posible prueba la app.

```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">

    <!--Aqui se va a mostrar el JSON-->
    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/tv1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

    </androidx.core.widget.NestedScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Nos debería de mostrar un texto como el siguiente dentro de la app:

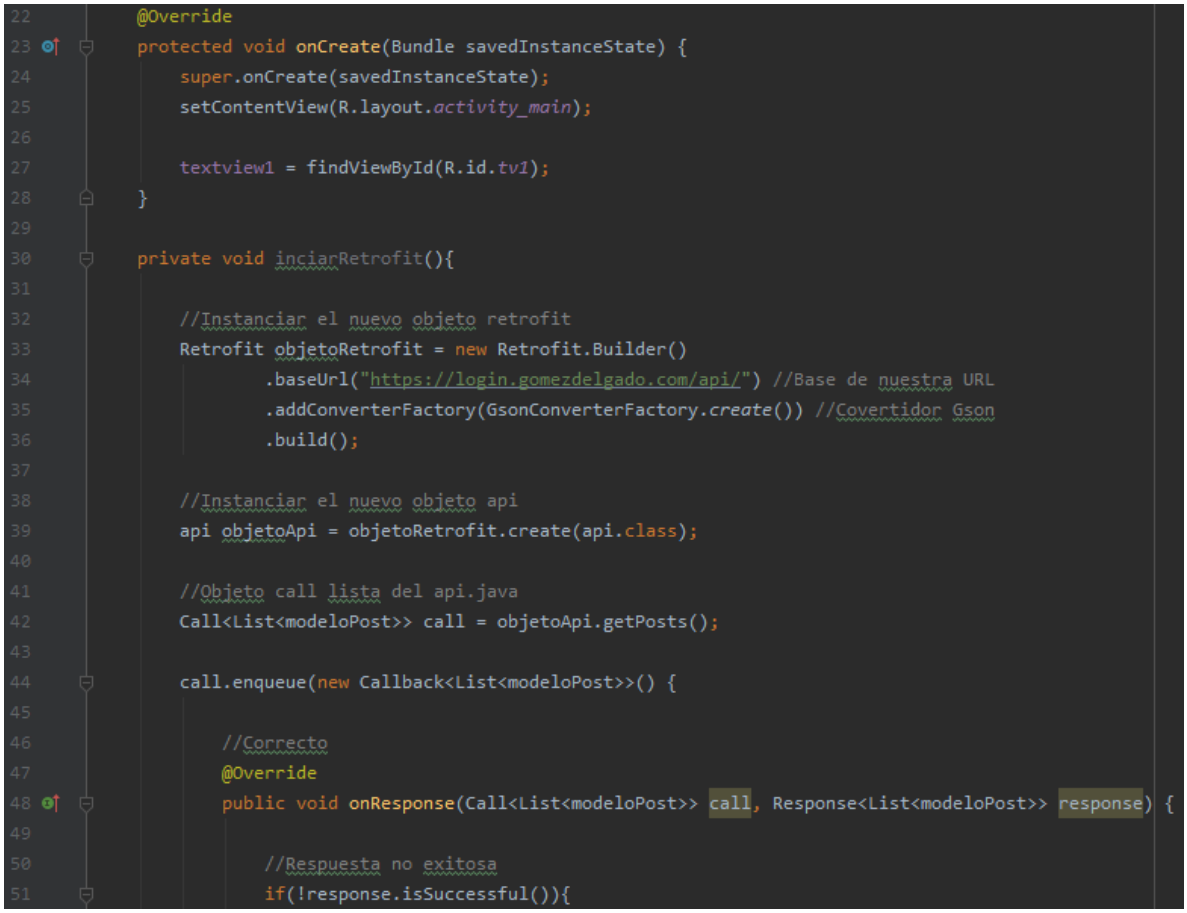


9. Ahora comenzaremos a escribir en el MainActivity.java lo primero será crear una variable llamada textview1 de tipo TextView la cual colocaremos con TextView de nuestro xml al cual le pusimos el id tv1.



```java
package com.davidgd.ejemploretrofit;

import ...

public class MainActivity extends AppCompatActivity {

    private TextView textview1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textview1 = findViewById(R.id.tv1);
    }

}
```

10. Ahora fuera del onCreate crearemos la función iniciarRetrofit.



```
22      @Override
23      protected void onCreate(Bundle savedInstanceState) {
24          super.onCreate(savedInstanceState);
25          setContentView(R.layout.activity_main);
26
27          textview1 = findViewById(R.id.tv1);
28      }
29
30      private void inciarRetrofit(){
31
32          //Instanciar el nuevo objeto retrofit
33          Retrofit objetoRetrofit = new Retrofit.Builder()
34                  .baseUrl("https://login.gomezdelgado.com/api/") //Base de nuestra URL
35                  .addConverterFactory(GsonConverterFactory.create()) //Covertidor Gson
36                  .build();
37
38          //Instanciar el nuevo objeto api
39          api objetoApi = objetoRetrofit.create(api.class);
40
41          //Objeto call lista del api.java
42          Call<List<modeloPost>> call = objetoApi.getPosts();
43
44          call.enqueue(new Callback<List<modeloPost>>() {
45
46              //Correcto
47              @Override
48              public void onResponse(Call<List<modeloPost>> call, Response<List<modeloPost>> response) {
49
50                  //Respuesta no exitosa
51                  if(!response.isSuccessful()){
```

```
                    //Muestra el codigo
                    textview1.setText("Codigo: " + response.code());
                }

                //Guarda la respuesta (El json)
                List<modeloPost> listaPost = response.body();

                //Recorremos la lista con un ciclo foreach
                for(modeloPost dato: listaPost){

                    String  contenido = ""; //Donde se guardaran los datos

                    //A contenido sumale lo siguiente
                    contenido += "id: " + dato.getId() + "\n";
                    contenido += "concepto:" + dato.getConcepto() + "\n";
                    contenido += "codigo:" + dato.getCodigo() + "\n";
                    contenido += "precio:" + dato.getPrecio() + "\n";
                    contenido += "control:" + dato.getControl() + "\n";
                    contenido += "fecha:" + dato.getFecha() + "\n";
                    contenido += "estado:" + dato.getEstado() + "\n";
                    contenido += "linea:" + dato.getLinea() + "\n";

                    //guarda los datos de contenido al final del textview
                    textview1.append(contenido);
                }

            }
```

```
            //Error
            @Override
            public void onFailure(Call<List<modeloPost>> call, Throwable t) {
                textview1.setText(t.getMessage());
            }

        }); //Fin del enqueue (Poner en cola)

    } //Fin de la funcion iniciarRetrofit

}
```

11. Ya para finalizar llamamos a nuestra función o invocamos el método dentro del onCreate.

```
22      @Override
23      protected void onCreate(Bundle savedInstanceState) {
24          super.onCreate(savedInstanceState);
25          setContentView(R.layout.activity_main);
26
27          textview1 = findViewById(R.id.tv1);
28
29          //Llamamos a la funcion o invocamos el metodo iniciarRetrofit
30          inciarRetrofit();
31      }
```

Resultado