## Setting up Retrofit for Android networking!

**Gradle**

Added feature viewBinding.

buildFeatures **{** viewBinding true  
**}**

Added dependencies for Retrofit, Moshi and Retrofit Moshi-Converter.

implementation "com.squareup.retrofit2:retrofit:2.9.0"  
implementation "com.squareup.moshi:moshi-kotlin:1.12.0"  
implementation "com.squareup.retrofit2:converter-moshi:2.9.0"

**AndroidManifest**

<uses-permission android:name="android.permission.INTERNET" />

**Interface MovieDbService**

Function to getMovieById was created.  
It has a @GET(“value”) where the value is url part where will define that function.  
For example to get a specific movie, we have to add the value “movie/{movieID}”, which will be attached to the base URL to generate the expected result.

**Main Activity**

At first the main activity was used to build the network connection, just for testing purpose.

moshi was instantiated to convert the json results from the API.

retrofit was also instantiated and built, using the base URL: <https://api.themoviedb.org/3/>. To build it, it’s also necessary to add a converter, in which is Moshi declare above.

movieDbService is instantiated as retrofit.create, passing the interface MovieDbService as class.java.

Then the end part is to call the function created in movieDbService.

movieDbService.getPopularMovies().enqueue(object : Callback<Any>{  
 override fun onResponse(call: Call<Any>, response: Response<Any>) {  
 Log.i("print", response.toString())  
 }  
  
 override fun onFailure(call: Call<Any>, t: Throwable) {  
 Log.i("print", t.message?: "Null message")  
 }  
})

As it’s a network call, it can’t be done on the main activity thread. Therefore, we use “enqueue”, which asynchronously send the request to the server.