```
compileOptions {

    sourceCompatibility JavaVersion.VERSION_11
    targetCompatibility JavaVersion.VERSION_11
}
kotlinOptions {
    jvmTarget = JavaVersion.VERSION_11.toString()
}
buildFeatures {
    viewBinding = true
}
```

```
implementation "com.squareup.retrofit2:retrofit:2.9.0"
implementation "com.squareup.retrofit2:converter-gson:2.9.0"
```

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

```
class MainActivity : AppCompatActivity() {
    private lateinit var activity: Activity
    private lateinit var context: Context
    private val TAG = "MainActivity"
    private val BASE_URL = "https://jsonplaceholder.typicode.com/"
```

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    activity = this
    context = this
    setContentView(R.layout.activity_main)
```

```
data class Comment(
    @SerializedName("postId")
    var postId: Int? = null,
    @SerializedName("id")
    var id: Int? = null,
    @SerializedName("name")
    var name: String? = null,
    @SerializedName("email")
    var email: String? = null,
    @SerializedName("body")
    var body: String? = null
)
```

```
interface RetroApi {
    @GET("/comments")
    fun getComments(): Call<List<Comment>>
}
```

```
//
val api = Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(GsonConverterFactory.create())
    .build()
    .create(RetroApi::class.java)
```

```
//
api.getComments().enqueue(object : Callback<List<Comment>> {
    override fun onResponse(call: Call<List<Comment>>, response: Response<List<Comment>>) {
        if (response.isSuccessful) {
            response.body()?.let {
                for (comment in it) {
                    Log.d(TAG, "COMMENT_DATA: ${comment.toString()}")
                }
            }
        }
    }

    override fun onFailure(call: Call<List<Comment>>, t: Throwable) {
        Log.e(TAG, "ERROR: $t")
    }
})
```

```
GlobalScope.launch(Dispatchers.IO) {
    val comments = api.getComments().await()
    for(comment in comments) {
        Log.d(TAG, comment.toString())
    }
}
```

```
GlobalScope.launch(Dispatchers.IO) {
    val response = api.getComments().awaitResponse()
    if(response.isSuccessful) {
        for(comment in response.body()!!) {
            Log.d(TAG, comment.toString())
        }
    }
}
```

```
interface RetroApi {
    @GET("/comments")
    suspend fun getComments(): Response<List<Comment>>
}
```

```
GlobalScope.launch(Dispatchers.IO) {
    val response = api.getComments()
    if(response.isSuccessful) {
        for(comment in response.body()!!) {
            Log.d(TAG, comment.toString())
        }
    }
}
```