

Retrofit

É uma API de transmissão de dados, mantida pela Square e apresentada em 2013 como uma alternativa simples, rápida e eficiente em desenvolvimento da camada de comunicação de aplicações que utilizem padrão REST.

Para utilizarmos o Retrofit em nosso projeto devemos primeiro adicionar sua dependência no arquivo build.gradle

```
dependencies {  
    implementation "com.squareup.retrofit2:retrofit:2.9.0"  
    implementation "com.squareup.retrofit2:retrofit:2.9.0"  
    implementation "com.squareup.okhttp3:okhttp:4.2.2"  
    implementation  
        "com.squareup.okhttp3:logging-interceptor:4.2.2"  
    implementation  
        "org.jetbrains.kotlin:kotlin-coroutines-core:1.3.7"  
    implementation  
        'android.arch.lifecycle:extensions:1.1.1'  
    implementation  
        "com.squareup.picasso:picasso:2.71828"
```

} O Retrofit trabalha com o uso de Interface onde ele se baseará para criar o código de requisição em tempo de de compilação, ou seja, quando executarmos

nossa aplicação.

Vamos simular uma requisição na api Rest do site da [MARVEL](#) o qual nos fornece dados sobre as HQ's de super heróis da Marvel. Digamos que queremos obter uma lista de HQ's. e para isso devemos buscar os dados na seguinte Url

<https://gateway.marvel.com:443/v1/public/comics?format=magazine&formatType=comic&noVariants=true&orderBy=focDate&ts=1551213595&hash=9584823101fd40c435d4c028df6daca3&apikey=6eb7e8896ec5850c52515a8a23ee97f0>

Para tão tarefa devemos primeiro criar uma Interface para o Retrofit saiba como buscar esses dados.

```
interface ServiceMarvel {
```

```
    @GET("comic?")
```

```
    suspend fun getComicService(@Query("ts") ts : String,
```

```
                                @Query("hash") hash : String,
```

```
                                @Query("apikey") apikey : String
```

```
    ) : ComicResponse
```

```
}
```

Nesta interface estamos utilizando a anotação **@GET** que requer o nome do endpoint que será acessado e fará com que o retrofit mande os parâmetros da requisição direto na url, para representar esses parâmetros usamos a anotação **@Query** que que o nome do parâmetro a ser enviado. Esse método nos retorna

Use uma suspend fun para a ação poder ser realizada em outra corotina.

ComicsResponse é a classe modelo para é retornada na requisição da API, ou seja a classe que obtemos após o Retrofit fazer a conversão da nossa requisição.

Após criar a interface de acesso temos que criar a classe de serviço que irá iniciar as ações do retrofit. Assim que instanciamos a classe chamamos o atributo “retrofit”, passando a URL base desejada e a interface com as requisições, será feito a conexão com a API.

```
private val gsonConverter: GsonConverterFactory = GsonConverterFactory.create()
```

```
class RetrofitInit (url: String) {
```

```
    private val retrofit: Retrofit = Retrofit.Builder()
```

```
        .baseUrl(url)
```

```
        .addConverterFactory(gsonConverter)
```

```
        .build()
```

```
    fun <T : Any> create(clazz: KClass<T>): T = retrofit.create(clazz.java)
```

```
}
```

Agora precisamos criar nosso repositório onde irão ter as configurações para a conexão com a API, isso é sua **URL** base e seu **Service** que é sua interface criada anteriormente, além das configurações básicas para efetuar a conexão com servidor da Marvel necessário ter uma **“private_key”**, **“public_key”**, **“ts”** e o nosso **“hash”**.

Nosso repositório terá toda a configuração para realizar a conexão com API assim o ViewModel não precisará se preocupar com isso.

```
//url sera a url base apresentada na doc da api
private var url = "https://gateway.marvel.com/v1/public/"

//interface que carrega todos os end points
private var service = ServiceMarvel::class

private val public_key = "6eb7e8896ec5850c52515a8a23ee97f0"
private val private_key = "0dd0c16fedb8a02985977eafca66b49f5e6a526f"
private var ts = (System.currentTimeMillis() / 1000).toString()

private val hash = md5 (ts+public_key+private_key)
private val serviceMarvel = RetrofitInit(url).create(service)

suspend fun getComicService(): ComicResponse =
serviceMarvel.getComicService(ts, hash, public_key)
```