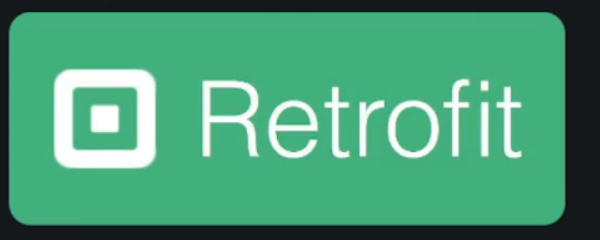
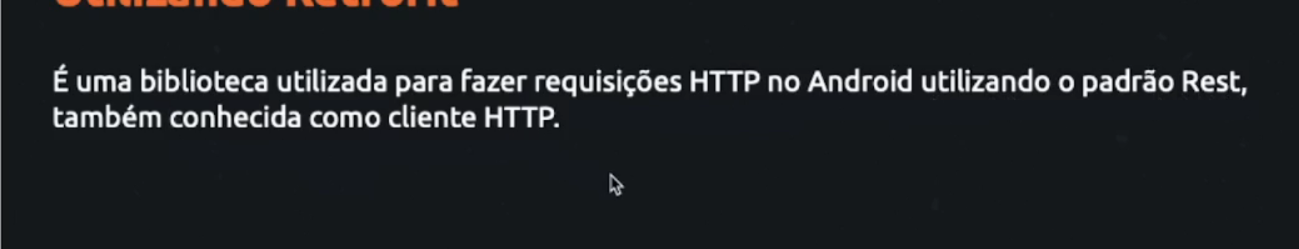
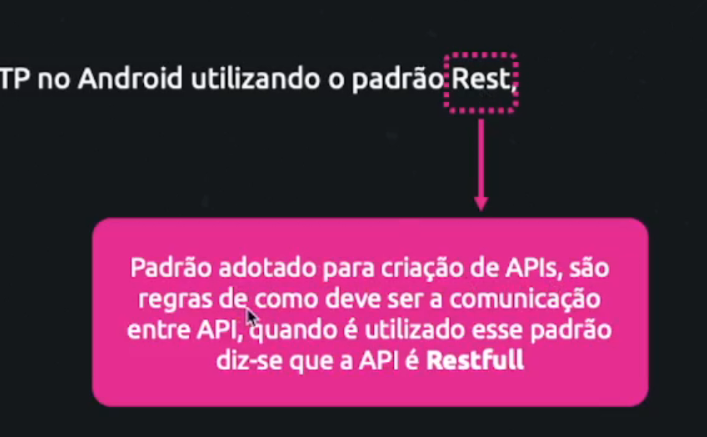
Oque é o retrofit?



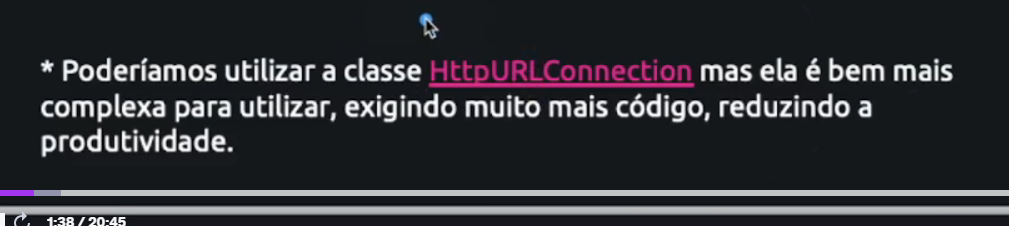




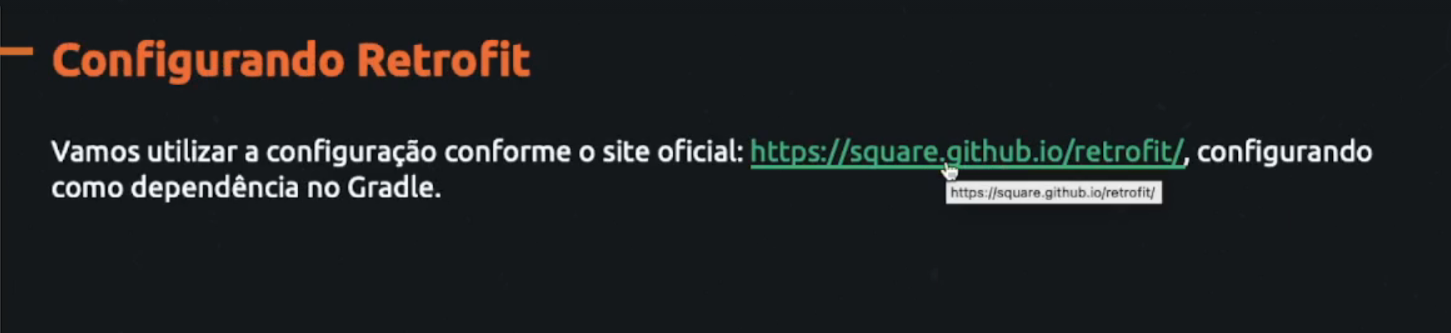
Rest = São as regras que devem ser utilizadas ao criar um api

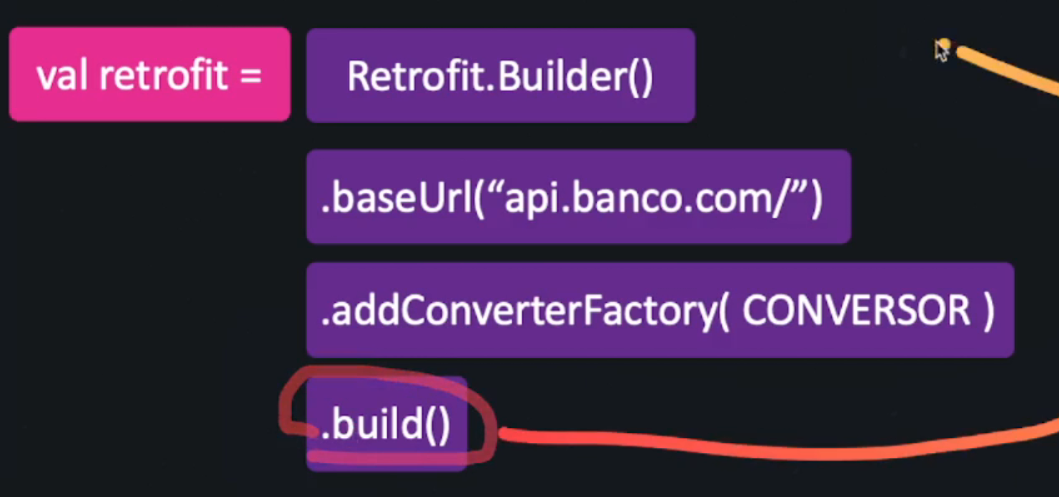
Restfull = É quando uma biblioteca ou api segue as regras.

Podemos também utilizar a classe httpURLConnection



# 1 configurando o Retrofit





Conversor = ele vai pegar o retorno JSON e vamos converter para OBJETOS

Build vai criar o objeto e esse objeto vai ficar dentro de val Retrofit.

## 1 implementar o gradle

implementação 'com.squareup.retrofit2:retrofit: *(inserir versão mais recente)* '

Usamos essa versão e ficou dessa forma

*implementation* ("com.squareup.retrofit2:retrofit:2.9.0")

## 2 implementando um conversor

Vamos utilizar o gson

// GSON  
*implementation*("com.squareup.retrofit2:converter-gson:2.9.0")

É recomendado usar sempre a mesma versão do Retrofit

## 3 class RetrofitHelper



Criamos um pacote e uma classe para organizar mais o código.

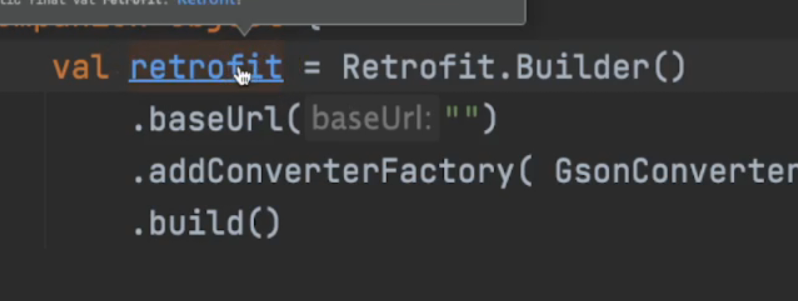
class RetrofitHelper {  
 // Queremos acessar os recursos sem precisar instanciar a classe  
 companion object {  
 val retrofit = Retrofit.Builder()  
 .baseUrl("")  
 .addConverterFactory(GsonConverterFactory.create()) // Conversor para JSON || XML  
 .build()  
 }  
}

## 4 instanciando a classe no MainActivity

private val retrofit by *lazy* **{** RetrofitHelper.retrofit  
**}**

Agora toda vez que precisar usar, só chamar a variável Retrofit, porque já esta instanciado a nossa classe dentro dela.

Lembrar que o Retrofit é o objeto



Onde esta sendo criado tudo dentro dele.

Agora quando quiser usar, podemos usar o Retrofit em qualquer local e vamos ter acesso a todos os recursos da biblioteca que estão disponíveis dentro dele.

