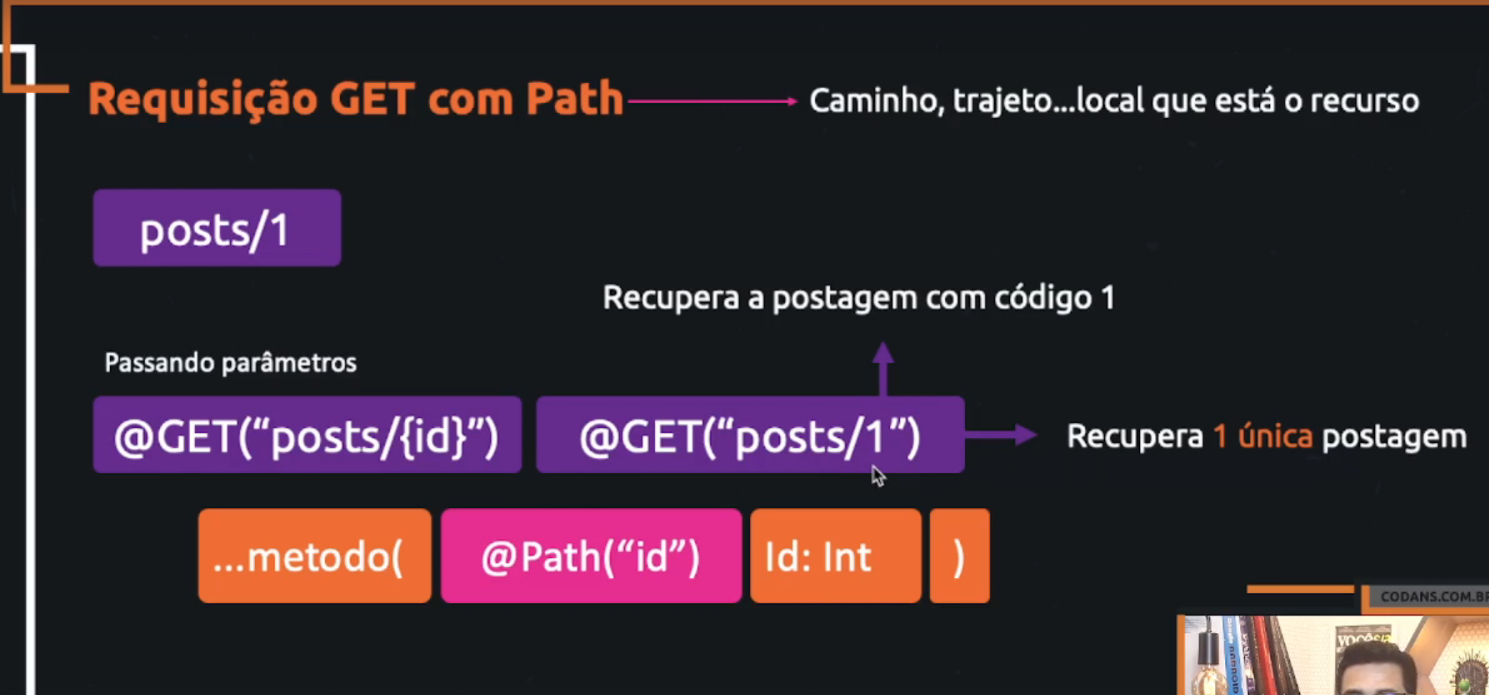
Nesse exemplo estamos recuperando apenas uma postagem, a postagem de código 1



# Exemplo recuperando apenas 1 postagem

## Interface

interface PostagemAPI {  
  
 @GET("posts")  
 suspend fun recuperarPostagens(): Response<List<Postagem>>  
  
  
 */\*\* itemProcurado = o parametro ID  
 \* id = id da postagem mesmo, como definimos la na classe dela  
 \* o itemProcurado pode ter qualquer nome, oque vale é o parametro que passamos no caso o Id  
 \*/* @GET("posts/{itemProcurado}")  
 suspend fun recuperarPostagemUnica(  
 @Path("itemProcurado") id: Int  
 ): Response<Postagem>  
}

## Fazendo o método

Passamos o ID, mas podemos fazer isso de diferentes maneiras, seja o um recyclerView, seja uma entrada de um usuário...

private suspend fun recuperarUmaPostagem() {  
 var retorno: Response<Postagem>? = null  
  
 try {  
 val postagemAPI = retrofit.create(PostagemAPI::class.*java*)  
 retorno = postagemAPI.recuperarPostagemUnica(1)  
 }catch (e: Exception) {  
 e.printStackTrace()  
 Log.i("info\_postagem", "ERRO: Não encontramos essa postagem.")  
 }  
  
 if (retorno !=null && retorno.*isSuccessful*) {  
 val postagemUnica = retorno.body()  
 val id = postagemUnica?.id  
 val titulo = postagemUnica?.title  
 val descricao = postagemUnica?.descricao  
  
 Log.i("info\_postagem", "ID: $id - Titulo: $titulo - Descrição: $descricao")  
 }  
}

## Atualizando resultado dentro de um textView

private suspend fun recuperarUmaPostagem() {  
 var retorno: Response<Postagem>? = null  
  
 try {  
 val postagemAPI = retrofit.create(PostagemAPI::class.*java*)  
 retorno = postagemAPI.recuperarPostagemUnica(1)  
 }catch (e: Exception) {  
 e.printStackTrace()  
 Log.i("info\_postagem", "ERRO: Não encontramos essa postagem.")  
 }  
  
 if (retorno !=null && retorno.*isSuccessful*) {  
 val postagemUnica = retorno.body()  
 val resultado = "${postagemUnica?.title} - ${postagemUnica?.descricao}"  
   
 withContext(Dispatchers.Main) **{** binding.textResultado.*text* = resultado  
 **}** Log.i("info\_postagem", resultado)  
 }  
}