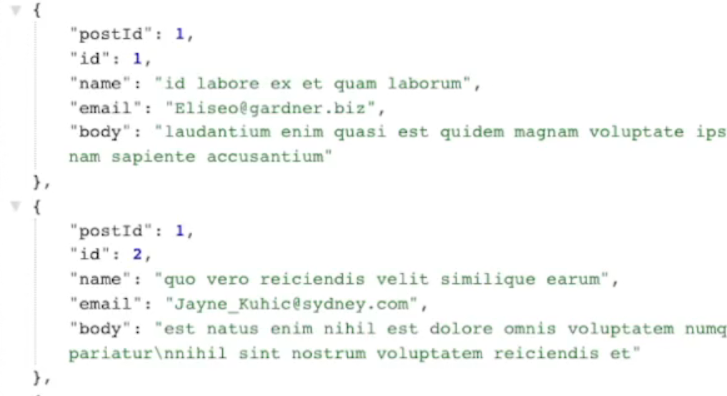
Nesse comentário vai retornar uma lista com vários comentários do usuário

Lembrar que estamos acessando uma única postagem e pegando os comentários dessa postagem.



## Criando uma interface

package com.allephnogueira.recuperandolista.api  
  
import com.allephnogueira.recuperandolista.model.Comentario  
import retrofit2.Response  
import retrofit2.http.GET  
import retrofit2.http.Path  
  
interface ComentarioAPI {  
  
 @GET("posts/{id}/comments")  
 suspend fun recuperarComentariosParaPostagem(@Path("id") id: Int) : Response<List<Comentario>>  
  
  
}

Nesse exemplo vamos passar o id do usuário.

## Criando a classe comentário para receber os dados

package com.allephnogueira.recuperandolista.model  
  
import com.google.gson.annotations.SerializedName  
  
data class Comentario(  
  
 @SerializedName("body")  
 val comentario: String,  
 val email: String,  
 val id: Int,  
 val name: String,  
 val postId: Int  
)

## Criando nosso método para processar os dados e exibir na tela

Lembrar de acumular os comentários em uma variavel

private suspend fun recuperarComentariosParaPostagem() {  
 var retorno : Response<List<Comentario>>? = null  
  
 try {  
 val ComentarioAPI = retrofit.create(ComentarioAPI::class.*java*)  
 retorno = ComentarioAPI.recuperarComentariosParaPostagem(1)  
 }catch (e : Exception) {  
 e.printStackTrace()  
 Log.i("info\_comentario", "ERRO: Comentario não localizado!")  
 }  
  
 if (retorno != null && retorno.*isSuccessful*) {  
 val listaDeComentario = retorno.body()  
 var comentariosJuntos = ""  
 listaDeComentario?.*forEach***{** val id = **it**.id  
 val nome = **it**.name  
 val email = **it**.email  
 val comentario = **it**.comentario  
  
  
 comentariosJuntos += "\nID: $id\n" +  
 "Nome: $nome\n" +  
 "E-mail: $email\n" +  
 "$comentario"  
  
 withContext(Dispatchers.Main) **{** binding.textComentario.*text* = comentariosJuntos  
 **}** Log.i("info\_comentarios", "$id - $comentario")  
  
 **}** }  
}

}  
}

