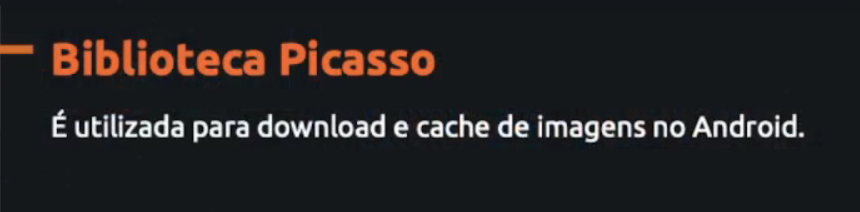
Pra que serve?

Serve para fazer download de imagem da WEB/API



Cache: Imagina que estamos usando o recyclerView com varias imagens, ai o usuário sai da tela e volta novamente.

Utilizando o cache essas imagens que já foram baixadas uma vez não vai precisar ser baixada novamente.

Pra utilizar

Get = recuperar

Load = carregar

Into = indicar onde vamos exibir essa imagem

Into = dentro de...

Resumindo ( pega a image, carrega a imagem, dentro de... ex: ImageViewFotoPerfil)



Site oficial da picaso: <https://square.github.io/picasso/>

# Pratica configurando Picaso

implementation 'com.squareup.picasso:picasso:2.8'

Adicionando no modulo do app

# Criando classe para recuperar fotos

**{**"albumId": **1**,"id": **1**,"title": "accusamus beatae ad facilis cum similique qui sunt","url": "<https://via.placeholder.com/600/92c952>","thumbnailUrl": "<https://via.placeholder.com/150/92c952>"**}**

**Tambem vamos criar uma interface para essas fotos**

interface FotosAPI {  
  
 @GET("photos/{id}")  
 suspend fun recuperarFoto(  
 @Path("id") id: Int  
 ) : Response<Fotos>  
  
}

Metodo para recuperar a URL da imagem.

private suspend fun postagemUnicaComFoto() {  
 var retorno : Response<Fotos>? = null  
  
 try {  
 val postagemComFotoAPI = retrofit.create(FotosAPI::class.*java*)  
 retorno = postagemComFotoAPI.recuperarFoto(1)  
 }catch (e: Exception) {  
 e.printStackTrace()  
 }  
  
 if(retorno != null && retorno.*isSuccessful*) {  
 val urlImagem = retorno.body()?.url  
  
  
 withContext(Dispatchers.Main) **{** binding.textResultadoFoto.*text* = urlImagem.*toString*()  
 **}** Log.i("url\_imagem", urlImagem.*toString*())  
 }else {  
 binding.textResultadoFoto.*text* = retorno?.code().*toString*()  
 }  
}

Agora que já recuperamos o URL vamos exibir essa imagem

Para isso vamos adicionar um ImageView na tela

# Colocando imagem recuperada dentro do ImageView

private suspend fun postagemUnicaComFoto() {  
 var retorno : Response<Fotos>? = null  
  
 try {  
 val postagemComFotoAPI = retrofit.create(FotosAPI::class.*java*)  
 retorno = postagemComFotoAPI.recuperarFoto(1)  
 }catch (e: Exception) {  
 e.printStackTrace()  
 }  
  
 if(retorno != null && retorno.*isSuccessful*) {  
 val urlImagem = retorno.body()?.url  
  
  
 withContext(Dispatchers.Main) **{** binding.textResultadoFoto.*text* = urlImagem.*toString*()  
  
 Picasso.get()  
 .load(urlImagem)  
 .into(binding.imageViewRecuperada)  
 **}**

Ali passamos a URL da imagem que recuperamos

E dizemos aonde vamos por ela, no caso estamos colocando ela dentro do imageView que criamos

Também estamos usando o Picaso dentro da Coroutine MAIN

Lembrar que somente ela pode atualizar interfaces.

Também podemos utilizar o picaso para carregar qualquer imagem, idependente de esta dentro da uma API ou Não

No exemplo a baixo eu peguei uma imagem do google, porque a imagem da API estava com problemas

withContext(Dispatchers.Main) **{** binding.textResultadoFoto.*text* = urlImagem.*toString*()  
  
 val urlImagemGoogle = "https://i.ytimg.com/vi/wDr4S69y6Uk/maxresdefault.jpg?sqp=-oaymwEmCIAKENAF8quKqQMa8AEB-AH-CYAC0AWKAgwIABABGHIgYSg2MA8=&rs=AOn4CLC4dZvrlF2aQuXA34mwoEtn\_QLZHQ"  
  
 Picasso.get()  
 .load(urlImagemGoogle)  
 .into(binding.imageViewRecuperada)  
**}**

