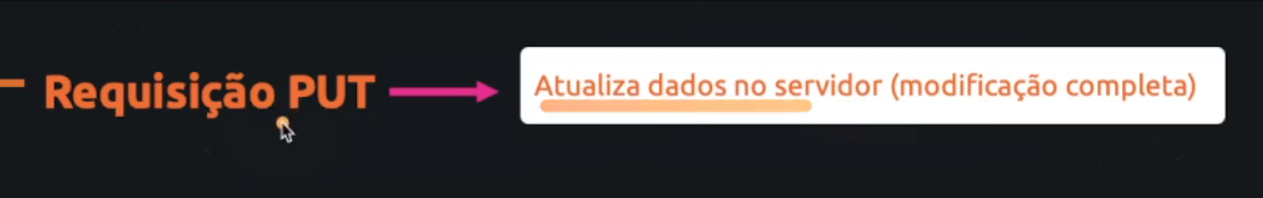
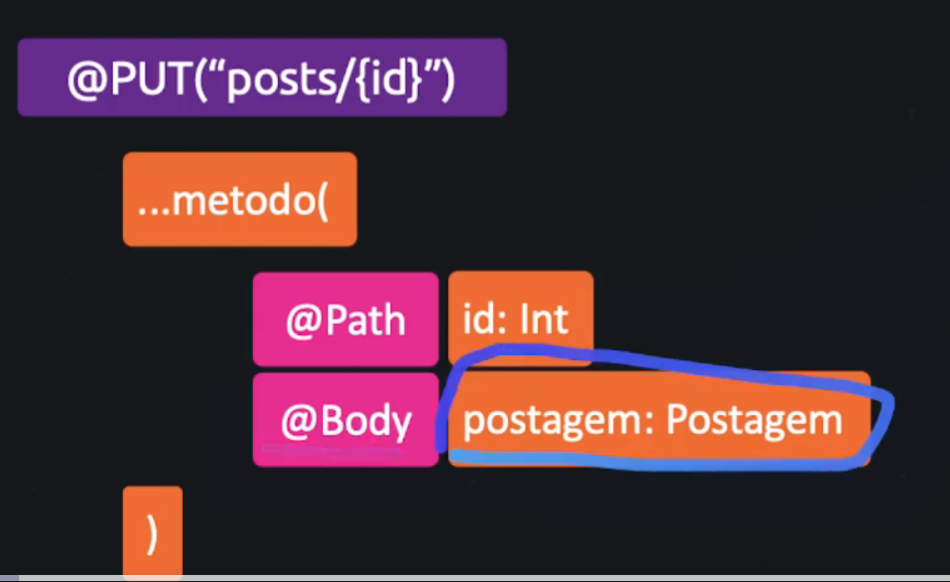
Oque são essas requisições





Vamos seguir a mesma pratica da aula anterior

O Body serve para atualizar o corpo

No caso de postagem o corpo seria os atributos que estão dentro de postagem.

# Mão na pratica, usando o PUT = Atualizando dados por completo

Vamos atualizar a postagem de numero {id}

Primeiro fizemos o método dentro da Interface

@PUT("posts/{id}")  
 suspend fun atualizarPostagem (  
 @Path("id") id: Int, @Body postagem: Postagem  
 ) : Response<Postagem>  
  
}

Agora vamos fazer o método dentro da MainActivity

private suspend fun atualizarPostagemMetodoPUT() {

// Response é a resposta HTTP

// É onde la em baixo vamos pegar os dados que vinheram da ROTA e vamos utilizar.  
 var retorno : Response<Postagem>? = null  
  
  
 /\* Imagina que aqui é onde vamos recuperar os dados que o usuário digitou da nova postagem \*/

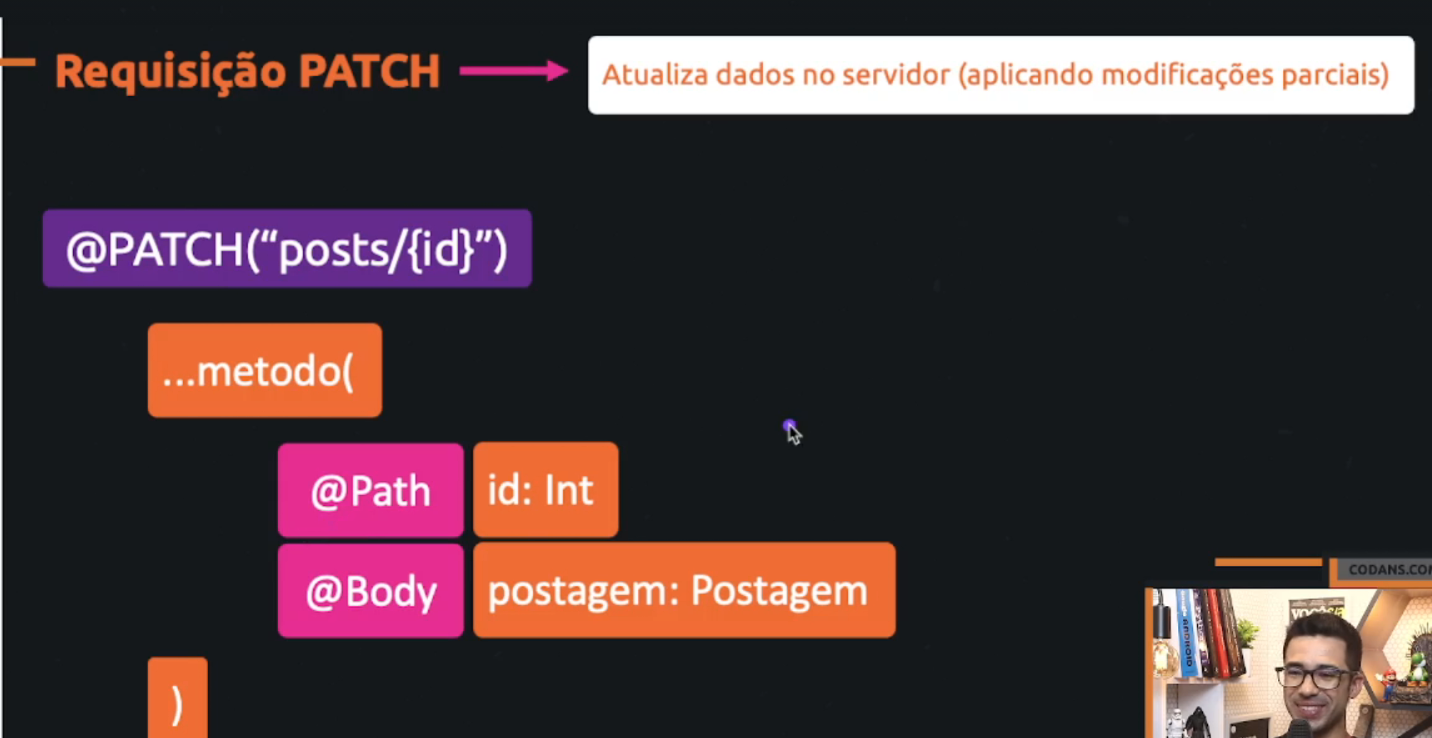
val postagem = Postagem(  
 "Atualizando postagem",  
 -1,  
 "Nova atualização",  
 -1  
 )  
  
  
 try {  
 **/\* Lembrar que o retrofit.create ele cria uma classe a partir da Interface  
 Ai podemos usar seu metodo como se fosse realmente uma classe.  
 \*/**  
 val postagemAPI = retrofit.create(PostagemAPI::class.*java*)  
 retorno = postagemAPI.atualizarPostagem(1, postagem)  
  
 }catch (e: Exception) {  
 e.printStackTrace()  
 Log.i("info\_atualizarPostagem", "ERRO: Não consegui atualizar sua postagem")  
 }  
  
  
 /\* Verificando se retorno esta tudo correto \*/  
  
 if (retorno != null && retorno.*isSuccessful*){  
 val postagem = retorno.body()  
 val id = postagem?.id  
 val titulo = postagem?.title  
 val descricao = postagem?.descricao  
  
 val descricaoCompleta = "ID: $id, Titulo: $titulo - Descrição: $descricao"  
  
 Log.i("info\_atualizarPostagem", descricaoCompleta)  
 }  
}

Nesse exemplo de cima poderíamos atualizar a postagem e receber um objeto do tipo Postagem e depois utilizar esses dados se for necessários.

# Metodo PATCH

Ele vai servi para atualizar os dados, mas somente os que você quer atualizar.

Lembrar do priston, onde baixávamos isso somente para alterar alguns dados.



Vamos supor que eu queira atualizar somente o titulo da postagem

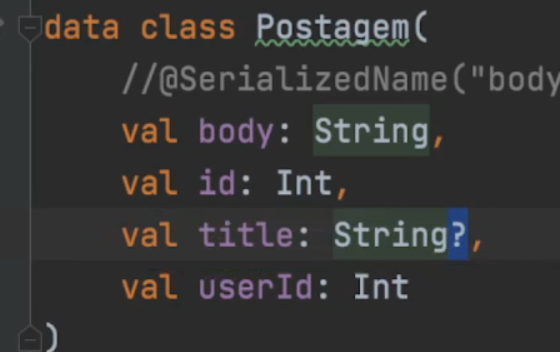
# Então mao na massa PATCH

Minha classe Postagem

Todos os itens estão dentro de um construtor, então todos os itens vai ser necessário, com isso vamos ter um problema, porque imagina que não queremos atualizar a descrição, somente o titulo, com isso vamos ser obrigados a passar também a descrição

## Resolvendo problema

Mas podemos fazer de uma forma que ele aceite valores nulos.



Agora dessa forma o titulo pode ser nulo, se caso o usuário não quiser atualizar ele

## Metodo para atualizar

Agora resolvemos o problema atualizamos somente o titulo e a descrição deixamos como nulo, assim ele segue o valor antigo

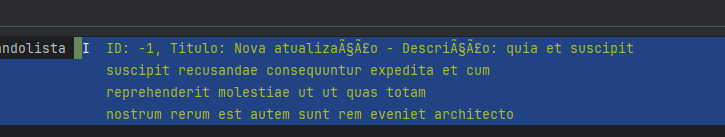
private suspend fun atualizarPostagemmetodoPATCH() {  
  
 var retorno : Response<Postagem>? = null  
  
  
 /\* Criando a postagem \*/  
 val postagem = Postagem(  
 null,  
 -1,  
 "Nova atualização",  
 -1  
 )  
  
  
 try {  
 /\* Lembrar que o retrofit.create ele cria uma classe a partir da Interface  
 Ai podemos usar seu metodo como se fosse realmente uma classe.  
 \*/  
 val postagemAPI = retrofit.create(PostagemAPI::class.*java*)  
 retorno = postagemAPI.atualizarPostagemPatch(1, postagem)  
  
 }catch (e: Exception) {  
 e.printStackTrace()  
 Log.i("info\_atualizarPostagem", "ERRO: Não consegui atualizar sua postagem")  
 }  
  
  
 /\* Verificando se retorno esta tudo correto \*/  
  
 if (retorno != null && retorno.*isSuccessful*){  
 val postagem = retorno.body()  
 val id = postagem?.id  
 val titulo = postagem?.title  
 val descricao = postagem?.descricao  
  
 val descricaoCompleta = "ID: $id, Titulo: $titulo - Descrição: $descricao"  
  
 Log.i("info\_atualizarPostagem", descricaoCompleta)  
 }  
}

Atenção esse Null só dá certo no PATCH se fosse no PUT ele iria atualizar todos os valores e colocaria NULL na descrição

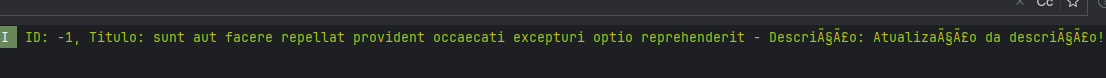
Já no PATCH ele entende que Null = nada, então ele não vai modificar nada ali.

Atualização:

Somente titulo atualizado.



Atualizando agora a descrição.



Não modificamos o titulo...