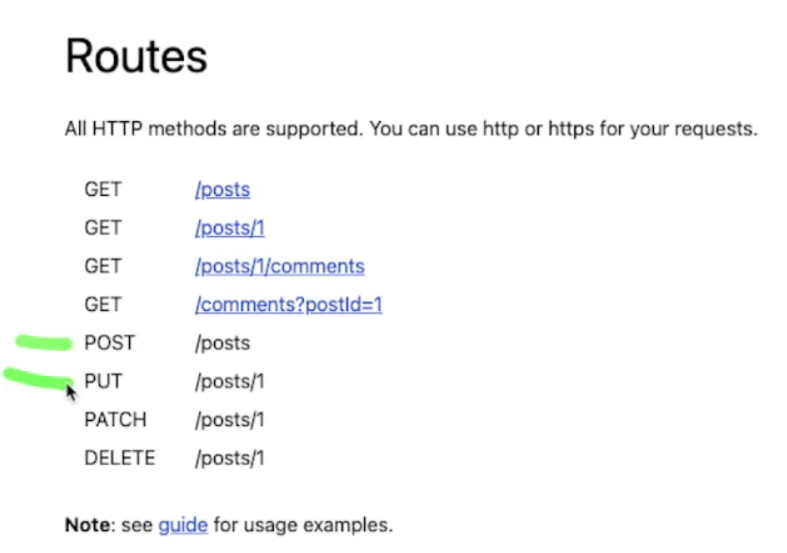
Vamos recuperar uma lista de itens para usar dentro do app

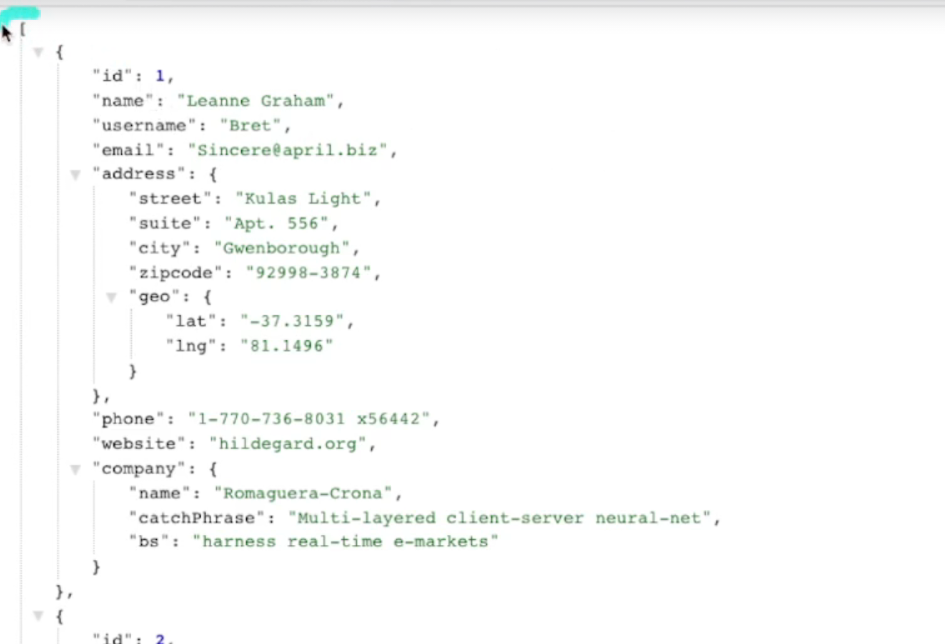
O passo a passo de configuração esta dentro do VIACEP

Vamos usar as rotas desse site

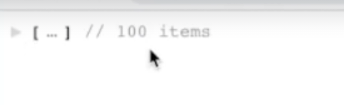
[JSONPlaceholder - Free Fake REST API](https://jsonplaceholder.typicode.com/)



Diferente da API Via CEP é que la so retornava um único objeto, nessa aqui vamos retornar uma lista.



Isso> [ significa que temos uma lista de itens.



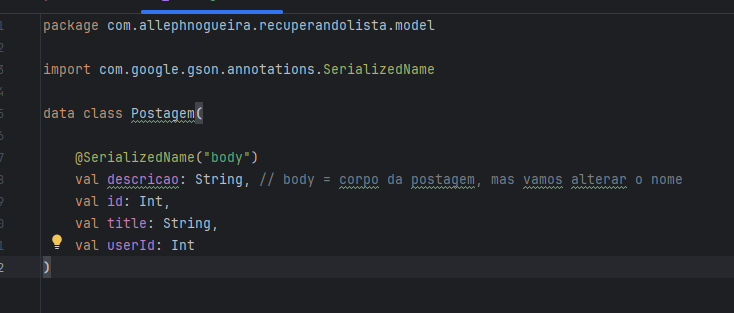
Mão na massa

# Criando a classe

Primeiro vamos selecionar os dados, reparar que paramos quando chegamos no ,

Isso indica que estamos pegando apenas o índice de numero 0

Vamos usar nosso plugin para gerar essa classe.



# Retornando a lista com os objetos

Lembra que essa api retorna uma lista com vários objetos, então precisamos configurar aqui.

import com.allephnogueira.recuperandolista.model.Postagem  
import retrofit2.Response  
import retrofit2.http.GET  
  
interface PostagemAPI {  
  
 @GET("posts")  
 suspend fun recuperarPostagens() : Response<**List**<Postagem>>  
}

Ele vai retornar uma lista de postagem e o próprio conversor que vai criar essa lista.

# Metodo para recuperar as postagens

binding.btnInicar.setOnClickListener **{** *CoroutineScope*(Dispatchers.IO).*launch* **{** recuperarPostagens()  
 **}  
 }** }

private suspend fun recuperarPostagens() {  
 /\* Agora estamos colocando dentro de retorno uma lista de objetos \*/  
 var retorno: Response<List<Postagem>>? = null  
 val postagemDigitada = ""  
  
 try {  
 val postagemAPI = retrofit.create(PostagemAPI::class.*java*)  
 retorno = postagemAPI.recuperarPostagens()  
  
 }catch (e: Exception){  
 e.printStackTrace()  
 Log.i("info\_postagens", "ERRO: Não consegui recuperar postagens!")  
 }  
  
 if (retorno != null){  
 if (retorno.*isSuccessful*) {  
  
 /\* Postagem agora é uma lista de Objetos, e vamos pegar item por item \*/  
  
 val listaPostagem = retorno.body()  
 listaPostagem?.*forEach***{** postagem **->** val userID = postagem.userId  
 val id = postagem.id  
 val titulo = postagem.title  
 val descricao = postagem.descricao  
  
 Log.i("info\_postagem", "SUCESSO: userID: $userID - idPostagem: $id - titulo: $titulo - descrição: $descricao")  
  
 **}** }  
 }  
 }  
}

# Colocando tudo dentro de uma variavel

Criando de outra forma para usar uma variável postagem em qualquer lugar do código

Dessa forma podemos usar a lista de postagens onde a gente quiser.

package com.allephnogueira.recuperandolista  
  
  
class MainActivity : AppCompatActivity() {  
  
 private val binding by *lazy* **{** ActivityMainBinding.inflate(*layoutInflater*) **}** private val retrofit by *lazy* **{** RetrofitHelper.retrofit  
 **}** private var postagem = *mutableListOf*<Postagem>()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 setContentView(binding.*root*)  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->** val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 **}** binding.btnInicar.setOnClickListener **{** *CoroutineScope*(Dispatchers.IO).launch **{** recuperarPostagens()  
  
 // Esperar pela recuperação das postagens  
 withContext(Dispatchers.Main) **{** if (postagem.*isNotEmpty*()) {  
 // Utilize a lista completa de postagens aqui  
 postagem.*forEach* **{** post **->** Log.i(  
 "info\_postagem",  
 "SUCESSO: userID: ${post.userId} - idPostagem: ${post.id} - titulo: ${post.title} - descrição: ${post.descricao}"  
 )  
 **}** }  
 **}  
 }  
 }** }  
  
 private suspend fun recuperarPostagens() {  
 var retorno: Response<List<Postagem>>? = null  
  
 try {  
 val postagemAPI = retrofit.create(PostagemAPI::class.*java*)  
 retorno = postagemAPI.recuperarPostagens()  
 } catch (e: Exception) {  
 e.printStackTrace()  
 Log.i("info\_postagens", "ERRO: Não consegui recuperar postagens!")  
 }  
  
 if (retorno != null && retorno.*isSuccessful*) {  
 val listaPostagem = retorno.body()  
 listaPostagem?.*let* **{** postagem.clear() // Limpar a lista se caso existir algo dentro  
 postagem.addAll(**it**) // Adicionar a lista inteira de postagens  
 **}** listaPostagem?.*forEach* **{** postagem **->** Log.i("info\_postagem", "SUCESSO: userID: ${postagem.userId} - idPostagem: ${postagem.id} - titulo: ${postagem.title} - descrição: ${postagem.descricao}")  
 **}** }  
 }  
}