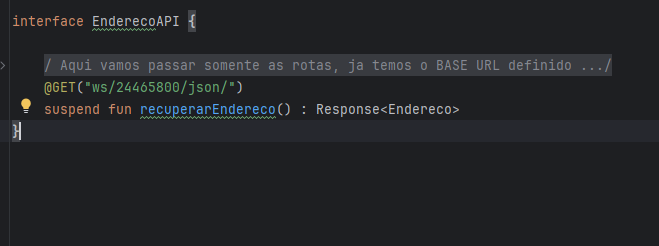
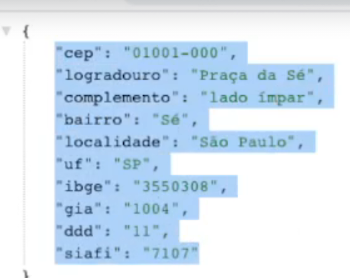
Lembrar da rota sem parâmetro apenas com endereço fixo



# Associação de JSON para Atributos da classe

Imagina que temos no JSON



Na nossa classe devemos ter o

Cep,

Endereco,

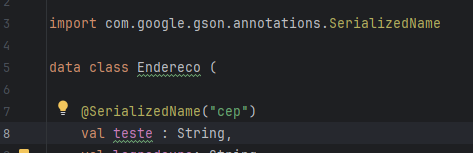
Numero,

Se tiver o nome diferente o conversor não consegue associar esses dados.

Imagina que no local do cep esteja com o nome de teste, então o conversor não vai conseguir associar e o teste vai ser nulo

Mas para resolver esse problema podemos utilizar o SerializadName

Ficando dessa forma

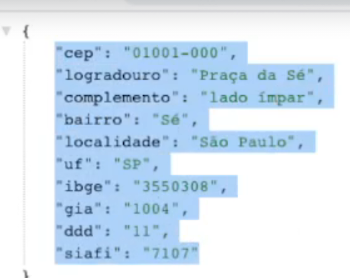


Ali podemos definir o nome que esta dentro do objeto JSON e dessa forma ele vai associar cep com teste.

Resumindo dessa forma CEP = Teste

# Plugins

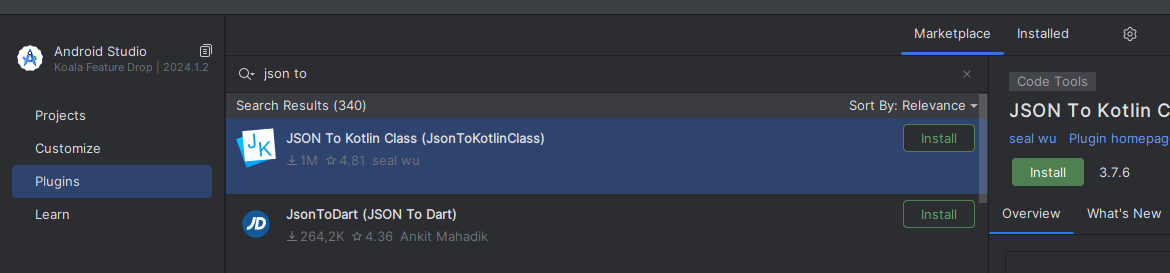
Imagina que estamos pegando dados da API e temos varias informações



Mas... é chato demais digitar cada um deles na nossa classe, então vamos fazer isso de forma automática.

Pra isso vamos na tela principal e vamos baixar esse plugin

JSON to



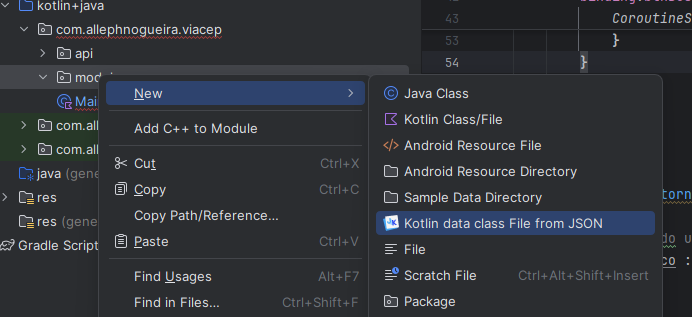
Esse plugin ele pega um texto json e converte de forma automática para uma classe.

## Como fazer

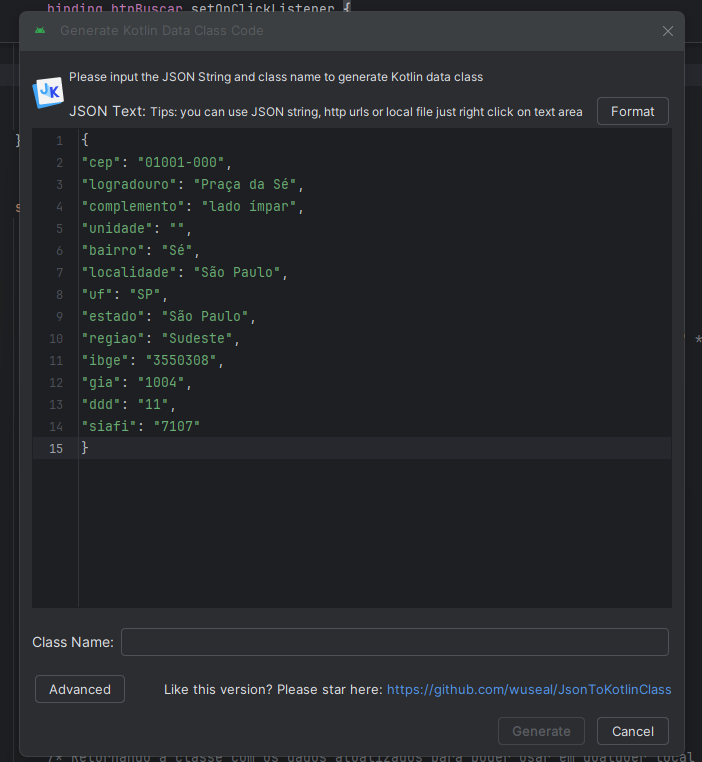
1 vamos copiar o JSON la do site

**{**"cep": "01001-000","logradouro": "Praça da Sé","complemento": "lado ímpar","unidade": "","bairro": "Sé","localidade": "São Paulo","uf": "SP","estado": "São Paulo","regiao": "Sudeste","ibge": "3550308","gia": "1004","ddd": "11","siafi": "7107"**}**

**2 Vamos selecionar essa opcao**



3 colar o código aqui dentro



Class Name: nome da classe

Depois clicar em Generate

Com isso vai gerar a classe já com os dados certinho e seus tipos.

# Rotas com CEP de forma dinâmica

Vamos simular como se o cep estivesse sendo digitado pelo usuário

Aqui estamos passando um parâmetro

private suspend fun recuperarEnderecoMetodo() {  
  
  
  
 /\* Retorno: estamos pegando o objeto convertido de JSON para nossa CLASSE  
  
 \*/  
 var retorno: Response<Endereco>? = null  
  
 val cepDigitadoPeloUsuario = "24465760"  
  
 try {  
 */\*\* Explicando o codigo  
 \* O metodo create ele faz assim:  
 \* Ele recebe a interface e ele mesmo cria e retorna para voce o objeto  
 \* Sem voce precisar de uma classe para usar o metodo  
 \* ai esse objeto permite voce acessa o metodo recuperarEndereco  
 \*/* val enderecoAPI = retrofit.create( EnderecoAPI::class.*java* )  
 retorno = enderecoAPI.recuperarEndereco(cepDigitadoPeloUsuario) // metodo dentro da INTERFACE

Agora precisamos la dentro de EnderecoAPI dizer que o método vai ter entrada de um parâmetro.

@GET("ws/{cep}/json/")  
 suspend fun recuperarEndereco(@Path("cep") cepDigitadoPeloUsuario: String) : Response<Endereco>  
}

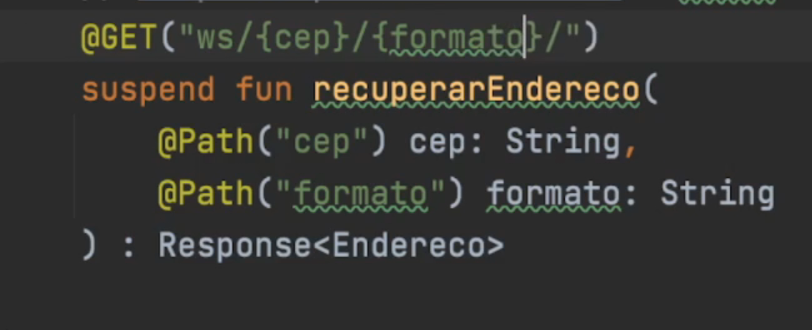
Atenção o nome do parâmetro cepDigitadoPeloUsuario estamos dizendo que ele é igual ao cep que tem na api.

E la em cima, precisamos passar o path

Ali indicamos que o cep vai ser colocado no caminho e quem faz isso é a própria API.

Lembrar que estamos buscando no link pelo CEP, por isso estamos usando CEP, se a gente busca pelo ID vamos ter que usar o ID, os nomes devem ser iguais.

Podemos também colocar mais parâmetros



Nesse exemplo a cima estamos escolhendo o formato (JSON, XML...)