Criamos esse aplicativo apenas para poder recuperar os dados de um CEP

# Configurando API

## 1 configurando a Retrofit

Vamos adicionar as duas dependências na mesma versão

*dependencies* **{** // RETROFIT  
 *implementation*("com.squareup.retrofit2:retrofit:2.9.0")  
 // GSON  
 *implementation*("com.squareup.retrofit2:converter-gson:2.9.0")

}

* 1. permissão para acessar a internet

<uses-permission android:name="android.permission.INTERNET"/>

## 2 pacote e classe Retrofit

Pacote api

Classe RetrofitHelper

package com.allephnogueira.viacep.api  
  
import retrofit2.Retrofit  
import retrofit2.converter.gson.GsonConverterFactory  
  
  
class RetrofitHelper {  
 companion object {  
 val retrofit = Retrofit.Builder()  
 .baseUrl("https://viacep.com.br/")  
 .addConverterFactory(GsonConverterFactory.create())  
 .build()  
 }  
}

## 3 adicionando viewBinding

private val binding by *lazy* **{** ActivityMainBinding.inflate(*layoutInflater*) **}**

setContentView(binding.*root*)

## 4 instanciando a classe

private val retrofit by *lazy* **{** RetrofitHelper.retrofit  
**}**

Assim vamos precisar usar somente a variável já configurada.

## 5 criando a date class com os dados da api

### Dados api:

**{**"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"**}**

### **Classe:**

package com.allephnogueira.viacep.model  
  
data class Endereco(  
 val cep: String,  
 val logradouro: String,  
 val complemento: String,  
 val unidade: String,  
 val bairro: String,  
 val localidade: String,  
 val uf: String,  
 val estado: String,  
 val regiao: String,  
 val ibge: Int,  
 val gia: Int,  
 val ddd: Int,  
 val siafi: Int  
)

## 6 configurando a interface/ ROTA:

BaseURL : https://viacep.com.br/

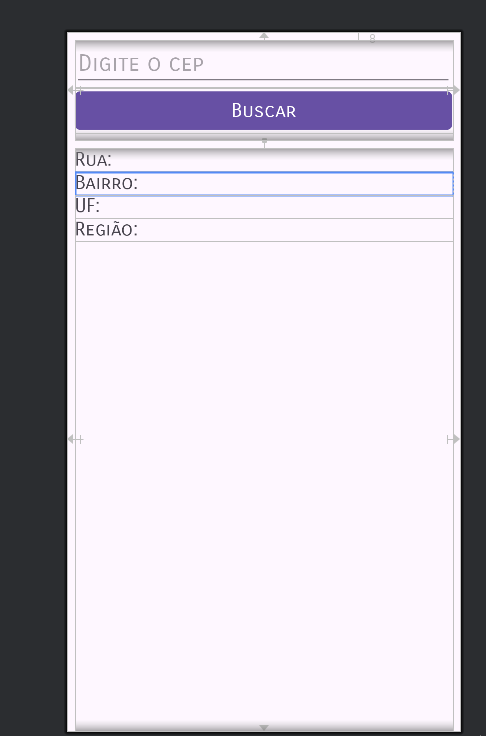
ROTA: ws/01001000/json/

package com.allephnogueira.viacep.api  
  
import com.allephnogueira.viacep.model.Endereco  
import retrofit2.Response  
import retrofit2.http.GET  
  
interface EnderecoAPI {  
  
 @GET("ws/01001000/json/")  
 suspend fun recuperarEndereco() : Response<Endereco>  
}

Response = Resposta

Aqui colocamos endereço porque o próprio Gson vai converter os dados de JSON para Objeto

## 7 fazendo a interface do app



## 7 metodo para retornar os dados

Criamos o método e vamos adicionar o try e catch porque não temos certeza se vamos conseguir recuperar esses dados

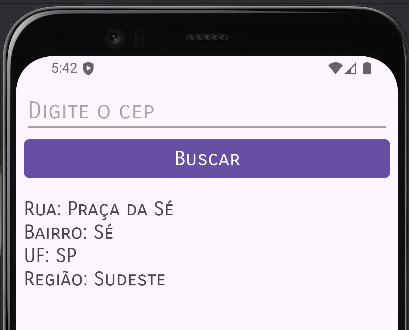
class MainActivity : AppCompatActivity() {  
  
 private val binding by *lazy* **{** ActivityMainBinding.inflate(*layoutInflater*) **}** private val retrofit by *lazy* **{** RetrofitHelper.retrofit  
 **}** private var retornoApi: Endereco? = null

suspend fun retornarDadosApi() {  
  
 /\* Retorno vai ser uma resposta do tipo Endereco  
 \* Que pode retornar ou não os dados \*/  
 var retorno : Response<Endereco>? = null  
  
 try {  
 val enderecoAPI = retrofit.create(EnderecoAPI::class.*java*)  
 retorno = enderecoAPI.recuperarEndereco()  
 }catch (e: Exception){  
 e.printStackTrace()  
 Log.i("info\_api", "Não conseguimos recuperar os dados")  
 }  
  
 /\* Verificando se os dados foi retornado para poder usar  
 \* Se retorno não for nulo e tiver sucesso  
 \* Dentro do body é onde vai estar todos os dados que ja foram convertidos.\*/  
 if (retorno != null && retorno.*isSuccessful*) {  
 /\* retornoApi é onde vai esta todos os dados ja em OBJETOS \*/  
 retornoApi = retorno.body()  
 }  
  
  
}

## 8 exibindo dados na tela

Atenção ainda não esta com o cep de forma dinâmica, apenas o CEP já esta configurado na rota, então o botão sempre vai pegar o CEP que esta já configurado na rota.

binding.btnBuscar.setOnClickListener **{** *CoroutineScope*(Dispatchers.IO).*launch* **{** retornarDadosApi()  
 withContext(Dispatchers.Main) **{** // Rua, Bairro, UF, Regiao  
 binding.textRua.*text* = "Rua: ${retornoApi?.logradouro}"  
 binding.textBairro.*text* = "Bairro: ${retornoApi?.bairro}"  
 binding.textUF.*text* = "UF: ${retornoApi?.uf}"  
 binding.textRegiao.*text* = "Região: ${retornoApi?.regiao}"  
  
 **}  
 }  
 }**}



# 6 buscando cep de maneira dinâmica

## 1 capturando o texto

Primeiro precisamos de uma variável para poder capturar o texto que o usuário digitou.

binding.btnBuscar.setOnClickListener **{** val cepBuscado = binding.editCep.*text*.toString()

Ela esta dentro do btnBuscar porque ela vai capturar o texto toda vez que o usuário clicar em buscar

## passando o texto como parâmetro

binding.btnBuscar.setOnClickListener **{** val cepBuscado = binding.editCep.*text*.toString()  
  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** endereco = retornarDadosApi(cepBuscado)

### 3 recebendo o texto como parâmetro

Estamos recebendo o texto como parâmetro no método que criamos para poder processar os dados

suspend fun retornarDadosApi(cepBuscado: String) : Endereco? {  
  
 /\* Iniciando uma variavel do tipo endereco \*/  
 var endereco : Endereco? = null  
 try {  
  
 /\* "Retrofit criando uma classe por baixo dos panos para recuperar endereco" \*/  
 val enderecoAPI = retrofit.create(EnderecoAPI::class.*java*)  
 val resposta = enderecoAPI.recuperarEndereco(cepBuscado)

Também estamos passando o texto para a nossa interface

## modificando interface para receber parâmetro

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

Repara o path deve ser o mesmo nome do atributo da nossa classe.

Podemos colocar uma virgula e colocar mais parâmetros, se for necessário.