

Retrofit

Após ter implementado Model, Network e Repository será necessário ter uma ViewModel. Nela iremos implementar um **MutableLiveData** do tipo que será nossa collection, no caso da API da marvel será uma **Set<Result>** pois a lista que iremos utilizar é do tipo **Result**, lembrando que Result é a representação de um quadrinho no json retornado pela marvel e gerado pelo plugin **JsonToKotlin**.

Também teremos o Repository na ViewModel para que possamos acessar diretamente a API através do método declarado nele. Por fim, escreva o método que irá trabalhar com uma Coroutine Scope, dentro dessa coroutine você atribui o retorno da API como valor da sua variável **MutableLiveData**.

```
val listMutableComic= MutableLiveData<Set<Comic>>()
```

```
private val repositoryMarvel = RepositoryMarvel()
```

```
fun getAllComics() = CoroutineScope(Dispatchers.IO).launch {  
    listMutableComic.postValue(repositoryMarvel.getComicService())  
}
```

Agora no arquivo da sua Activity onde já deverá ter implementado toda a configuração do recyclerview, adicione a inicialização da ViewModel para a Activity reconhecer a classe como sua view model. Lembrando que é preferível inicializar a viewModel preguiçosamente, utilizando **by lazy**. Ou seja, a propriedade só será inicializada no momento que for chamada a primeira vez.

```
private val viewModelMarvel by lazy {  
    ViewModelProviders.of(this).get(ViewModelMarvel::class.java)  
}
```

Agora usaremos os recursos da **viewModelMarvel** para requerer da API uma lista de quadrinhos. Primeiro chamamos o método **getAllComics()** através da viewModel. esse método irá acessar a API e mudar o valor do nosso **mutableLiveData**. Logo após, iremos acessar o mutableLiveData através da viewModel e adicionaremos um observador para que observe alguma alteração de valor nessa variável, logo que a API retorna essa lista checamos se não é nula e adicionamos ela ao adapter.

```
viewModelMarvel.getAllComics()  
  
viewModelMarvel.listMutableComic.observe(this, Observer {  
    it?.let { itChar -> // similar a if(it != null)  
        listComics.addAll(itChar)  
        recycler.adapter?.notifyDataSetChanged()  
    }  
})
```