



Faça como eu fiz: gerando um arquivo json

Agora é com você! Faça o mesmo procedimento que eu fiz na aula, escrevendo a sua lista de filmes em json.

Opinião do instrutor



Você vai precisar alterar a classe **PrincipalComBusca**, fazendo com que ela primeiramente leia vários filmes e armazene-os na lista. Para fazer isso, vamos inicializar uma variável chamada `busca` com uma string vazia e também já criar nossa lista de títulos antes do loop. Uma das sugestões é fazer o loop utilizando o `while`.

```
public class PrincipalCo
    public static void m
        Scanner leitura
        String busca = "
```

```
List<Titulo> tit

        while (!busca.eq
                //código omi
        }
    }
}
```

[COPIAR CÓDIGO](#)

Dentro desse *while*, caso o usuário digite `sair` vamos interromper as buscas. Por esse motivo, logo após a digitação do nome do filme, iremos comparar se foi digitada a palavra `sair`.

```
while (!busca.equalsIgno

        System.out.p
        busca = leit

        if(busca.equ
            break;
        }

        //código omi
```

```
}
```

[COPIAR CÓDIGO](#)

Após alimentar a variável

`endereco` concatenando a url base com o nome do título a buscar, faremos a requisição http, que está posicionada no código de exemplo dentro de um try, pois pode acontecer algum erro e ser lançada alguma exceção, como vimos em aula:

```
//código omitido
```

```
if(busca.equalsIgnor  
    break;  
}
```

```
String endereco = "h
```

```
try {  
    HttpClient clien  
    HttpRequest requ  
        .uri(URI.cr  
        .build());  
    HttpResponse<Str  
        .send(reques
```

```
String json = re
```

```
        //código omitido
    } catch (NumberFormatException
        System.out.print
        System.out.print
    } catch (IllegalArgu
        System.out.print
    } catch (ErroDeConve
        System.out.print
    }

}
```

//código omitido

COPIAR CÓDIGO

A nossa variável `json` está nesse momento com o conteúdo retornado no corpo da resposta, ou seja, os dados encontrados referente ao filme buscado.

O próximo passo é convertermos esse conteúdo textual em objeto do tipo `Titulo`, onde usaremos a biblioteca Gson. Antes de utilizá-la, vamos instanciar uma variável com a mesma:

```
//código omitido
```

```
List<Titulo> titulos  
Gson gson = new Gson  
    .setFiel  
    .create(
```

```
//código omitido
```

[COPIAR CÓDIGO](#)

Agora iremos utilizá-la na
transformação dos dados:

```
//código omitido
```

```
String json = respon  
TituloOmdb meuTitulo  
Titulo meuTitulo = n
```

```
//código omitido
```

[COPIAR CÓDIGO](#)

Por fim, incluiremos o título,
representado na variável
`meuTitulo` à nossa lista de
títulos, conforme abaixo:

```
//código omitido  
Titulo meuTitulo = n  
titulos.add(meuTitul
```

```
//código omitido
```

[COPIAR CÓDIGO](#)

Esse fluxo vai ocorrer até que o usuário digite a palavra `sair` ao ser perguntado sobre o filme que deseja buscar. Ao finalizar as buscas e sair do loop, basta que façamos a escrita do arquivo, conforme abaixo:

```
while (!busca.equals  
    //código omitido  
}
```

```
FileWriter escrita =  
escrita.write(gson.t  
escrita.close();
```

```
System.out.println("  
}
```

```
}
```

[COPIAR CÓDIGO](#)

Com isso, teremos um arquivo chamado `filmes.json` com todos os nossos filmes buscados. Pode ser que esse arquivo fique num formato estranho, com todos os dados na mesma linha. Conforme vimos na aula, o Gson permite que a gente faça essa escrita com uma formatação mais amigável. Para isso basta incluir o código abaixo, informando que desejamos o *"pretty printing"*.

```
Gson gson = new GsonBuilder()
    .setFieldNamingStrategy(FieldNamingStrategy.LOWER_CASE_WITH_UNDERSCORES)
    .setPrettyPrinting()
    .create();
```

[COPIAR CÓDIGO](#)

Com isso já conseguimos armazenar nossos filmes preferidos num arquivo json, que poderá ser lido por outras aplicações e futuramente persistidos num banco de dados. Espero que tenha gostado dessa atividade!

