



Guilherme Bedin <guilherme0bedin@gmail.com>

#7DaysOfCode - Java 2/7: 🧑 Trábalhar com JSON, Strings e Regex

1 mensagem

Paulo Silveira <paulo@7daysofcode.io>

8 de junho de 2022 08:03

Responder a: reply-caelum57945.activehosted.1326.8167.46734@d32a.emsend3.com

Para: Guilherme Bedin <guilherme0bedin@gmail.com>



E aí, Guilherme Bedin, conseguiu desenvolver o desafio do primeiro dia e já está tudo pronto para o próximo?

Só pra garantir, vou deixar durante esses primeiros 2 dias a minha resolução lá no final deste e-mail, beleza? Assim você pode comparar com a sua, mas não esqueça de **compartilhar comigo as suas resoluções, quero ver também!**

Hoje é o segundo dia do seu desafio **#7DaysOfCode** com Java, bora praticar?

No primeiro dia, você fez uma chamada para a API do IMDB para receber o JSON dos top 250 filmes como resposta. Você deve ter recebido algo como:

```
{"items": [
  {"id": "tt5491994", "rank": "1", "title": "Planet
Earth II", "fullTitle": "Planet Earth II
(2016)", "year": "2016", "image": "...", ...},
  {"id": "tt0903747", "rank": "2", "title": "Breaking
```

```
Bad", "fullTitle": "Breaking Bad  
(2008)", "year": "2008", "image": "...." ...},  
...  
], "errorMessage": ""}
```

A sua tarefa de hoje será **parsear essa resposta**. Em outras palavras, você vai **extrair as informações desse JSON**. Repare que o JSON possui um array de filmes, e cada filme possui vários atributos como id, título, etc.

Trabalhar com dados em JSON é algo muito comum no dia a dia para uma pessoa desenvolvedora Java e, por isso, **existem várias bibliotecas para tal**. Essas bibliotecas abstraem todos os detalhes, como ler e extrair informações.

No entanto, nesse desafio, a ideia é praticar os fundamentos do Java! Ou seja, **trabalhar com as principais bibliotecas incluídas no JRE**, como a classe `String` e as famosas Expressões Regulares.

Sua tarefa será **extrair o título do filme e a URL da imagem a partir da resposta JSON**.

Existem várias maneiras de fazer isso e, neste momento, não se preocupe ainda em escrever um código elegante. Tente usar os métodos da classe `java.lang.String` como `substring()`, `split()` e `replace()`. Você também pode usar `Regex` (através das classes `Matcher` e `Pattern` do pacote `java.util.regex`) para encontrar uma string que siga um determinado padrão.

Com o resultado do parseamento, você deverá **criar diferentes listas, cada uma com um atributo do filme**. Uma lista com os títulos, outra com a URL da imagem e assim por diante. Exemplo:

```
List<String> titles = //parseia o título de
```

cada filme do JSON;

```
List<String> urlImages = //parseia a URL do  
pôster de cada filme do JSON;
```

```
// outras listas, com os anos (year) e as  
notas (imDbRating)
```

DICA

Não se assuste e vá por partes, dividindo a tarefa maior em sub-tarefas que possam ser implementadas através de métodos. Por exemplo, você pode primeiro extrair o JSON inteiro dos filmes, ou seja, buscar na resposta JSON tudo o que estiver dentro dos colchetes (“[” , “]”) e guardar esse JSON.

Com ele em mãos, você pode usar o método `split()` para separar cada filme:

```
Posição 1: {"id":"tt5491994","rank":"1", "  
title":"Planet Earth II",, .... , "image":"http  
..."} }
```

```
Posição 2: {"id":"tt0903747","rank":"2", "  
title":"Breaking Bad", ..., "image":"http ..."} }
```

Agora, com esse array disponível, você pode tratar cada filme separadamente para extrair os atributos que você quiser. Basta fazer a quebra do JSON de cada filme nas vírgulas para ter acesso aos atributos:

```
String [] atributos = jsonFilmes.split("\\", "\\");
```

E a partir daí, “só” vai faltar pegar o atributo na posição correta usando métodos da classe `String` e guardá-lo em uma lista.

Como resultado final, você deverá ter no seu método main algo como:

```
public static void main(String[] args) throws
Exception {

    String json = //chamada da API omitida
    String[] moviesArray = parseJsonMovies(json);

    List<String> titles =
parseTitles(moviesArray);
    titles.forEach(System.out::println);

    List<String> urlImages =
parseUrlImages(moviesArray);
    urlImages.forEach(System.out::println);

    //outras listas para rating e years
}
```

EXTRA

Como esse trabalho é bem de baixo nível, existem as famosas bibliotecas de *Binding* que possibilitam ler ou criar um JSON ou XML a partir do código Java sem ter que trabalhar com Regex e métodos da classe `String`.

Para trabalhar com JSON temos duas bibliotecas famosas:

- Jackson
- GSON

Fique à vontade para praticar com alguma dessas bibliotecas.

RESPOSTA DO DESAFIO DO DIA 1

- Solução do Dia 1

Lembre-se de compartilhar o seu código no seu GitHub e nas suas redes sociais com a hashtag **#7DaysOfCode**, e também com **#feedback7DoC** caso você queira alguma ajuda.

Bons estudos, até amanhã!

Paulo Silveira

CEO e fundador da Alura



Enviado para: guilherme0bedin@gmail.com

[Cancelar a inscrição](#)

Alura, Rua Vergueiro 3185, São Paulo - SP, 04101-300, Brasil