
Python com JSON

— Luiz Felipe Cavaleiro —

Ferramentas utilizadas

- Sistema Operacional: **Linux Mint 20 Ulyana**;
- Linguagem: **Python**;
- Formatação: **JSON**;
- Editor de código-fonte: **VSCode**;
- Validador JSON Schema: **JSON Schema Validator**;
- Bibliotecas utilizadas:
 - **xmltodict**: `sudo apt install python3-xmltodict`
 - **json (padrão)**
 - **Beautiful Soup (padrão)**

Tópicos

- 1) Transformar o arquivo de filmes em XML para JSON, sem perder conteúdo;
- 2) Validar o JSON criado com JSON Schema correspondente;
- 3) Aplicar consultas em JSON;
- 4) Aplicar transformações em JSON;

1. Transformar arquivo em XML para JSON

- JSON:

JavaScript Object Notation, é um formato utilizado para estruturar dados em formato de texto, sua simplicidade e independência pode ser amplamente utilizada em diferentes linguagens de programação e sistemas, para transmissão de dados através de APIs.

1. Transformar arquivo em XML para JSON

- JSON:

Os dados contidos em um arquivo no formato JSON devem ser estruturados por meio de uma coleção de pares com atributo e valor ou ser uma lista ordenada de valores. Seus elementos devem conter:

chave ou atributo: corresponde ao identificador do conteúdo. Por isso, deve ser uma string delimitada por aspas;

valor: representa o conteúdo correspondente e pode conter os seguintes tipos de dados: string, array, object, number, boolean ou null.

1. Transformar arquivo em XML para JSON

- JSON:

Além da terminação .json em todos os arquivos que utilizam esse formato, os dados armazenados devem seguir uma notação específica, ou seja, precisam ser organizados com os seguintes elementos básicos:

chaves “{ }” para delimitar os objetos e obrigatórias para iniciar e encerrar o conteúdo;

colchetes “[]” para indicar um array;

dois pontos “:” para separar a chave e seu valor correspondente;

vírgula “,” para indicar a separação entre os elementos.

1. Transformar arquivo em XML para JSON

O arquivo **xmlToJson.py** irá pegar outro arquivo, chamado **GioMovies.xtm** (formato XML) e com base nele criar o arquivo **GioMovies.json**, que conterà todos os dados presentes no XML, porém, agora no formato JSON, para isso execute o seguinte comando no terminal dentro do diretório que contém os arquivos:

```
python3 xmlToJson.py
```

1. Transformar arquivo em XML para JSON

- JSON é um formato de arquivo usado para representar e armazenar dados, enquanto um dicionário em Python é a estrutura de dados de fato (o objeto) que é mantida na memória enquanto um programa em Python é executado.
- Quando trabalhamos com arquivos JSON em Python, não podemos simplesmente ler e usar os dados em nosso programa diretamente. Isso ocorre porque o arquivo inteiro seria representado como uma única string e não conseguiríamos acessar os valores de chave-valor de modo individual.

1. Transformar arquivo em XML para JSON

- Então, usa-se os pares chave-valor do arquivo JSON para criar um dicionário em Python para que seja possível usar no programa para ler os dados, usá-los e modificá-los (se necessário).
- Essa é a conexão principal entre o JSON e os dicionários em Python. O JSON é a representação em formato string dos dados e os dicionários são as estruturas de dados que estão de fato na memória e que são criadas quando o programa é executado.

1. Transformar arquivo em XML para JSON

- Haverá vezes, no entanto, em que poderemos precisar fazer exatamente o contrário, criar uma string em formato JSON a partir de um objeto (por exemplo, um dicionário) para imprimi-lo, exibi-lo, armazená-lo ou trabalhar com ele como uma string.
- Para fazer isso, podemos usar a função `dumps` do módulo `json`, passando o objeto como argumento:

`json.dumps(<obj>)`

2. Validar o JSON criado com JSON Schema correspondente

- Para validar, o JSON Schema criado, será utilizada a ferramenta online e gratuita **JSON Schema Validator** que está disponível no seguinte endereço:

<https://www.jsonschemavalidator.net/>

- Para usar, basta copiar o conteúdo do JSON Schema e do arquivo JSON que contém o conteúdo, cada um em seu respectivo campo;

3. Aplicar consultas em JSON

Devem ser aplicadas as seguintes consultas ao documento:

- a) Quais são os tipos de gênero de filmes, sem repetição?
- b) Quais são os títulos dos filmes que foram produzidos em 2000, ordenados alfabeticamente?
- c) Quais são os títulos em inglês dos filmes que tem a palavra “especial” na sinopse?
- d) Quais são os sites dos filmes que são do tipo “thriller”?
- e) Quantos filmes contém mais de 3 atores como elenco de apoio?
- f) Quais são os ID dos filmes que tem o nome de algum membro do elenco citado na sinopse?

3. Aplicar consultas em JSON

Para realizar as consultas solicitadas, basta executar o arquivo **consultasGioMovies.py** no terminal, será mostrada a consulta realizada e seu resultado, para isso é necessário digitar o seguinte comando:

```
python3 consultasGioMovies.py
```

4. Aplicar transformações em JSON

Para gerar o conjunto de páginas HTML contendo a lista de filmes e suas respectivas informações, basta executar o arquivo **geradorHTML.py**, será gerado o arquivo **index.html** na raiz do diretório e também uma página para cada filme dentro do diretório **paginas**, os arquivos HTML tem como base um template que está disponível na internet, esse arquivo chama-se 'template.html' e sua estrutura serve como base para encontrar e inserir tags nas páginas, utilizando Python.

4. Aplicar transformações em JSON

Para executar o arquivo, digite no terminal:

```
python3 geradorHTML.py
```

Agora que as páginas já foram geradas, para "navegar" por ela, basta abrir com o navegador o arquivo **index.html**.