

Atividade T3 - ECM251

"O banco de dados CLI de Animes e Mangas"

Prof. Murilo Zanini de Carvalho

Setembro|2020

Descrição:

Descobrir qual o seu anime favorito, quais são as novas tendências e quais são os mangás que estão sendo lançados sempre foi um desafio. Diversas fontes não confiáveis e que espalham software mal intencionado sempre estão presentes nestes meios.

Tendo em vista este cenário nada amigável, você e um colega (como pode ser feito individual, pode ser só sua consciência mesmo), decidiram agir. Vocês pesquisaram e viram que o portal <https://jikan.moe/>, disponibiliza uma API que traz dados sobre animes e mangás. Ao ler sua documentação mais a fundo, vocês viram que ao fazer uma requisição do tipo:

```
https://api.jikan.moe/v3/search/anime?q=naruto
```

Vocês obtêm como resposta:

```
{
  "request_hash": "request:search:b5021fb5569664933c0ddceeda2cfd5ad745",
  "request_cached": true,
  "request_cache_expiry": 261022,
  "results": [
    {
      "mal_id": 20,
      "url": "https://myanimelist.net/anime/20/Naruto",
      "image_url": "https://cdn.myanimelist.net/images/anime/13/17405.jpg?s=59241469eb470604a792ad",
      "title": "Naruto",
      "airing": false,
      "synopsis": "Moments prior to Naruto Uzumaki's birth, a huge demon known as the Kyuubi, the Nine-T",
      "type": "TV",
      "episodes": 220,
      "score": 7.88,
      "start_date": "2002-10-03T00:00:00+00:00",
      "end_date": "2007-02-08T00:00:00+00:00",
      "members": 1555274,
      "rated": "PG-13"
    },
    {
      "mal_id": 1735,
      "url": "https://myanimelist.net/anime/1735/Naruto__Shippuuden",

```

Ainda sobre a documentação (<https://jikan.docs.apiary.io/#introduction>), vocês viram que a utilização da API, mesmo que sem a necessidade de chaves de acesso, era limitada a quantidade de requisições que vocês poderiam fazer. Para contornar este problema, tendo em vista que vocês desejam distribuir seu sistema com uma interface CLI (command line interface) para que todos pudessem utilizar, vocês decidem que o ideal seria criar um banco de dados local, utilizando SQLite mesmo para guardar os dados que vocês já tivessem buscado.

O seu banco precisaria de no mínimo 2 tabelas, uma para guardar os dados dos animes e outra para guardar os dados dos mangás. A construção destas tabelas deve ser realizada de acordo com as informações disponibilizadas na API.

Implementação:

Para testar seu sistema, você deve:

- Apresentar uma tela que permita ao usuário informar se ele deseja procurar um anime ou um mangá;
- Quando o usuário escolher a sua categoria, você deve ler o nome do título que ele deseja procurar, consultando primeiro em seu banco de dados local. Caso nenhum registro for encontrado localmente, pesquisar na API.
- Quando uma requisição for realizada na API, salvar os dados no banco.
- Os dados dos animes que devem ser armazenados em cada registro são:
 - URL do Poster do Anime;
 - Nome do Anime;
 - Sinópse;
 - Quantidade de episódios;
 - Nota.
- Os dados dos mangás que devem ser armazenados em cada registro são:
 - URL do Poster do Mangá;

- Nome do Mangá;
 - Sinótese;
 - Quantidade de capítulos;
 - Quantidade de volumes;
 - Tipo;
 - Nota.
- Atenção, os dados dos bancos, depois de salvos, são acessíveis para o usuário apenas com sua aplicação, não sendo necessário outras operações além de consulta e inserção de dados.
 - Possibilitar que o usuário possa ver todos os dados que estão cadastrados no sistema.

A sua classe main.java deve apenas iniciar o seu sistema e TODA interação com o usuário deve acontecer apenas na sua classe principal do sistema.

Utilize os padrões de projeto que já foram vistos em aula (DAO, Methods Extraction) para a construção da solução. A interface para utilização do sistema deve ser por linha de comando. Para a construção do arquivo de DB do banco de dados SQLite, utilizar: <https://sqliteonline.com/>.

Para utilização da biblioteca de parser de JSON, recomenda-se o uso de: <https://github.com/stleary/JSON-java>

README.md

JSON in Java [package org.json]

maven-central

v20200518

[Click here if you just want the latest release jar file.](#)

Overview

JSON is a light-weight language independent data interchange format.

Regras:

- A atividade pode ser realizada em duplas (apenas para deixar claro, uma dupla tem no máximo 2 pessoas).
- A cada nova classe ou interface que for criada, comitar ela no gerenciador de código que estiver sendo utilizado.
- Os nomes e os R.A.s da dupla devem estar na classe main.java do sistema.
- Utilizar a documentação do tipo Javadoc para todos os elementos que forem descritos no sistema.
- Notificar o professor que a atividade foi finalizada, enviando para ele uma mensagem na ferramenta de comunicação utilizada na disciplina (Slack). Apenas um dos membros da dupla deve fazer isso. Aguardar o professor confirmar que já clonou o repositório.
- A aula do dia 21/09/2020 fica reservada para o atendimento e retirada de dúvidas durante a elaboração do problema.
- Data de entrega: 04/10/2020

Boa Atividade!

Critérios de Avaliação Utilizados:

Critério	Peso
Pilar de abstração da Orientação a Objetos.	1
Pilar do encapsulamento da Orientação a Objetos.	1
Pilar da herança da Orientação a Objetos.	1
Pilar do polimorfismo da Orientação a Objetos.	1
Utilização correta das enumerações (enums).	2
Utilização correta das interfaces.	2
Modelos criados para representar os dados do sistema.	6
Implementação do padrão DAO conforme apresentado em sala de aula.	6
Utilização correta das estruturas de dados.	4
Documentação do sistema utilizando Javadocs.	2
Cadastro dos animes e mangás.	4
Consulta no banco local.	4
Atualização de dados cadastrados.	4
Recuperação de dados cadastrados.	4
Boas práticas de programação O.O..	2

Método de Entrega: Link para o repositório no Github com o projeto.

ATENÇÃO: Todos os personagens, lugares, incidentes, organizações e religiões retratados são fictícios.