

PROJETO BD

Simulação de Serviço de Streaming de Música
“Musify”

Guilherme Melo (PL2) – 2021217138

João Gaspar (PL2) – 2021221276

Tiago Almeida (PL1) – 2021221615

Índice

- Manual de Instalação
- Manual de Utilizador
- *Plano de Desenvolvimento*
- Informações Adicionais
- Conclusão

Manual de Instalação

Este manual fornece instruções detalhadas sobre como instalar e configurar a aplicação Flask com o uso do PostgreSQL e o cliente REST Postman.

■ Requisitos do Sistema:

- *Python 3.7 ou superior instalado*
- *Banco de dados PostgreSQL instalado e configurado*
- *Postman instalado*
- *Conexão à Internet*

Manual de Instalação

■ **PgAdmin4:**

- *Criar base de dados*
- *Copiar o SCRIPT enviado e correr*

■ **PyCharm:**

- *Instalar extensões com base nos imports (no nosso caso foi jwt e psycpg2)*
- *Dar run ao programa em Python*

■ **PostMan:**

- *Arrastar o ficheiro das API Request enviado*
- *Utilizar as funções descritas utilizando "SEND" (em alguns casos em que seja preciso dar login, é preciso colocar o token no header da função que utiliza o login)*

Manual de Utilizador

Autenticação:

A API de Música utiliza autenticação baseada em tokens JWT (JSON Web Token). Antes de começar a utilizar a API, é necessário obter um token válido. Siga os passos abaixo para autenticar e obter um token:

- Crie um novo utilizador:

- *Envie uma requisição POST para /user/ com as informações do utilizador no corpo da requisição. Certifique-se de incluir campos obrigatórios, como nome de utilizador, senha, e-mail, nome completo, etc.*

- Autentique-se:

- *Envie uma requisição PUT para /user/ com o nome de utilizador e senha no corpo da requisição.*
- *O token será retornado como resposta. Copie o token para usar nas requisições subsequentes.*

Manual de Utilizador

Utilizando a API de Música:

Agora que você obteve um token válido, você pode utilizar as funcionalidades da API de Música. Abaixo estão algumas das principais funcionalidades disponíveis:

- Consultar todas as músicas:

- *Envie uma requisição GET para /music/ para obter uma lista de todas as músicas cadastradas na API.*
- *Consultar detalhes de uma música específica:*
- *Envie uma requisição GET para /music/{id} substituindo {id} pelo ID da música desejada. Isso retornará os detalhes da música específica.*

Manual de Utilizador

- Adicionar uma nova música:

- *Envie uma requisição POST para /music/ com os detalhes da nova música no corpo da requisição. Certifique-se de incluir campos obrigatórios, como título, artista, gênero, etc.*

- Atualizar informações de uma música:

- *Envie uma requisição PUT para /music/{id} substituindo {id} pelo ID da música que deseja atualizar. Inclua as informações atualizadas no corpo da requisição.*

- Excluir uma música:

- *Envie uma requisição DELETE para /music/{id} substituindo {id} pelo ID da música que deseja excluir. Isso removerá a música da API.*

Manual de Utilizador

Pesquisa de Música:

A API de Música permite pesquisar músicas com base em diferentes critérios. Aqui estão alguns exemplos de como realizar pesquisas:

■ Pesquisar por título:

- *Envie uma requisição GET para `/music/search?title={termo}` substituindo `{termo}` pelo título da música que deseja pesquisar. Isso retornará uma lista de músicas correspondentes ao termo pesquisado no título.*

■ Pesquisar por artista:

- *Envie uma requisição GET para `/music/search?artist={termo}` substituindo `{termo}` pelo nome do artista que deseja pesquisar. Isso retornará uma lista de músicas correspondentes ao artista pesquisado.*

■ Pesquisar por género:

- *Envie uma requisição GET para `/music/search?genre={termo}` substituindo `{termo}` pelo género musical que deseja pesquisar. Isso retornará uma lista de músicas correspondentes ao género pesquisado.*

Manual de Utilizador

Manipulação de Listas de Reprodução:

A API de Música permite criar e gerir listas de reprodução personalizadas. Veja como utilizar essas funcionalidades:

- Criar uma lista de reprodução:
 - *Envie uma requisição POST para /playlist/ com os detalhes da nova lista de reprodução no corpo da requisição. Inclua campos obrigatórios, como nome da lista de reprodução e as músicas que deseja adicionar.*
- Consultar detalhes de uma lista de reprodução:
 - *Envie uma requisição GET para /playlist/{id} substituindo {id} pelo ID da lista de reprodução desejada. Isso retornará os detalhes da lista de reprodução, incluindo as músicas adicionadas.*
- Adicionar músicas a uma lista de reprodução:
 - *Envie uma requisição POST para /playlist/{id}/add substituindo {id} pelo ID da lista de reprodução à qual deseja adicionar músicas. Inclua os detalhes da música a ser adicionada no corpo da requisição.*
- Remover músicas de uma lista de reprodução:
 - *Envie uma requisição DELETE para /playlist/{id}/remove substituindo {id} pelo ID da lista de reprodução da qual deseja remover uma música. Inclua os detalhes da música a ser removida no corpo da requisição.*

Plano de Desenvolvimento

- 1- Especificação de Requisitos e Design
 - *Identifique todos os requisitos do sistema. Cada funcionalidade necessária deve ser bem documentada e deve incluir detalhes sobre o que o sistema deve fazer e como se comportará em diferentes cenários.*
 - *Comece a projetar o esquema do banco de dados. Identifique todas as entidades necessárias e suas relações. Decida os atributos que cada entidade deve ter.*
- 2- Desenvolvimento do Banco de Dados
 - *Desenvolva a estrutura do banco de dados usando um sistema de gestão de banco de dados (SGBD) SQL. Este passo incluirá a criação de tabelas para todas as entidades e a configuração de relações entre elas.*
 - *Garanta que a base de dados seja otimizada para consultas comuns que o sistema precisa fazer. Por exemplo, é provável que a plataforma precise frequentemente de encontrar todas as músicas de um determinado artista.*
- 3- Desenvolvimento da API REST
 - *Desenvolva a API REST que interage com o banco de dados. Assegure-se de que a API tem endpoints que correspondem a todas as funcionalidades necessárias.*
 - *Implemente a lógica do servidor para tratar dos pedidos HTTP. Cada pedido deve ser validado antes de interagir com o banco de dados.*
 - *Certifique-se de que todos os erros são tratados corretamente. O servidor deve devolver o código de status HTTP apropriado para cada situação.*

Plano de Desenvolvimento

■ 4 - Teste

- *Escreva testes para cada endpoint da API. Verifique se cada endpoint está a fazer o que é esperado e a devolver a resposta correta.*
- *Realize testes de carga para garantir que o sistema é capaz de lidar com uma grande quantidade de utilizadores ao mesmo tempo.*

■ 5 - Desenvolvimento de Documentação

- *Escreva documentação detalhada para a API. Esta documentação deve explicar como usar cada endpoint, incluindo o tipo de pedido HTTP, os parâmetros necessários e a resposta esperada.*

ER

