

Distributed Music Editor - Advanced Sound Systems

Overview

A firma Advanced Sound Systems (ASS) encontra-se a desenvolver uma aplicação de karaoke para músicos. Esta aplicação distingue-se por não só remover a voz das músicas, mas também remover instrumentos individuais, permitindo a um músico substituir a performance do músico original pela sua. Este novo serviço será disponibilizado online através de um portal web em que o músico pode fazer upload de um ficheiro de música, analisar os instrumentos que compõem a música, seleccionar vários instrumentos e finalmente receber um ficheiro novo em que a música contém apenas esses instrumentos.

A tarefa de separar uma música em várias pistas por instrumento é intensiva do ponto de vista de processamento pelo que a ASS contratou os alunos de Computação Distribuída para desenvolver um serviço online capaz de atender às necessidades da empresa com uma qualidade de experiência elevada para o utilizador final (identificação rápida dos instrumentos e construção do novo ficheiro sem o(s) instrumento(s) seleccionados).

Para esta tarefa deverão desenvolver todo o código necessário para a criação de um serviço Web que possa servir em paralelo múltiplos clientes de forma rápida e eficiente recorrendo a computação paralela (aqui conseguido através de paralelização em múltiplos processos/workers independentes que podem estar no mesmo computador ou não).

[Find out more about Swagger](#)

Tags

music

Processamento de Musicas

system

Informação sobre o sistema

Paths

GET /music Listar todas as músicas submetidas

Listar todas as músicas submetidas

Responses

Code	Description	Links
200	successful operation <i>Content</i> application/json	No Links

POST /music Adicionar nova música

Submeter ficheiros de áudio em formato MP3. De seguida o sistema deverá partir o ficheiro áudio em trechos de duração mais curta que podem ser analisados paralelamente. Em resposta à submissão do ficheiro de áudio, o serviço deverá retornar um ficheiro JSON contendo um identificador do trabalho (*music_id*), e uma lista de identificadores das pistas de instrumentos (cada identificador deverá corresponder a um instrumento) assim como um link para o ficheiro de cada instrumento (permitindo ao utilizador fazer o download e reprodução do ficheiro).

Responses

Code	Description	Links
200	successful operation <i>Content</i> application/json	No Links
405	Invalid input	No Links

GET /music/{music_id} Estado de processamento de uma musica

Requisitar o estado de processamento de uma música, quando o estado é 100% a resposta deve incluir o url do ficheiro final.

Parameters

Type	Name	Description	Schema
path	music_id <i>required</i>	ID da musica a procesar	integer (int64)

Responses

Code	Description	Links
200	successful operation <i>Content</i> application/json <i>Example</i> <div>43</div>	No Links
404	Music not found	No Links

POST /music/{music_id} Processar uma musica

Requisitar o processamento da música <music_id> com apenas os instrumentos listados através do seu identificador num ficheiro JSON submetido.

Parameters

Type	Name	Description	Schema
path	music_id <i>required</i>	ID da musica a procesar	integer (int64)

Responses

Code	Description	Links
200	successful operation	No Links
404	Music not found	No Links

GET /job Listar todos Jobs

Listar todos Jobs (Job é uma tarefa distribuída a um worker) submetidos

Responses

Code	Description	Links
200	successful operation <i>Content</i> application/json	No Links
405	Invalid input	No Links

GET /job/{job_id} Informação sobre um Job

estatísticas de um dado Job (tamanho do ficheiro, link para os vários ficheiros, tempos de processamento, número de worker requests, etc.)

Parameters

Type	Name	Description	Schema
path	job_id <i>required</i>	ID do JOB	integer (int64)

Responses

Code	Description	Links
200	successful operation <i>Content</i> application/json	No Links
404	Music not found	No Links

POST /reset Reiniciar o sistema

Apagar todos ficheiros temporários e músicas (sistema limpo), cancelar todos os workers ainda em funcionamento.

Responses

Code	Description	Links
200	successful operation	No Links

Components

Schemas

Musica

Properties

Name	Description	Schema
music_id <i>optional</i>		integer (int64)
name <i>optional</i>		string
band <i>optional</i>		string
tracks <i>optional</i>		< Track > array

Track

Properties

Name	Description	Schema
name <i>optional</i>		string
track_id <i>optional</i>		integer (int64)

Job

Properties

Name	Description	Schema
job_id <i>optional</i>		integer (int64)
size <i>optional</i>	in bytes	integer
time <i>optional</i>	in seconds	integer (timestamp)
music_id <i>optional</i>	Id da musica que deu origem a este job	integer (int64)
track_id <i>optional</i>		