



Récupération des données musicales par des techniques sur des données sémi-structurées



Jingwei ZUO
Qiuhao QIAN

Sommaire



01 | Introduction

- ✓ Environnement & Techniques
- ✓ Sources de données



02 | Structure globale

- ✓ Des données envisagées
- ✓ Structure de BD
- ✓ Structure de services web



03 | Implémentation

- ✓ Récupération des sources de données
- ✓ Conversion par XSLT et DOM
- ✓ Web service exposé

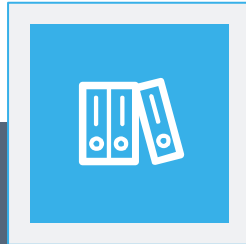


04 | Problèmes & Solutions

- ✓ Sources de données
- ✓ Conversion
- ✓ Manipulation de BD



05 | Conclusion



01 | Introduction

✓ **Environnement du travail**

✓ Environnement & Techniques

- Eclipse + Tomcat 7.0 + Axis2 (Web services), JRE1.8
- Postman(Test de REST requête)
- XSLT + DOM/SAX(Transition et extraction de données démi-structurées)
- Mysql(BD pour stocker des données envisagées)
- SOA
- SoapUI for Mac(Test de services Web générés)



Introduction



Structure



Implémentation



Problème



Conclusion



✓ Sources de données musicales

- MusicBrainZ
- Lastfm



Introduction



Structure



Implémentation



Problème



Conclusion

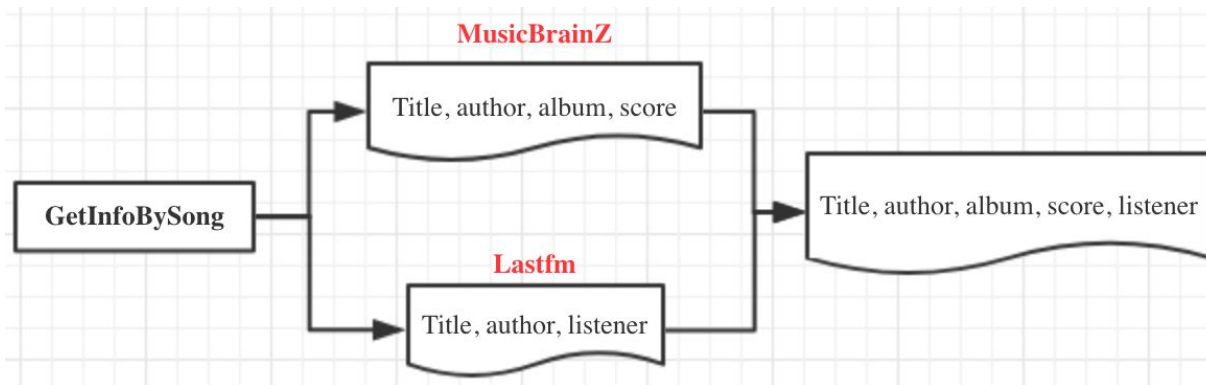


02 | Structure globale

- ✓ Des données envisagées
- ✓ Structure de BD
- ✓ Structure des services web

✓ Des données envisagées

- GetSongsByAuthor: Song title
- GetAlbumsByAuthor: Albums name
- GetInfoBySong: Title, author, album, score, listener



Introduction

Structure

Implémentation

Problème

Conclusion

✓ Structure de base de données

- Deux tables:
- Table1: Songs (Title, author, album, score_brainz, listeners_lastfm)
- Table 2: Albums(Title, author)

Songs
title: Varchar(100)
author: Varchar(100)
album: Varchar(100)
score_brainz: Varchar(100)
listeners_lastfm: Varchar(10)

Primary key(title, author)

Albums
title: Varchar(100)
author: Varchar(100)

Primary key(title, author)

Introduction

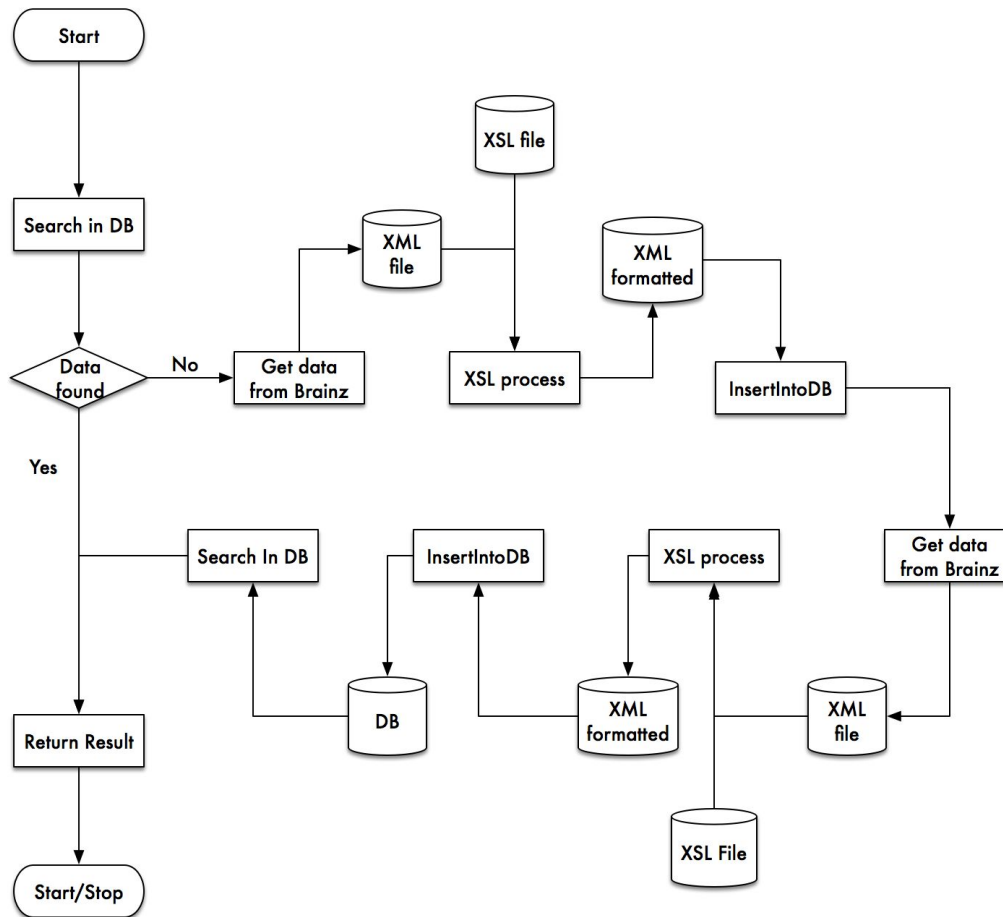
Structure

Implémentation

Problème

Conclusion

✓ Structure de processus



Introduction

Structure

Implémentation

Problème

Conclusion



03 | Implémentation

- ✓ **Récupération des sources de données**
- ✓ **Conversion par XSLT et DOM/SAX**
- ✓ **Web service exposé**

MusicBrainZ

- Source de musique plus complète
- REST API
- 3rd party libraries supporté de Java

Lastfm

- Sources de musique sont principalement copiées depuis MusicBrainZ
- REST API



Introduction



Structure



Implémentation



Problème



Conclusion



MusicBrainZ, REST API

Introduction

Structure

Implémentation

Problème

Conclusion

getSongsByAuthor:

[http://musicbrainz.org/ws/2/recording/?query=artistname:"Luca Barbarossa"](http://musicbrainz.org/ws/2/recording/?query=artistname:)

MBID, Title, author, album

getAlbumsByAuthor:

<http://musicbrainz.org/ws/2/release/?query=artist:Dave>

MBID, Title, author, status, language, aritst_name, data, country

getInfo:

[http://musicbrainz.org/ws/2/recording/?query=recording:"Sufriendo de amor"](http://musicbrainz.org/ws/2/recording/?query=recording:)

Title, length, author, album, score

Lastfm, REST API

Introduction

Structure

Implémentation

Problème

Conclusion

getSongsByAuthor: **artist_getTopAlbums(artist, api_key)**

http://ws.audioscrobbler.com/2.0/?method=artist.gettoptracks&artist=wason&api_key=fdc873135eefe53670783bd15d92eed5

Title, author

getAlbumsByAuthor: **artist.getTopTracks(artist, api_key)**

http://ws.audioscrobbler.com/2.0/?method=artist.gettopalbums&artist=cher&api_key=fdc873135eefe53670783bd15d92eed5

MBID, Title, author

getInfo: **track.search(track, api_key)** ou **Track.getInfo(track, artist, api_key)**

http://ws.audioscrobbler.com/2.0/?method=track.search&track=Believe&api_key=fdc873135eefe53670783bd15d92eed5 OU

http://ws.audioscrobbler.com/2.0/?method=track.getInfo&track=Believe&artist=xxx&api_key=fdc873135eefe53670783bd15d92eed5

Title, author, listener OU Title, author, listener, Album, play_numbers, etc.



Résultats exposé aux clients

getSongsByAuthor:

Title, author

getAlbumsByAuthor:

Title, author

getInfoBySong, fusionner des attributs depuis des deux sources

Title, author, album, score_brainz, listeners_lastfm

Introduction

Structure

Implémentation

Problème

Conclusion

Conversion par XSLT et DOM/SAX

getSongsByAuthor & GetInfoBySong

Node	Content
?? xml	version="1.0" encoding="utf-8"
Brainz_1	
xmlns:ext	http://musicbrainz.org/ns/ext#-2.0
xmlns:n	http://musicbrainz.org/ns/mmd-2.0#
song	
song_title	Believe
Artist	Ministers De-La-Funk
Album	2020
Score	100
listeners	
song	
song	

Primary key

Node	Content
?? xml	version="1.0" encoding="utf-8"
lastfm_3	
song	
song_title	Believer
Artist	Imagine Dragons
Album	
Score	
listeners	184571
song	
song	

Primary key

Conversion par XSLT et DOM/SAX

getAlbumsByAuthor:

Node	Content
 xml	version="1.0" encoding="utf-8"
▼  Brainz_2	
 xmlns:ext	http://musicbrainz.org/ns/ext#-2.0
 xmlns:n	http://musicbrainz.org/ns/mmd-2.0#
▼  Album	
 Album_title	Alma mía
 Artist	Wason
▶  Album	

Node	Content
 xml	version="1.0" encoding="utf-8"
▼  lastfm_2	
▼  album	
 Album_title	Alma Mía
 Artist	Wason
▶  album	

Introduction

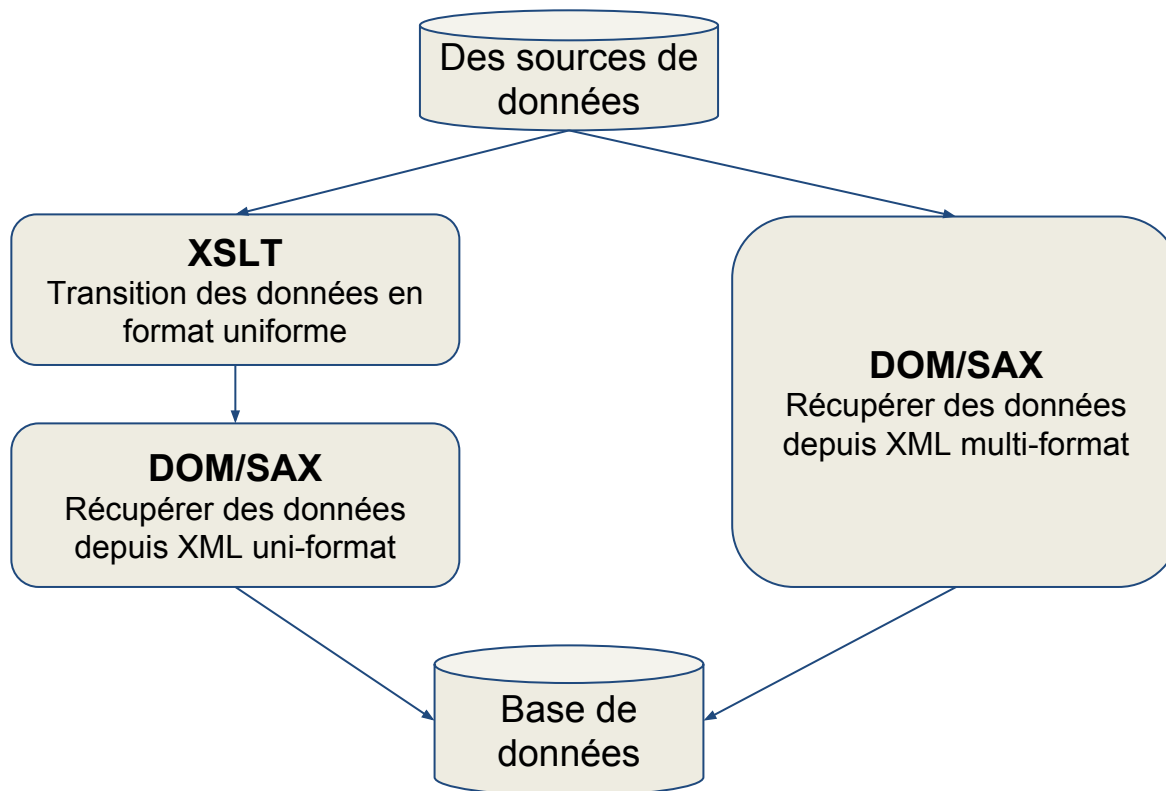
Structure

Implémentation

Problème

Conclusion

Conversion par XSLT et DOM/SAX





04 | Problèmes & Solutions

✓ Des attributs des sources sont différents

- ✓ -> Choisir des attributs en commun
- ✓ -> Fusionner des attributs

✓ Namespace de données venant de MusicBrainZ et Lastfm

- ✓ -> Dans notre script XSLT, il faut ajouter des namespaces

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:n = "http://musicbrainz.org/ns/mmd-2.0#" xmlns:ext="http://musicbrainz.org/ns/ext#-2.0">
```


```
<xsl:for-each select="n:recording">
  <song>
    <song_title>
      <xsl:value-of select="n:title"/>
    </song_title>
    <Artist>
      <xsl:value-of select="n:artist-credit/n:name-credit/n:artist/n:name"/>
    </Artist>
```

✓ Namespace OpenSearch dans le XML retourné par Lastfm

- ✓ -> Pré-traitement du fichier XML avant de faire la transition XSLT, pour enlever une partie de données(OpenSearch)

```
<?xml version="1.0" encoding="UTF-8" ?>
<lfm status="ok">
  <results>
    <opensearch:Query role="request" startPage="1"></opensearch:Query>
    <opensearch:totalResults>634211</opensearch:totalResults>
    <opensearch:startIndex>0</opensearch:startIndex>
    <opensearch:itemsPerPage>30</opensearch:itemsPerPage>
    <trackmatches>
      <track>
        <name>Believer</name>
        <artist>Imagine Dragons</artist>
        <url>https://www.last.fm/music/Imagine+Dragons/_/Believer</ur
        <streamable>FIXME</streamable>
        <listeners>184571</listeners>
        <image size="small">https://lastfm-img2.akamaized.net/i/u/34s
        <image size="medium">https://lastfm-img2.akamaized.net/i/u/64
        <image size="large">https://lastfm-img2.akamaized.net/i/u/174
        <image size="extralarge">https://lastfm-img2.akamaized.net/i/
        <mbid></mbid>
      </track>
    </trackmatches>
  </results>
</lfm>
```

✓ Web service client, connexion avec des services



DB_test.java *services.xml

```
1 <service name="GetAlbumByAuthor" >
2   <Description>
3     Please Type your service description here
4   </Description>
5   <messageReceivers>
6     <messageReceiver mep="http://www.w3.org/2004/08/wsd1/in-only"
7     <messageReceiver mep="http://www.w3.org/ns/2004/08/in-out"
8   </messageReceivers>
9   <parameter name="ServiceClass" locked="false">service.GetAlbumBy
10  </service>
```

La version de **messageReceiver** par défaut est trop ancienne. Il faut remplacer le lien par celui montré dans l'image au-dessous



DB_test.java services.xml

```
1 <service name="GetAlbumByAuthor" >
2   <Description>
3     Please Type your service description here
4   </Description>
5   <messageReceivers>
6     <messageReceiver mep="http://www.w3.org/ns/wsd1/in-only"
7     <messageReceiver mep="http://www.w3.org/ns/wsd1/in-out"
8   </messageReceivers>
9   <parameter name="ServiceClass" locked="false">service.GetAl
10  </service>
```



✓ Upload des fichiers sur le serveur de Web service

✓ -> Des fichiers XSLT sont stockés à local

Proposition de solutions:

1. Lire des URLs de nos scripts XSLT dans le code(e.g. Github)

www.github.com/XXX/transition.xslt

2. Ecriture en dur des scripts XSLT dans le code

-> Meilleure solution?



Introduction



Structure



Implémentation



Problèmes



Conclusion



05 | Conclusion

✓ Résultats

Introduction

Base de Cloud

Docker

Kubernetes

Conclusion

REST API

- Récupérer des données depuis MusicBrainZ et lastfm
- Pré-traitement sur des données avant de faire la transition

Transitions et extraction

- Des transitions sur des données vers un format unique par XSLT
- Extractions des données à insérer dans la DB par DOM/SAX



Résultats

- Analyse sur des données récupérées depuis le Web
- Définition des structure du processus. e.g fusionner des attributs, etc.

- 3 services répondant aux besoins sont implémenté et testé

Analyse sur des données

Exposer des 3 services web

Introduction

Base de Cloud

Docker

Kubernetes

Conclusion

Request 1

http://localhost:8080/Projet_test/services/GetSongsByAuthor.GetSongsByAuthorHttpSoap11Endpoint/

XML

Raw

```
<?xml version='1.0' encoding='utf-8'>
<env:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header/>
  <soapenv:Body>
    <ser:getSongsByAuthor>
      <!--Optional:-->
      <ser:artist>Wang</ser:artist>
    </ser:getSongsByAuthor>
  </soapenv:Body>
</env:Envelope>
```

XML

Raw

```
<?xml version='1.0' encoding='utf-8'>
<env:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getSongsByAuthorResponse xmlns:ns='http://service' xmlns:ax219='http://schemas.xmlsoap.org/soap/encoding/'>
      <ns:return>Back To Back 1</ns:return>
      <ns:return>Back To Back 2</ns:return>
      <ns:return>Back To Back 3</ns:return>
      <ns:return>Back To Back 4</ns:return>
      <ns:return>Back To Back 5</ns:return>
      <ns:return>Beat It</ns:return>
      <ns:return>Born2CreateFam</ns:return>
      <ns:return>Chase and Finale</ns:return>
      <ns:return>Chung - Dance Hall Days</ns:return>
      <ns:return>Come Together</ns:return>
      <ns:return>Come Together (Jeremy Juno, Daniel Beasley Remix)</ns:return>
      <ns:return>Dance Hall Days</ns:return>
      <ns:return>Dang-Doodle</ns:return>
      <ns:return>Dilpreet Dhillon | Parmish Verma | Latest Punjabi Song 2020</ns:return>
      <ns:return>Down Me</ns:return>
      <ns:return>Everybody Have Fun Tonight</ns:return>
      <ns:return>Eye of the Sun</ns:return>
      <ns:return>Free (Sick Elektrik rmx)</ns:return>
      <ns:return>Fuck Your Wobble</ns:return>
      <ns:return>Getting Ever So Slightly Better Pill (Dub)</ns:return>
      <ns:return>Getting Ever So Slightly Better Pt II (Dub)</ns:return>
      <ns:return>Getting Ever So Slightly Better Pt. Ii (Dub)</ns:return>
      <ns:return>Good Night</ns:return>
      <ns:return>Hundre sanger (m/Wang fra Fjorden Baby)</ns:return>
      <ns:return>Keep It Inside</ns:return>
      <ns:return>Love Lockdown Remix</ns:return>
      <ns:return>Memories</ns:return>
      <ns:return>Movin</ns:return>
      <ns:return>Music Of Life</ns:return>
    </ns:getSongsByAuthorResponse>
  </soapenv:Body>
</env:Envelope>
```

... H... Atta... ... JM... JMS ...

Headers (5) Attachments (0) SSL Info WSS (0) JMS (0)

response time: 194ms (2453 bytes) 1 : 1

Introduction

Base de Cloud

Docker

Kubernetes

Conclusion

Request 1

http://localhost:8080/Projet_test/services/GetAlbumByAuthor.GetAlbumByAuthorHttpSoap11Endpoint/

XML

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:getAlbumsByAuthor>
      <!--Optional:-->
      <ser:artist>Wang</ser:artist>
    </ser:getAlbumsByAuthor>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw

XML

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getAlbumsByAuthorResponse xmlns:ns="http://service" xmlns:ax21="http://schemas.xmlsoap.org/soap/encoding/">
      <ns:return>deep_128</ns:return>
      <ns:return>lere session</ns:return>
      <ns:return>????</ns:return>
      <ns:return>??????</ns:return>
      <ns:return>Abducted By The 80's</ns:return>
      <ns:return>Accident Compilation</ns:return>
      <ns:return>Accident Compilation - Alternative Music From Christch</ns:return>
      <ns:return>BackToBack</ns:return>
      <ns:return>BEAT</ns:return>
      <ns:return>Best of 2013</ns:return>
      <ns:return>BigT on Tour</ns:return>
      <ns:return>Blues Helping</ns:return>
      <ns:return>Born2CreateFam</ns:return>
      <ns:return>Bravo - The Classical Album 2014</ns:return>
      <ns:return>Come Together</ns:return>
      <ns:return>Disney Resort: California Adventure</ns:return>
      <ns:return>DJ Double K - Parkdale Funk</ns:return>
      <ns:return>Du betyr meg</ns:return>
      <ns:return>Encores</ns:return>
      <ns:return>Fantasy Studios</ns:return>
      <ns:return>Flash FM</ns:return>
      <ns:return>From My Manor 2 Your Manor Part 2</ns:return>
      <ns:return>Fuck Your Wobble</ns:return>
      <ns:return>Giblets</ns:return>
      <ns:return>I'm Sorry</ns:return>
      <ns:return>Ibiza Closing Session</ns:return>
      <ns:return>instrumental</ns:return>
      <ns:return>i"?</ns:return>
      <ns:return>Keep It Inside</ns:return>
      <ns:return>Legal Download</ns:return>
      <ns:return>lo mejor del dance</ns:return>
      <ns:return>M...</ns:return>
    </ns:getAlbumsByAuthorResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw

... He... Atta... ... JM... JMS P... Headers (5) Attachments (0) SSL Info WSS (0) JMS (0)

Introduction

Base de Cloud

Docker

Kubernetes

Conclusion

Request 1

http://localhost:8080/Projet_test/services/GetInfoBySong.GetInfoBySongHttpSoap11Endpoint/

XML Raw

```
<?xml version='1.0' encoding='utf-8'>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
      <ser:getInfoBySong>
        <!--Optional:-->
        <ser:song_title>Believe</ser:song_title>
      </ser:getInfoBySong>
    </soapenv:Body>
  </soapenv:Envelope>
```

XML Raw

```
<?xml version='1.0' encoding='utf-8'>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
      <ns:getInfoBySongResponse xmlns:ns="http://service" xmlns:ax217="http://schemas.xmlsoap.org/soap/envelope/">
        <ns:return xsi:type="ax217:GetInfoBySong_Track" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <ax217:album/>
          <ax217:author>Breaking Benjamin</ax217:author>
          <ax217:play_count_lastfm>156121</ax217:play_count_lastfm>
          <ax217:score_brainz/>
          <ax217:title>Believe</ax217:title>
        </ns:return>
        <ns:return xsi:type="ax217:GetInfoBySong_Track" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <ax217:album/>
          <ax217:author>Cher</ax217:author>
          <ax217:play_count_lastfm>418360</ax217:play_count_lastfm>
          <ax217:score_brainz/>
          <ax217:title>Believe</ax217:title>
        </ns:return>
        <ns:return xsi:type="ax217:GetInfoBySong_Track" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <ax217:album>Deejay Parade 2003</ax217:album>
          <ax217:author>DB Boulevard</ax217:author>
          <ax217:play_count_lastfm/>
          <ax217:score_brainz>100</ax217:score_brainz>
          <ax217:title>Believe</ax217:title>
        </ns:return>
        <ns:return xsi:type="ax217:GetInfoBySong_Track" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <ax217:album>Believe</ax217:album>
          <ax217:author>Disturbed</ax217:author>
          <ax217:play_count_lastfm>210239</ax217:play_count_lastfm>
          <ax217:score_brainz>100</ax217:score_brainz>
          <ax217:title>Believe</ax217:title>
        </ns:return>
        <ns:return xsi:type="ax217:GetInfoBySong_Track" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <ax217:album>Love Songs</ax217:album>
          <ax217:author>Elton John</ax217:author>
          <ax217:play_count_lastfm/>
          <ax217:score_brainz/>
          <ax217:title>Believe</ax217:title>
        </ns:return>
      </ns:getInfoBySongResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

Hea... Attach... W... JMS ... JMS Pro... Headers (5) Attachments (0) SSL Info WSS (0) JMS (0)



Morel

THANK YOU

Questions?