



Vrije  
Universiteit  
Brussel

Bachelorproef AMuRate  
3e Bachelor Computerwetenschappen

David Sverdlov  
dsverdlo@vub.ac.be

January 23, 2013



## Contents

<b>1</b>	<b>Inleiding</b>	<b>3</b>
1.1	Doel . . . . .	3
1.2	Referenties . . . . .	3
1.2.1	AMuRate . . . . .	3
1.2.2	Android . . . . .	3
1.2.3	Ice Cream Sandwich . . . . .	3
1.2.4	Last.fm . . . . .	3
1.3	Afkortingen en definities . . . . .	3
1.3.1	API . . . . .	3
1.3.2	XML/JSON . . . . .	3
<b>2</b>	<b>Achtergrond</b>	<b>4</b>
2.1	Last.fm/api: REST . . . . .	4
2.2	REST architectuur . . . . .	4
<b>3</b>	<b>Voorgrond</b>	<b>4</b>
3.1	Website . . . . .	4
3.2	Naam en logo . . . . .	4
<b>4</b>	<b>Vereisten</b>	<b>5</b>
<b>5</b>	<b>Implementatie</b>	<b>5</b>
5.1	Database . . . . .	5
5.1.1	Lokale Database . . . . .	5
5.1.2	Externe Database . . . . .	6
<b>6</b>	<b>Problemen</b>	<b>6</b>
6.1	Synchronous tasks . . . . .	6
6.2	Tracks omzetten van verschillende representaties . . . . .	6
6.3	Database layout . . . . .	7
<b>7</b>	<b>Voortgang</b>	<b>8</b>

# 1 Inleiding

## 1.1 Doel

Het doel van dit project is om een applicatie te ontwikkelen voor het mobiele platform Android. Deze bachelorproef werd gepromote door prof. De applicatie moet gebruikers in staat stellen om muzieknnummers op te zoeken en een beoordeling te kunnen geven. Deze scores worden dan opgeslagen in een database. Er wordt gewerkt met een lokale en een externe database. De applicatie zal ontwikkeld worden voor Android versie 4.0 (Ice Cream Sandwich) om het grootste doelpubliek aan te spreken.

## 1.2 Referenties

### 1.2.1 AMuRate

<http://dsverdlo.github.com/AMuRate>

### 1.2.2 Android

<http://www.android.com/>

### 1.2.3 Ice Cream Sandwich

<http://www.android.com/about/ice-cream-sandwich/>

### 1.2.4 Last.fm

<http://www.last.fm/home>

## 1.3 Afkortingen en definities

### 1.3.1 API

API staat voor Application Programming Interface en zorgt vaak voor de communicatie tussen programmas, door de scheiding te vormen tussen verschillende lagen van abstracties.

### 1.3.2 XML/JSON

XML staat voor Extensible Markup Language en is een van de meest gebruikte opmaaktalen, die gestructureerde gegevens kunnen omzetten in platte tekst. (Om het zo makkelijk(er) door te kunnen sturen.)

JSON is een afkorting van JavaScript Object Notation en is een alternatieve simpele manier om objecten voor te stellen als platte tekst.

## 2 Achtergrond

### 2.1 Last.fm/api: REST

Om (informatie over) muziek te kunnen opzoeken, moet er gebruik gemaakt worden van een online database met een openbare en hanteerbare API. *Last.fm* is een muziek recommendation service met een enorme online muziek database, die ook een gratis (lees: voor geregistreerde gebruikers) API aan, die iedereen toelaat om mobiele/desktop programmas of web services te bouwen met hun data.

Om data uit de database te halen moet er een call (oproep) gestuurd worden. Deze gebeurt via HTTP naar de Last.fm server die antwoordt via REST style XML (of JSON op aanvraag).

### 2.2 REST architectuur

REST staat voor Representational State Transfer en is een type van architectuur voor bepaalde gedistribueerde systemen (zoals het World Wide Web), waarin een duidelijke onderscheiding wordt gemaakt tussen client en server. Clients kunnen oproepen doen naar de server, die de oproepen analyseert, een antwoord formuleert, en dat antwoord terug naar de klant stuurt. REST maakt gebruik van werkwoorden en zelfstandige naamwoorden om de oproepen ook leesbaar voor mensen te maken.

## 3 Voorgond

### 3.1 Website

In dit project wordt er gebruik gemaakt van Git als versiebeheersysteem, omdat het een gratis, eenvoudige en betrouwbare manier is om de broncode te beheren. Nog een voordeel is dat Github de mogelijkheid biedt om snel en simpel websites te maken(en te behoren) voor de bestaande projecten. De website voor dit project is terug te vinden op: <http://dsverdlo.github.com/AMuRate/>. Hier kan men de open-source broncode bekijken, het logboek of dit document raadplegen, en de applicatie (.apk) downloaden voor Android 4.0 toestellen.

### 3.2 Naam en logo

De naam van een project geeft meestal een kleine beschrijving van wat de applicatie doet/ waar hij voor dient. AMuRate komt van "Android Music Rating". Omdat het project vooral geïmplementeerd zou worden via een Android emulator, en simpelweg "AMR" te kort en onduidelijk zou zijn, werd er gekozen voor AMuRate. Het logo (zoals te zien op de voorpagina) is een compositie van een 5-ster in een bol die iets weg heeft van een "dragon

ball” (uit de Japanse animatieserie DragonBall Z). In DBZ bestaan er maar 7 bollen op de hele wereld en wanneer die allemaal verzameld zijn, mag de beheerder een wens doen. Een beetje gelijkaardig aan hoe elke artiest hoopt 5 sterren (/ 7 dragon balls) te verzamelen. In de ster zelf zijn drie letters gekleurd (”AMR”), naar de naam van de applicatie.

## 4 Vereisten

ID	Priority	Beschrijving	Status
1	necessity	Titel (en artiest) opzoeken	done
2	necessity	Track weergeven	done
3	necessity	Enkel op artiest opzoeken	in progress
4	optional	Album tonen	done
5	necessity	Score opslagen in database	done
6	necessity	Score ophalen in lokale database	done
7	necessity	Score opsturen naar remote db	NYI
8	optional	Zoek geschiedenis opslagen	in progress
9	optional	Stream/preview 30sec songs	in progress
10	optional	Caching implementeren	NYI
11	optional	Onderzoek uitbreidbaarheid (ios?)	NYI
12	optional	Informatie ophalen van media speler	NYI

Tabel 1: Vereisten van het project met uitleg en status.

## 5 Implementatie

### 5.1 Database

#### 5.1.1 Lokale Database

Ratings	
PK	_id
text	mbid
float	rating

Afbeelding 1: Dit schema stelt het model voor van de gebruikte database.

De lokale database moet data bijhouden die offline geraadpleegd zou willen worden. Hierbij reken ik op een geschiedenis van de laatste zoekopdrachten en van de laatst gegeven scores. Aangezien deze applicatie vooral beroep doet op

### 5.1.2 Externe Database

Het design van de externe database zal ontworpen moeten worden met oog op de recommender algoritmen, die er later op 'losgelaten' gaan worden. Hier zal later meer informatie over komen.

## 6 Problemen

### 6.1 Synchronous tasks

Wanneer data binnengehaald moet worden van de API, mag die operatie niet gebeuren op de main thread van het programma, anders zal de applicatie zeer traag/onbruikbaar zijn elke keer dat er iets gedownload moet worden. Hierom werd een klasse "MyConnection" gemaakt, die async<sup>1</sup> data begint binnen te laden terwijl de applicatie vloeiend kan blijven voortdraaien.

### 6.2 Tracks omzetten van verschillende representaties

Een tegengekomen probleem in dit project, is dat de Last.fm API oproepen (zoals track.getinfo en track.search) antwoorden met verschillende representaties. Om een voorbeeld te geven: hieronder bevinden zich de resultaten van twee oproepen naar methoden voor het lied 'Bohemian Rhapsody' van Queen. Er kan opgemerkt worden dat in het eerste resultaat 'artist' een top level attribuut is van 'track'. In het tweede snippet is 'artist' opnieuw een object met zijn eigen velden. Om zo een object<sup>2</sup> om te zetten in een 'Track' klasse, moet de Track klasse een onderscheid maken in wat voor soort object gebruikt wordt om de klasse te initialiseren. De beste oplossing hiervoor is om per representatie een loadFromX functie te maken die weet hoe X eruit ziet en waar alle informatie staat.

Track.search resultaat:

```
▼<track>
  <name>Bohemian Rhapsody</name>
  <artist>Queen</artist>
  <url>http://www.last.fm/music/Queen/_/Bohemian+Rhapsody</url>
  <streamable fulltrack="0">0</streamable>
  <listeners>1048163</listeners>
  <image size="small">http://userserve-ak.last.fm/serve/34s/84536633.png</image>
  <image size="medium">http://userserve-ak.last.fm/serve/64s/84536633.png</image>
  <image size="large">http://userserve-ak.last.fm/serve/126/84536633.png</image>
  ▼<image size="extralarge">
    http://userserve-ak.last.fm/serve/300x300/84536633.png
  </image>
  <mbid>050349b6-7879-4eb3-81db-5345f51a6c1f</mbid>
</track>
```

*Afbeelding 2: Resultaat van Track.search GET voor Bohemian Rhapsody - Queen.*

---

<sup>1</sup>Asynchronous betekent dat een process tegelijkertijd naast een ander process kan werken.

<sup>2</sup>In dit document gebruik ik XML notatie omdat deze makkelijker leesbaar is, maar in de broncode wordt er enkel met JSON gewerkt.

Track.getinfo resultaat:

```
▼<track>
  <id>12198</id>
  <name>Bohemian Rhapsody</name>
  <mbid>050349b6-7879-4eb3-81db-5345f51a6c1f</mbid>
  <url>http://www.last.fm/music/Queen/_/Bohemian+Rhapsody</url>
  <duration>356000</duration>
  <streamable fulltrack="0">0</streamable>
  <listeners>1048163</listeners>
  <playcount>6554308</playcount>
  ▼<artist>
    <name>Queen</name>
    <mbid>0383dada-f2a4e-4d10-a46a-e9e041da8eb3</mbid>
    <url>http://www.last.fm/music/Queen</url>
  </artist>
  ▼<album position="11">
    <artist>Queen</artist>
    <title>A Night at the Opera</title>
    <mbid>525a40fc-b3c2-4537-8359-6849467cf79a</mbid>
    ▼<url>
      http://www.last.fm/music/Queen/A+Night+at+the+Opera
    </url>
    <image size="small">http://userserve-ak.last.fm/serve/64s/84536633.png</image>
    <image size="medium">http://userserve-ak.last.fm/serve/126/84536633.png</image>
    ▼<image size="large">
      http://userserve-ak.last.fm/serve/174s/84536633.png
    </image>
    ▼<image size="extralarge">
      http://userserve-ak.last.fm/serve/300x300/84536633.png
    </image>
  </album>
  ►<toptags>...</toptags>
  ►<wiki>...</wiki>
</track>
```

newline Afbeelding 3: Resultaat van Track.getinfo GET voor Bohemian Rhapsody - Queen.

## 6.3 Database layout

nyi

## 7 Voortgang

Datum	Informatie
24-10	Informatie ontvangen i.v.m. opdracht
07-11	Afspraak met begeleiders en promoter, acceptatie bachelorproef
21-11	Android project aangemaakt met simpele GUI. Ook Last.fm account aangemaakt en unieke API sleutel aangevraagd.
05-12	Gebruikers kunnen echte calls maken en muziek opzoeken. Zitten nog bugs in en er kan nog niet gezocht worden naar artiesten
19-12	Er kan ook gezocht worden op artiesten, GUI werd herzien, abstractie in code toegevoegd.
03-01	Een score kan worden opgeslagen in de lokale database, en het totaal en gemiddelde kan uitgelezen worden.
xx-xx	Todo: Titel en artiest opslagen voor geschiedenis GUI. Ook nog op artiest kunnen zoeken. Caching implementeren. Ook alle bugs eruit halen.

*Tabel 2: Enkele belangrijke highlights in de voortgang van het project.*