IIIA - CSIC

Goldsmiths
UNIVERSITY OF LONDON

Mining the social **web** for **music**-related data: a hands-on **tutorial**

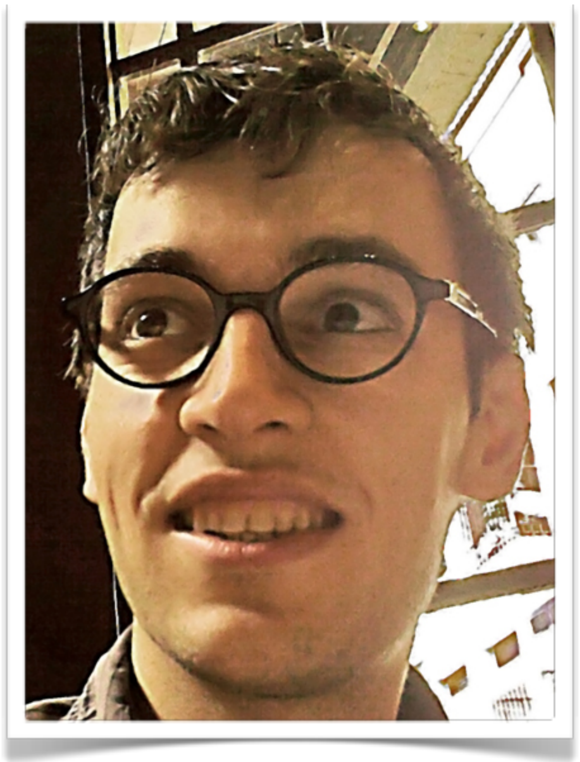
wifi password: yh4zs

Please install the required software!

Check the details at:

ismir2009.benfields.net

Welcome!



Claudio Baccigalupo
IIIA-CSIC
Barcelona, Spain

Ben Fields
Goldsmiths
University of London, UK

ismir2009.benfields.net

Source code archive

All the code examples are included in the file:
ismir2009.benfields.net/sources.zip

Unzip the archive and open a shell in its folder.
Then check that Python and Ruby are installed:

```
$ ruby --version
$ python --version
```

Or download them from:
python.org and ruby-lang.org

It's a hands-on tutorial

You will write code for real MIR applications:

1. Evaluating hypotheses
2. Comparing lyrics by genre
3. Revealing trends
4. Performing audio analysis
5. Capturing social data
6. Collecting feedback

exploring multiple languages and web sites:

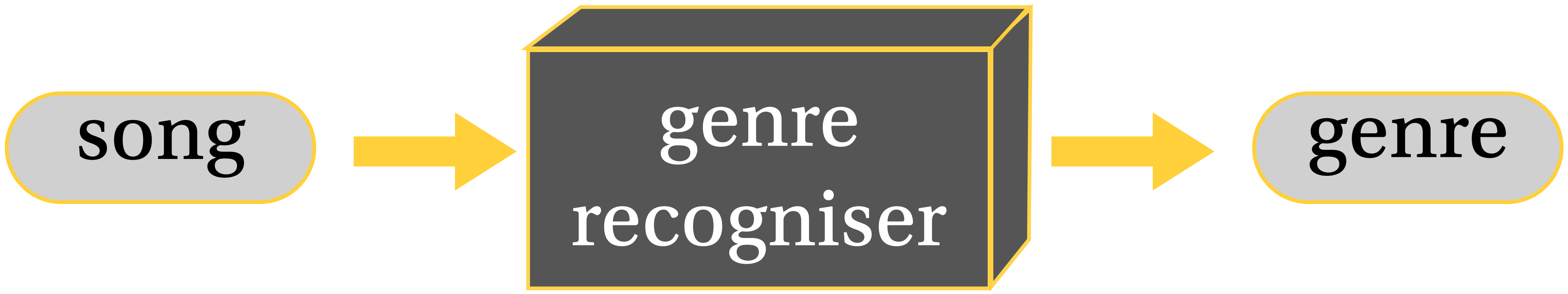
#1

EVALUATING HYPOTHESES

The evaluation process

The evaluation process

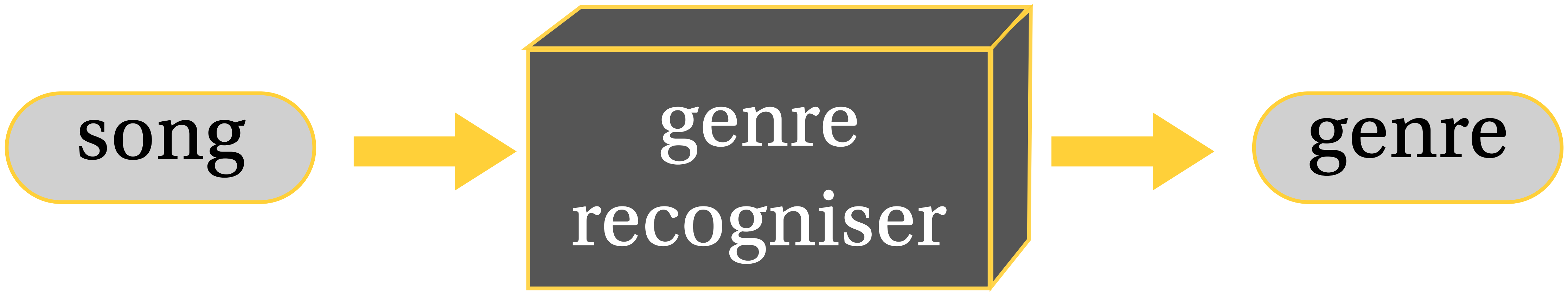
Say you have built the *"ultimate genre recogniser"*



How would you evaluate its precision rate?

The evaluation process

Say you have built the *"ultimate genre recogniser"*



How would you evaluate its precision rate?

1. Build a local collection of varied songs
2. Assign them with a genre label
3. Run the algorithm and check its output

A cumbersome, boring process!

The traditional approach

The traditional approach

Evaluate with a few, manually labelled examples

```
$ cd <PACKAGE PATH>/c

$ python isrock.py ../m/rock.mp3

$ python isrock.py ../m/metal.mp3

$ python isrock.py ../m/vocal.mp3

$ python isrock.py ../m/experimental.mp3
```

The traditional approach

Evaluate with a few, manually labelled examples

```
$ cd <PACKAGE PATH>/c

$ python isrock.py ../m/rock.mp3              =» True

$ python isrock.py ../m/metal.mp3             =» True

$ python isrock.py ../m/vocal.mp3             =» False

$ python isrock.py ../m/experimental.mp3      =» False
```

The *"ultimate"* recogniser or just a coincidence?

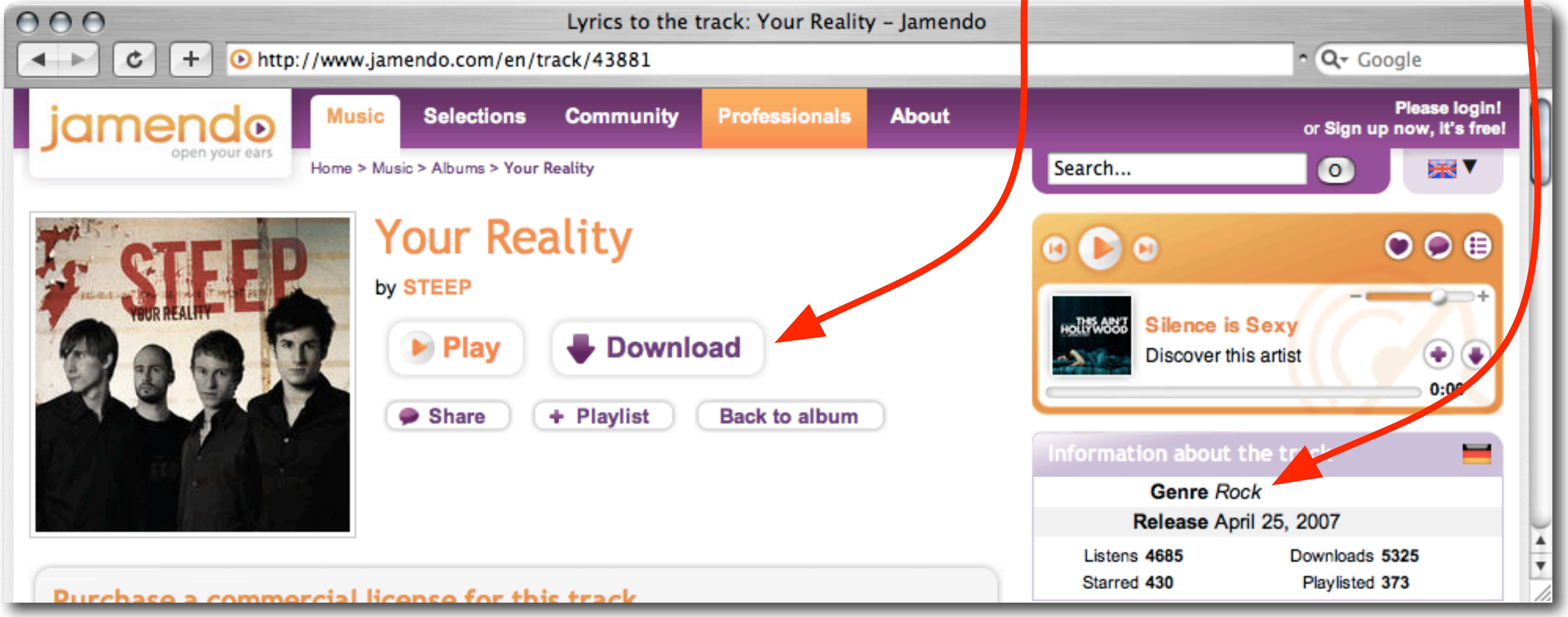
The web-based approach

The web-based approach

The web contains thousands of genre-classified songs that can be legally downloaded for free

The web-based approach

The web contains thousands of genre-classified songs that can be legally downloaded for free


Lyrics to the track: Your Reality – Jamendo

http://www.jamendo.com/en/track/43881

jamendo open your ears

Music Selections Community Professionals About

Please login! or Sign up now, it's free!

Home > Music > Albums > Your Reality

Search...

Your Reality

by STEEP

Play Download

Share + Playlist Back to album

Silence is Sexy

Discover this artist

Information about the track

Genre Rock

Release April 25, 2007

Listens 4685	Downloads 5325
Starred 430	Playlisted 373

Purchase a commercial license for this track

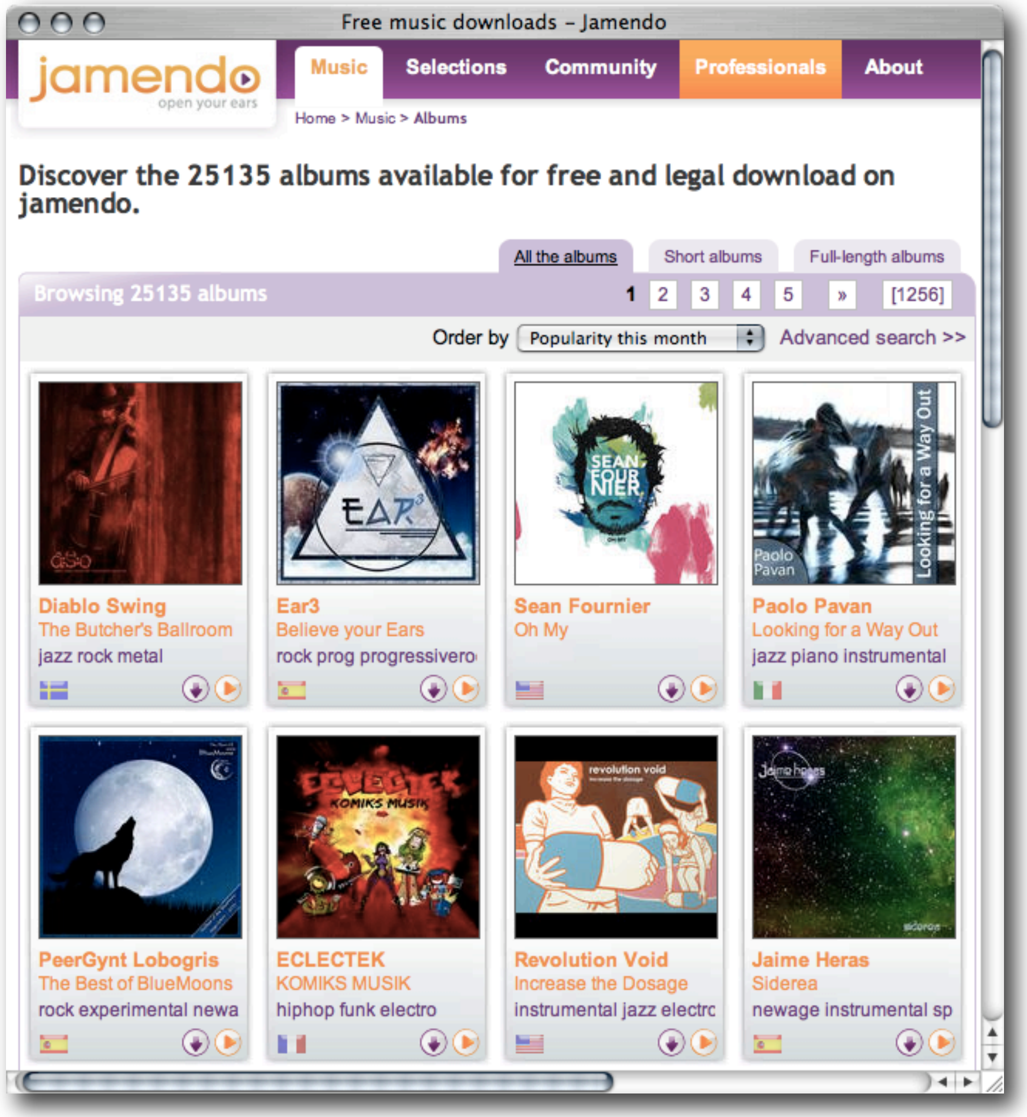
Jamendo includes 170K songs by 14K artists

From browser to web API

A web API allows you to retrieve data in a **compact** format from a site via simple **queries**

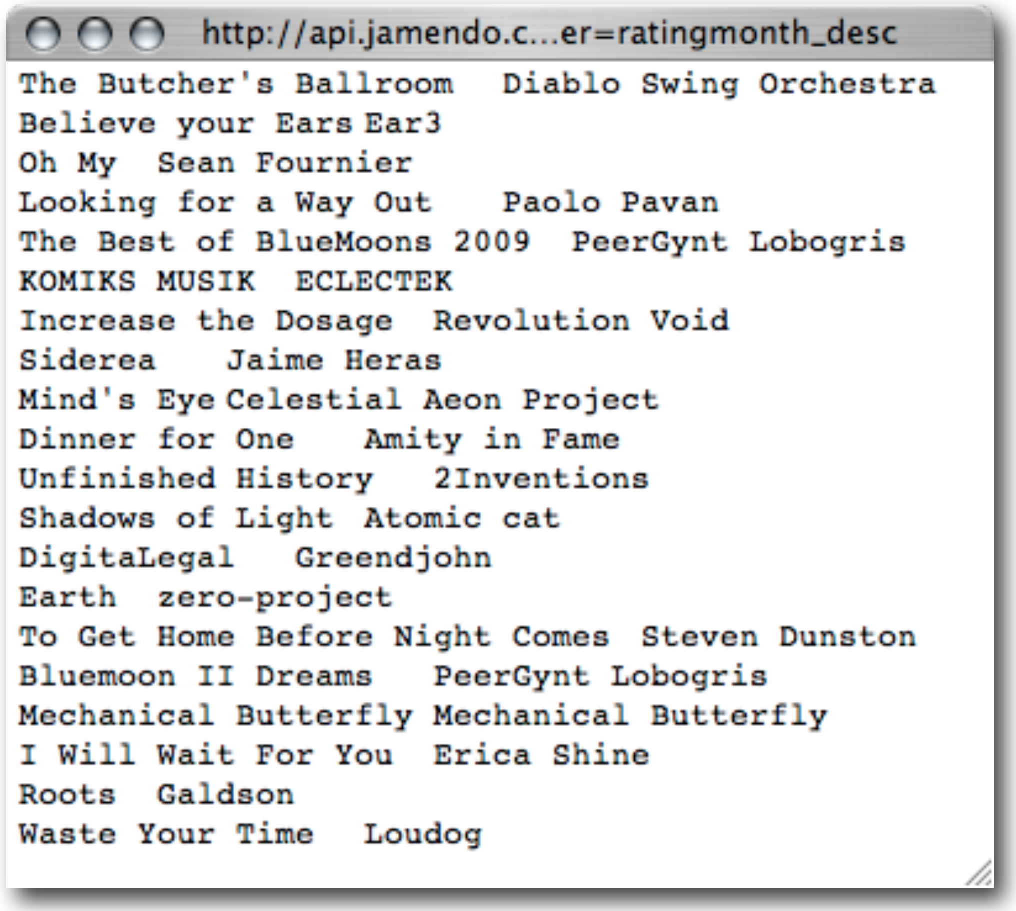
From browser to web API

A web API allows you to retrieve data in a **compact** format from a site via simple **queries**



Discover the 25135 albums available for free and legal download on jamendo.

jamendo.com/en/albums

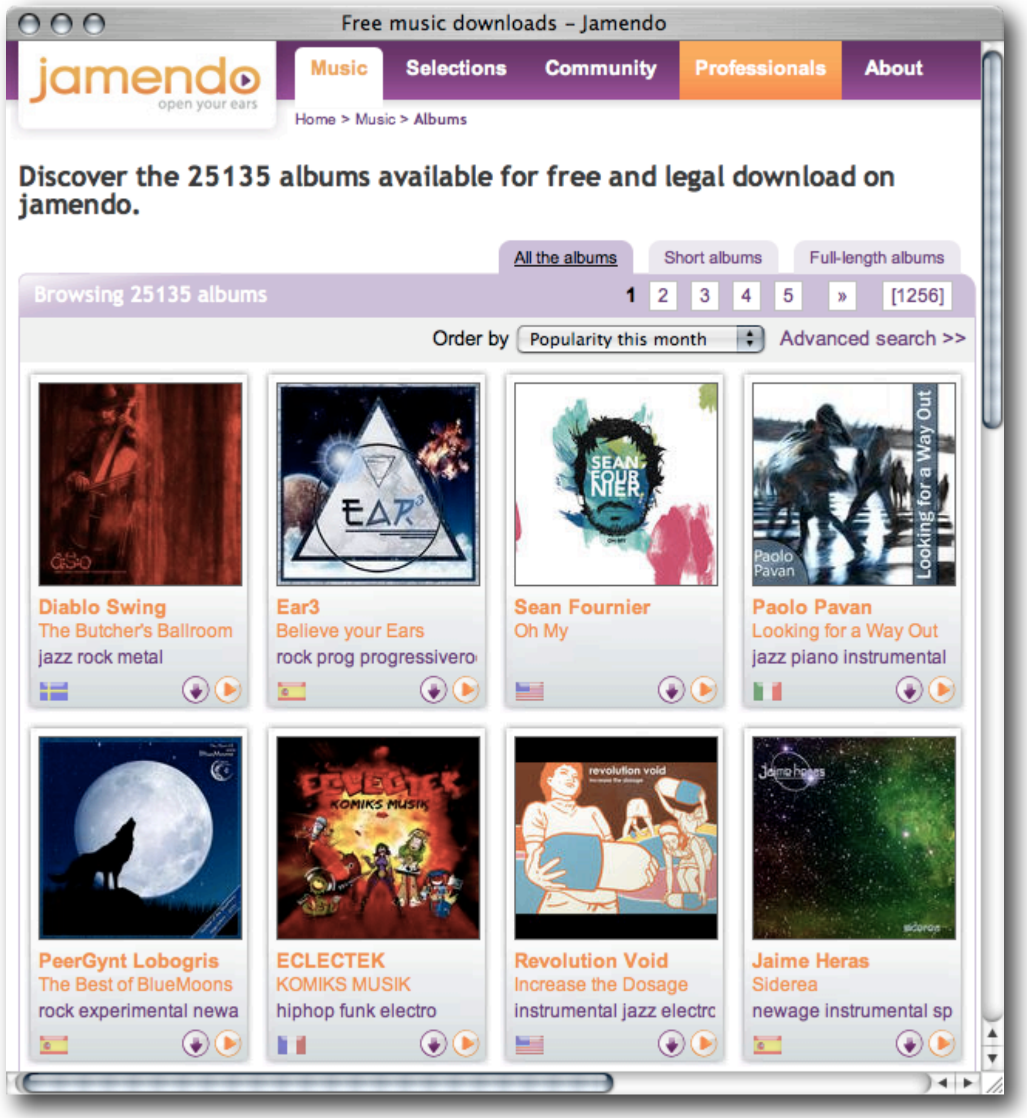


```
The Butcher's Ballroom  Diablo Swing Orchestra
Believe your Ears Ear3
Oh My  Sean Fournier
Looking for a Way Out   Paolo Pavan
The Best of BlueMoons 2009  PeerGynt Lobogris
KOMIKS MUSIK  ECLECTEK
Increase the Dosage  Revolution Void
Siderea   Jaime Heras
Mind's Eye Celestial Aeon Project
Dinner for One   Amity in Fame
Unfinished History  2Inventions
Shadows of Light  Atomic cat
DigitaLegal  Greendjohn
Earth  zero-project
To Get Home Before Night Comes  Steven Dunston
Bluemoon II Dreams  PeerGynt Lobogris
Mechanical Butterfly Mechanical Butterfly
I Will Wait For You  Erica Shine
Roots  Galdson
Waste Your Time  Loudog
```

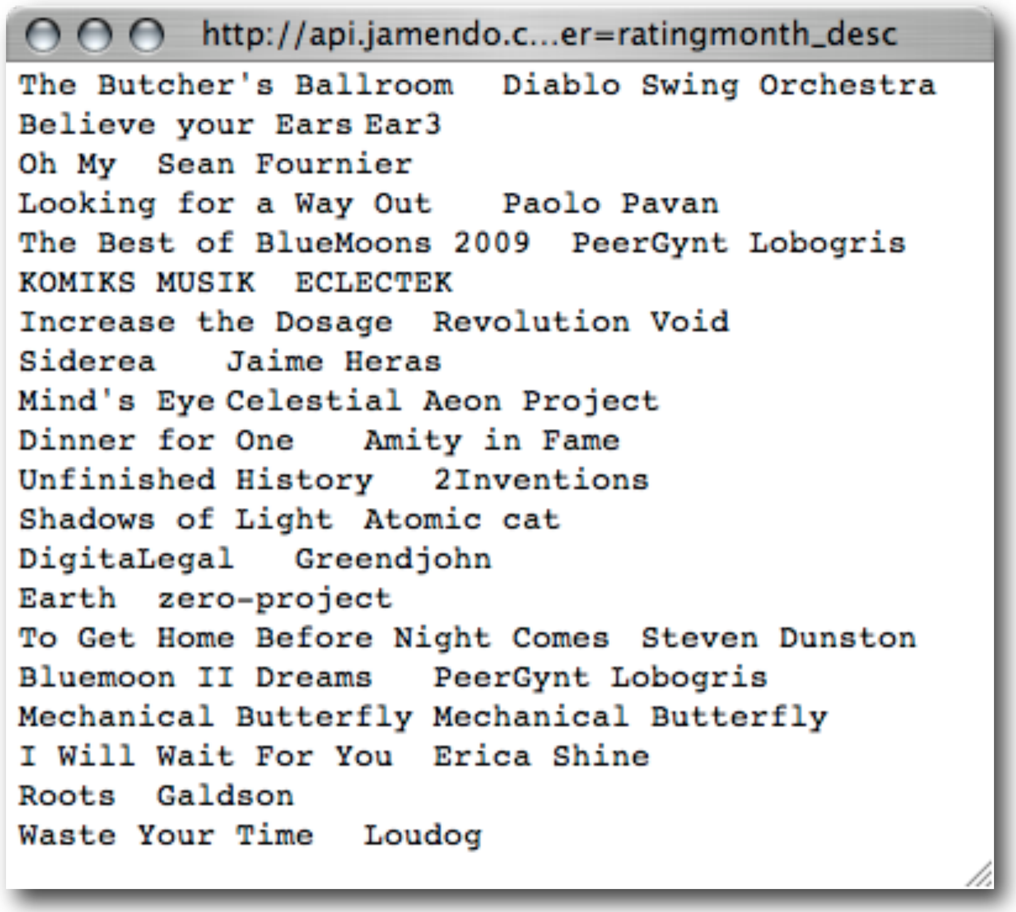
api.jamendo.com/get2/name+artist_name/album/plain/?order=ratingmonth_desc

From browser to web API

A web API allows you to retrieve data in a **compact** format from a site via simple **queries**



jamendo.com/en/albums

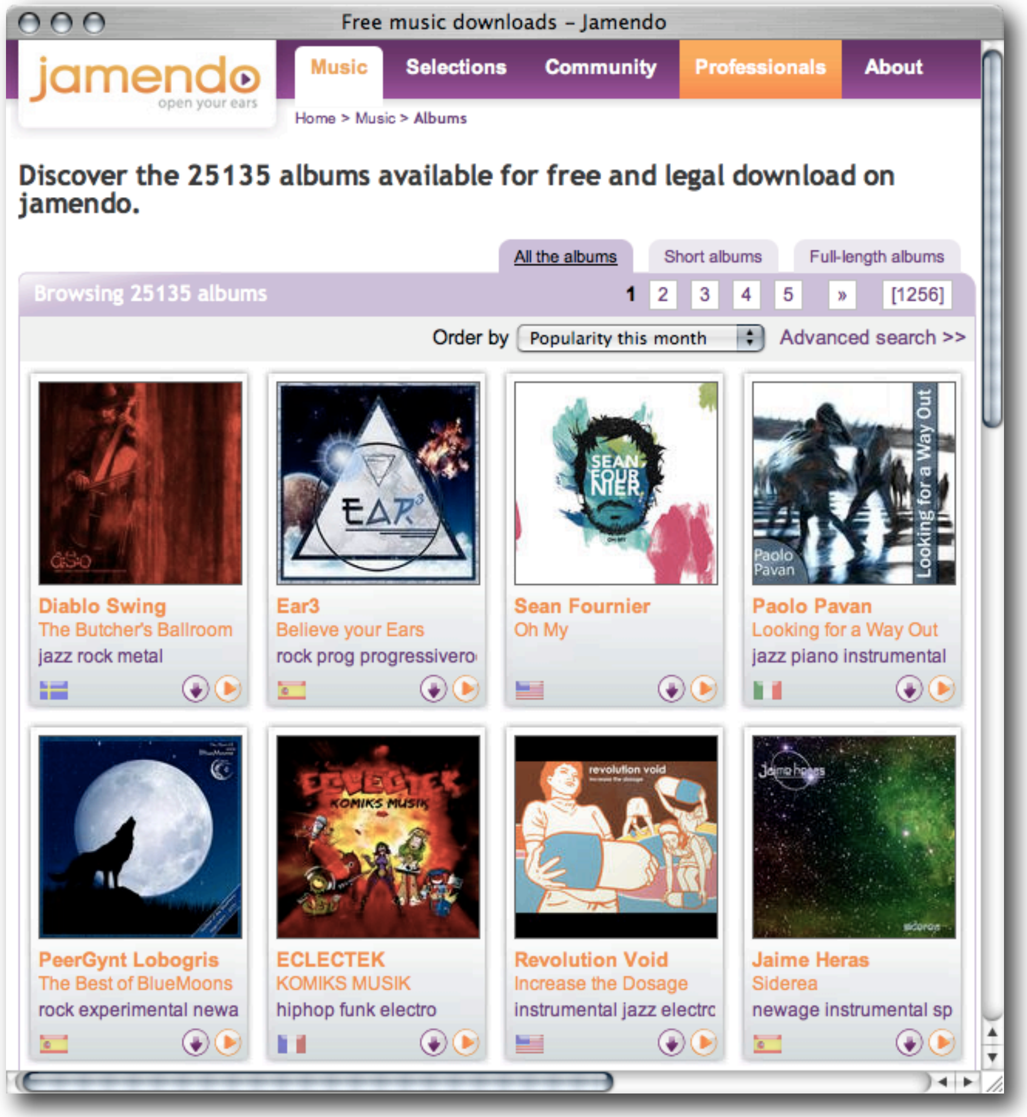


```
The Butcher's Ballroom   Diablo Swing Orchestra
Believe your Ears Ear3
Oh My   Sean Fournier
Looking for a Way Out    Paolo Pavan
The Best of BlueMoons 2009  PeerGynt Lobogris
KOMIKS MUSIK  ECLECTEK
Increase the Dosage  Revolution Void
Siderea    Jaime Heras
Mind's Eye Celestial Aeon Project
Dinner for One    Amity in Fame
Unfinished History   2Inventions
Shadows of Light  Atomic cat
DigitaLegal   Greendjohn
Earth  zero-project
To Get Home Before Night Comes  Steven Dunston
Bluemoon II Dreams   PeerGynt Lobogris
Mechanical Butterfly Mechanical Butterfly
I Will Wait For You  Erica Shine
Roots  Galdson
Waste Your Time   Loudog
```

api.jamendo.com/get2/name+artist_name/album/plain/?order=ratingmonth_desc

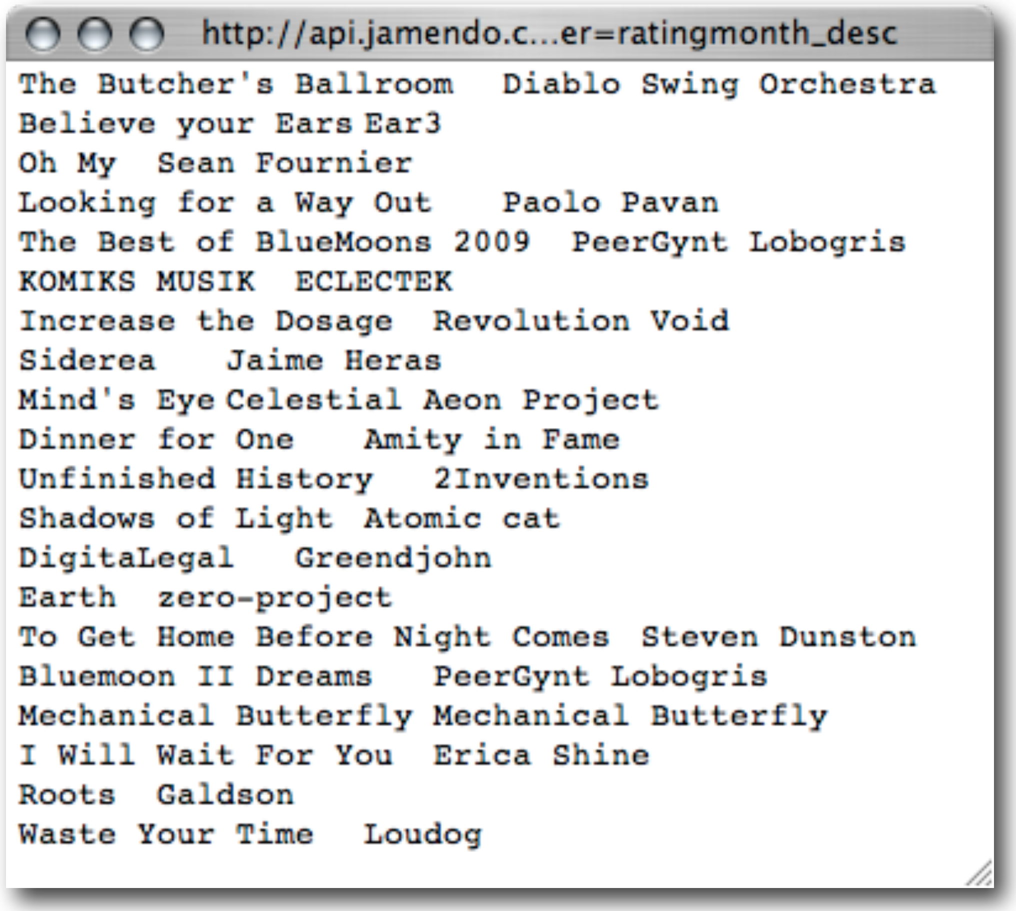
From browser to web API

A web API allows you to retrieve data in a **compact** format from a site via simple **queries**



Discover the 25135 albums available for free and legal download on jamendo.

jamendo.com/en/albums

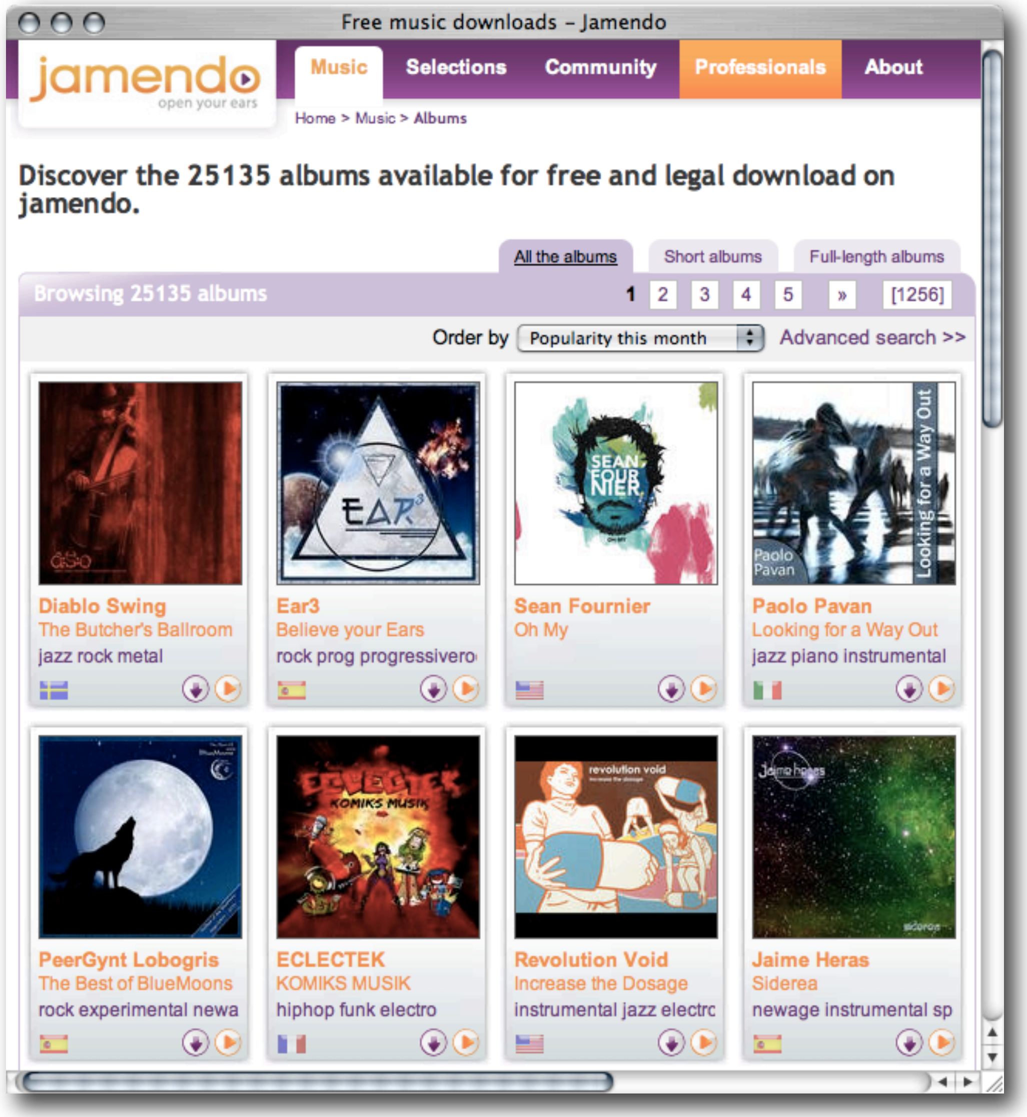


```
The Butcher's Ballroom   Diablo Swing Orchestra
Believe your Ears Ear3
Oh My   Sean Fournier
Looking for a Way Out    Paolo Pavan
The Best of BlueMoons 2009   PeerGynt Lobogris
KOMIKS MUSIK   ECLECTEK
Increase the Dosage   Revolution Void
Siderea    Jaime Heras
Mind's Eye Celestial Aeon Project
Dinner for One    Amity in Fame
Unfinished History   2Inventions
Shadows of Light Atomic cat
DigitaLegal   Greendjohn
Earth   zero-project
To Get Home Before Night Comes  Steven Dunston
Bluemoon II Dreams   PeerGynt Lobogris
Mechanical Butterfly Mechanical Butterfly
I Will Wait For You   Erica Shine
Roots   Galdson
Waste Your Time   Loudog
```

api.jamendo.com/get2/name+artist_name/album/plain/?order=ratingmonth_desc

From browser to web API

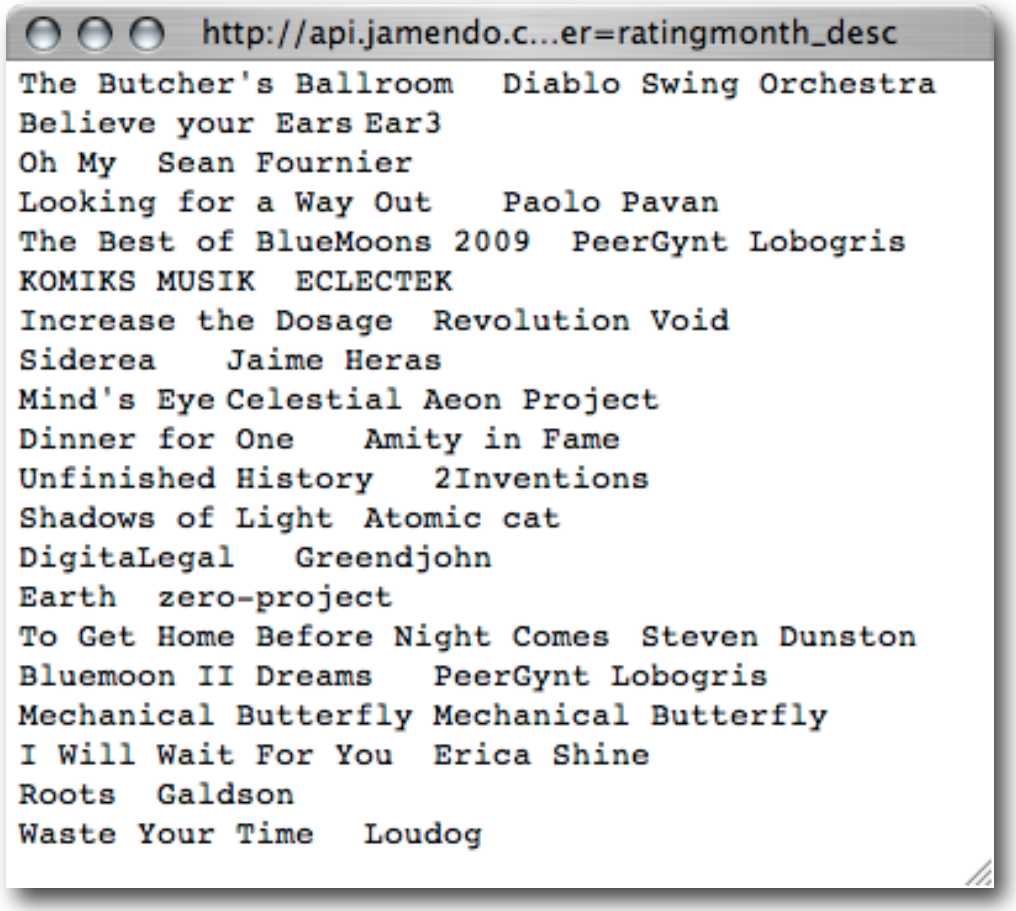
A web API allows you to retrieve data in a **compact** format from a site via simple **queries**



Discover the 25135 albums available for free and legal download on jamendo.

Diablo Swing The Butcher's Ballroom jazz rock metal	Ear3 Believe your Ears rock prog progressivero	Sean Fournier Oh My	Paolo Pavan Looking for a Way Out jazz piano instrumental
PeerGynt Lobogris The Best of BlueMoons rock experimental newa	ECLECTEK KOMIKS MUSIK hiphop funk electro	Revolution Void Increase the Dosage instrumental jazz electrc	Jaime Heras Siderea newage instrumental sp

jamendo.com/en/albums

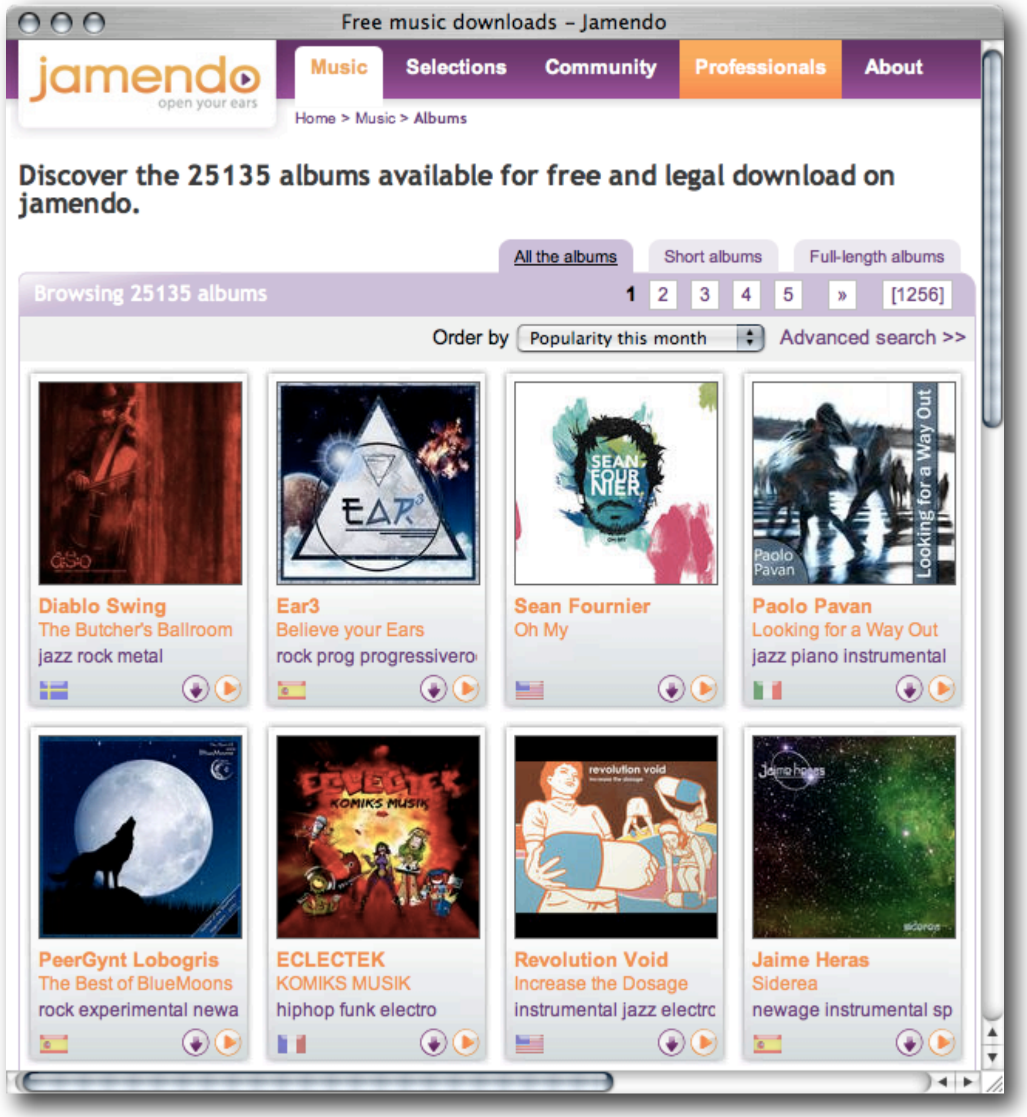


```
The Butcher's Ballroom   Diablo Swing Orchestra
Believe your Ears Ear3
Oh My   Sean Fournier
Looking for a Way Out    Paolo Pavan
The Best of BlueMoons 2009   PeerGynt Lobogris
KOMIKS MUSIK   ECLECTEK
Increase the Dosage   Revolution Void
Siderea    Jaime Heras
Mind's Eye Celestial Aeon Project
Dinner for One    Amity in Fame
Unfinished History   2Inventions
Shadows of Light  Atomic cat
DigitaLegal   Greendjohn
Earth   zero-project
To Get Home Before Night Comes  Steven Dunston
Bluemoon II Dreams   PeerGynt Lobogris
Mechanical Butterfly Mechanical Butterfly
I Will Wait For You   Erica Shine
Roots   Galdson
Waste Your Time   Loudog
```

api.jamendo.com/get2/name+artist_name/album/**plain**/?order=ratingmonth_desc

From browser to web API

A web API allows you to retrieve data in a **compact** format from a site via simple **queries**

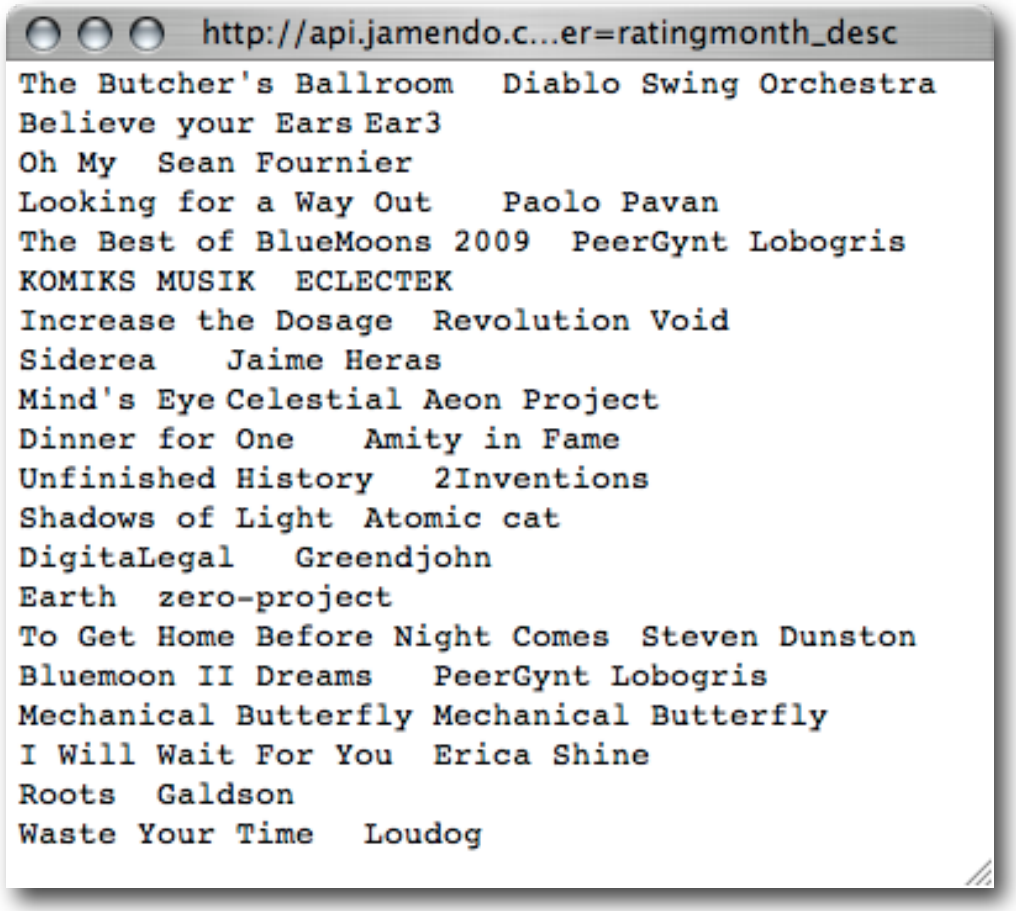


Discover the 25135 albums available for free and legal download on jamendo.

Order by: Popularity this month

Diablo Swing	Ear3	Sean Fournier	Paolo Pavan
The Butcher's Ballroom	Believe your Ears	Oh My	Looking for a Way Out
jazz rock metal	rock prog progressivero		jazz piano instrumental
PeerGynt Lobogris	ECLECTEK	Revolution Void	Jaime Heras
The Best of BlueMoons	KOMIKS MUSIK	Increase the Dosage	Siderea
rock experimental newa	hiphop funk electro	instrumental jazz electrc	newage instrumental sp

jamendo.com/en/albums



```
The Butcher's Ballroom   Diablo Swing Orchestra
Believe your Ears Ear3
Oh My   Sean Fournier
Looking for a Way Out    Paolo Pavan
The Best of BlueMoons 2009  PeerGynt Lobogris
KOMIKS MUSIK  ECLECTEK
Increase the Dosage  Revolution Void
Siderea    Jaime Heras
Mind's Eye Celestial Aeon Project
Dinner for One    Amity in Fame
Unfinished History   2Inventions
Shadows of Light  Atomic cat
DigitaLegal   Greendjohn
Earth  zero-project
To Get Home Before Night Comes  Steven Dunston
Bluemoon II Dreams   PeerGynt Lobogris
Mechanical Butterfly Mechanical Butterfly
I Will Wait For You  Erica Shine
Roots  Galdson
Waste Your Time   Loudog
```

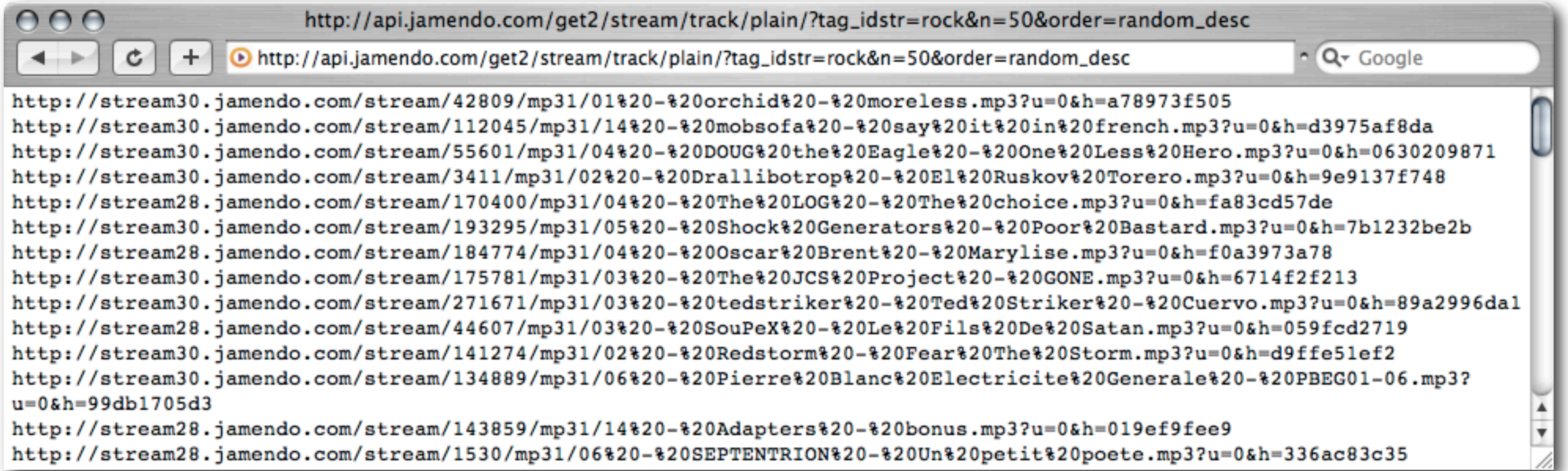
api.jamendo.com/get2/name+artist_name/album/plain/?order=ratingmonth_desc

From browser to web API

From browser to web API

Documentation at: developer.jamendo.com

API query to retrieve 50 random Rock songs:
api.jamendo.com/get2/stream/track/plain/?tag_idstr=rock&n=50&order=random_desc



http://api.jamendo.com/get2/stream/track/plain/?tag_idstr=rock&n=50&order=random_desc
http://api.jamendo.com/get2/stream/track/plain/?tag_idstr=rock&n=50&order=random_desc
Google

```
http://stream30.jamendo.com/stream/42809/mp31/01%20-%20orchid%20-%20moreless.mp3?u=0&h=a78973f505
http://stream30.jamendo.com/stream/112045/mp31/14%20-%20mobsofa%20-%20say%20it%20in%20french.mp3?u=0&h=d3975af8da
http://stream30.jamendo.com/stream/55601/mp31/04%20-%20DOUG%20the%20Eagle%20-%20One%20Less%20Hero.mp3?u=0&h=0630209871
http://stream30.jamendo.com/stream/3411/mp31/02%20-%20Drallibotrop%20-%20El%20Ruskov%20Torero.mp3?u=0&h=9e9137f748
http://stream28.jamendo.com/stream/170400/mp31/04%20-%20The%20LOG%20-%20The%20choice.mp3?u=0&h=fa83cd57de
http://stream30.jamendo.com/stream/193295/mp31/05%20-%20Shock%20Generators%20-%20Poor%20Bastard.mp3?u=0&h=7b1232be2b
http://stream28.jamendo.com/stream/184774/mp31/04%20-%20Oscar%20Brent%20-%20Marylise.mp3?u=0&h=f0a3973a78
http://stream30.jamendo.com/stream/175781/mp31/03%20-%20The%20JCS%20Project%20-%20GONE.mp3?u=0&h=6714f2f213
http://stream30.jamendo.com/stream/271671/mp31/03%20-%20tedstriker%20-%20Ted%20Striker%20-%20Cuervo.mp3?u=0&h=89a2996da1
http://stream28.jamendo.com/stream/44607/mp31/03%20-%20SouPeX%20-%20Le%20Fils%20De%20Satan.mp3?u=0&h=059fcd2719
http://stream30.jamendo.com/stream/141274/mp31/02%20-%20Redstorm%20-%20Fear%20The%20Storm.mp3?u=0&h=d9ffe51ef2
http://stream30.jamendo.com/stream/134889/mp31/06%20-%20Pierre%20Blanc%20Electricite%20Generale%20-%20PBEG01-06.mp3?
u=0&h=99db1705d3
http://stream28.jamendo.com/stream/143859/mp31/14%20-%20Adapters%20-%20bonus.mp3?u=0&h=019ef9fee9
http://stream28.jamendo.com/stream/1530/mp31/06%20-%20SEPTENTRION%20-%20Un%20petit%20poete.mp3?u=0&h=336ac83c35
```

Creating a Python script

Verify the genre recogniser on Jamendo tracks:

Creating a Python script

Verify the genre recogniser on Jamendo tracks:

```
$ python
```

Creating a Python script

Verify the genre recogniser on Jamendo tracks:

```
$ python
from urllib import urlopen
from isrock import isRock
```

Creating a Python script

Verify the genre recogniser on Jamendo tracks:

```
$ python
from urllib import urlopen
from isrock import isRock
query = "http://api.jamendo.com/get2/stream/track/
plain/?n=50&tag_idstr=rock&order=random_desc"
result = urlopen(query).read()
```

Creating a Python script

Verify the genre recogniser on Jamendo tracks:

```
$ python
from urllib import urlopen
from isrock import isRock
query = "http://api.jamendo.com/get2/stream/track/plain/?n=50&tag_idstr=rock&order=random_desc"
result = urlopen(query).read()
songs = result.split()
```

Creating a Python script

Verify the genre recogniser on Jamendo tracks:

```
$ python
from urllib import urlopen
from isrock import isRock
query = "http://api.jamendo.com/get2/stream/track/
plain/?n=50&tag_idstr=rock&order=random_desc"
result = urlopen(query).read()
songs = result.split()
rock = [isRock(song) for song in songs]
print "The ratio of rock songs is: %.2f" % (float
(rock.count(True))/len(songs))
```

Evaluating the genre recogniser

Evaluating the genre recogniser

The code is included in the file **c/jamendo_1.py**:

```
$ python jamendo_1.py
```

```
=» The ratio of rock
songs is: 0.58
```

Evaluating the genre recogniser

The code is included in the file **c/jamendo_1.py**:

```
$ python jamendo_1.py
```

```
=» The ratio of rock
songs is: 0.58
```

The script **c/jamendo_2.py** allows to specify the number of tests and multiple genres at once:

```
$ python jamendo_2.py 30 rock jazz country rnb
```

The result shows that the isRock recogniser **is not able** to distinguish Rock from other genres

Lessons learnt

Music data can easily be **retrieved** from the web

Thousands of songs can be downloaded **for free**

Songs in Jamendo already have a **genre label** attached, so you do not have to decide for one

Working with a **web API** simplifies the process

Different **musical objects** are available (songs, artists, albums, playlists, users) to work on

QUESTIONS?

#2

COMPARING LYRICS BY GENRE

The relevance of lyrics

Recent interest for lyrics-based analysis:

1. Knees, Schedl, Widmer, *Multiple Lyrics Alignment: Automatic Retrieval of Song Lyrics*, 2005

2. Geleijnse, Korst, *Efficient Lyrics Extraction from the web*, 2006

3. Kleedorfer, Knees, Pohle, *Oh Oh Oh Whoah! Towards Automatic Topic Detection in Song Lyrics*, 2008

4. Mayer, Neumayer, Rauber, *Rhyme and Style Features for Musical Genre Categorisation By Song Lyrics*, 2008

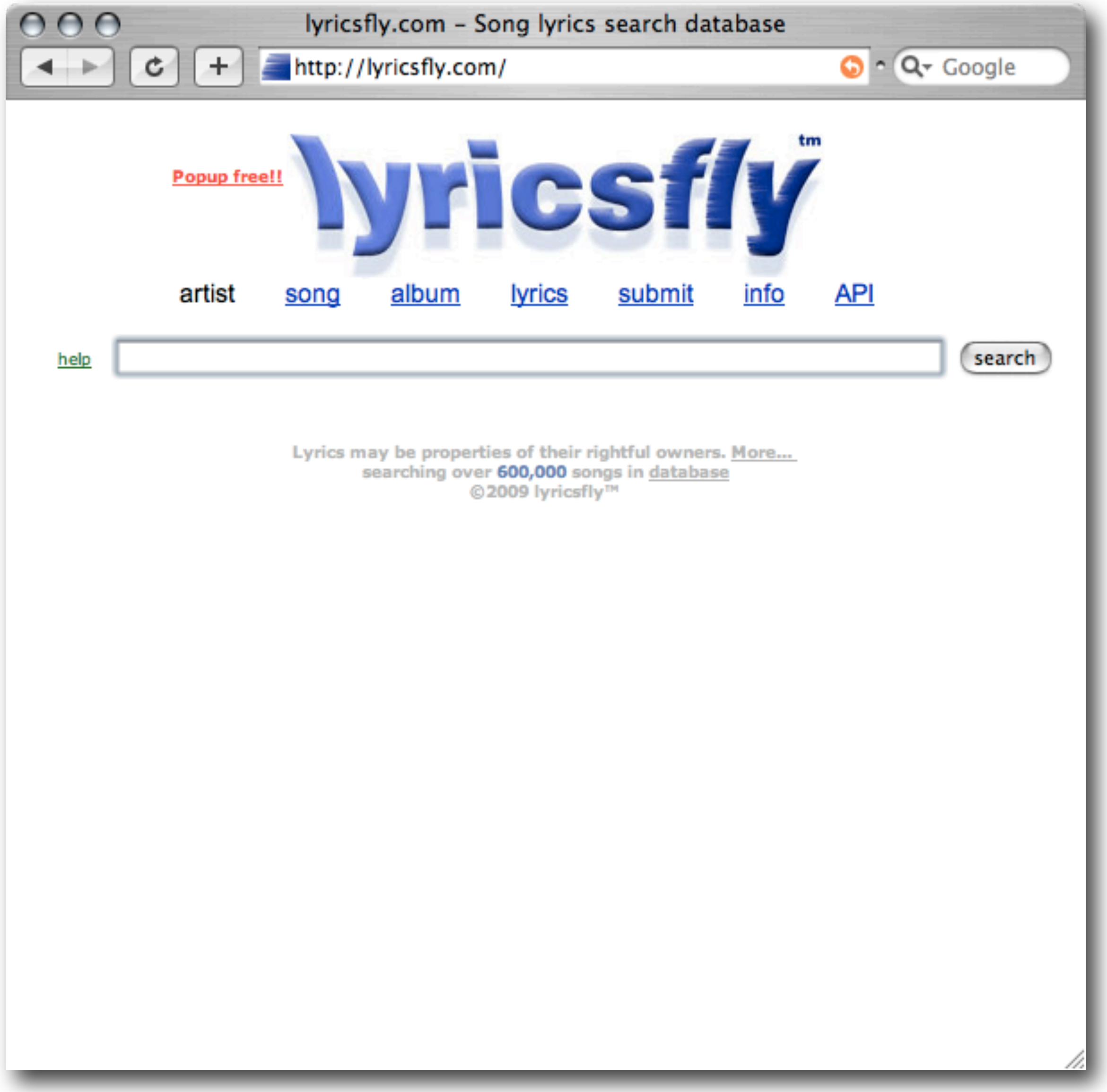
Lyrics were retrieved without using any **web API**

An online song lyrics database

Lyricsfly provides a web API to retrieve lyrics

An online song lyrics database

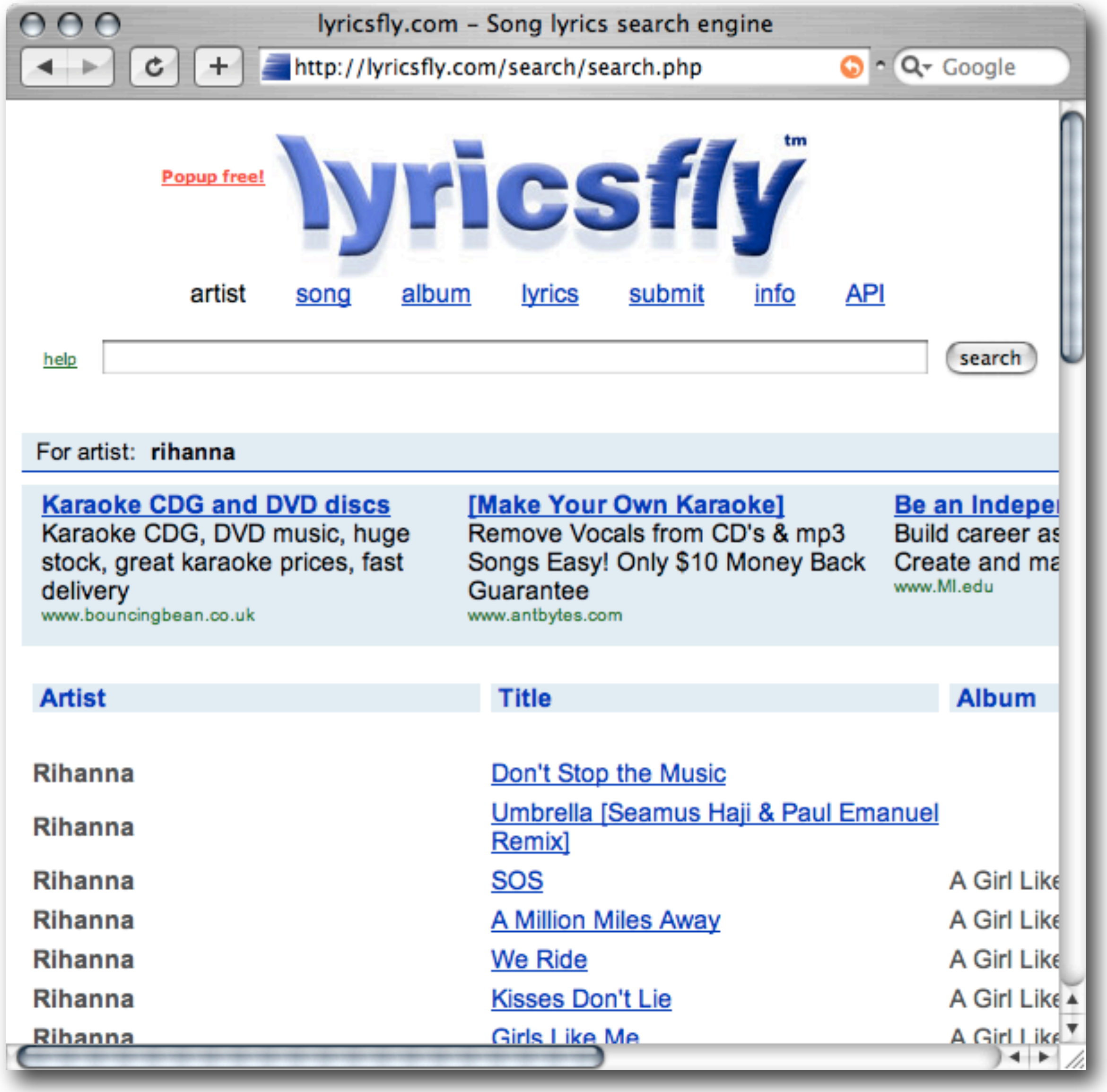
Lyricsfly provides a web API to retrieve lyrics



Lyrics may be properties of their rightful owners. More...
searching over 600,000 songs in database
©2009 lyricsfly™

An online song lyrics database

Lyricsfly provides a web API to retrieve lyrics


lyricsfly.com – Song lyrics search engine

http://lyricsfly.com/search/search.php

For artist: **rihanna**

Karaoke CDG and DVD discs
Karaoke CDG, DVD music, huge stock, great karaoke prices, fast delivery
www.bouncingbean.co.uk

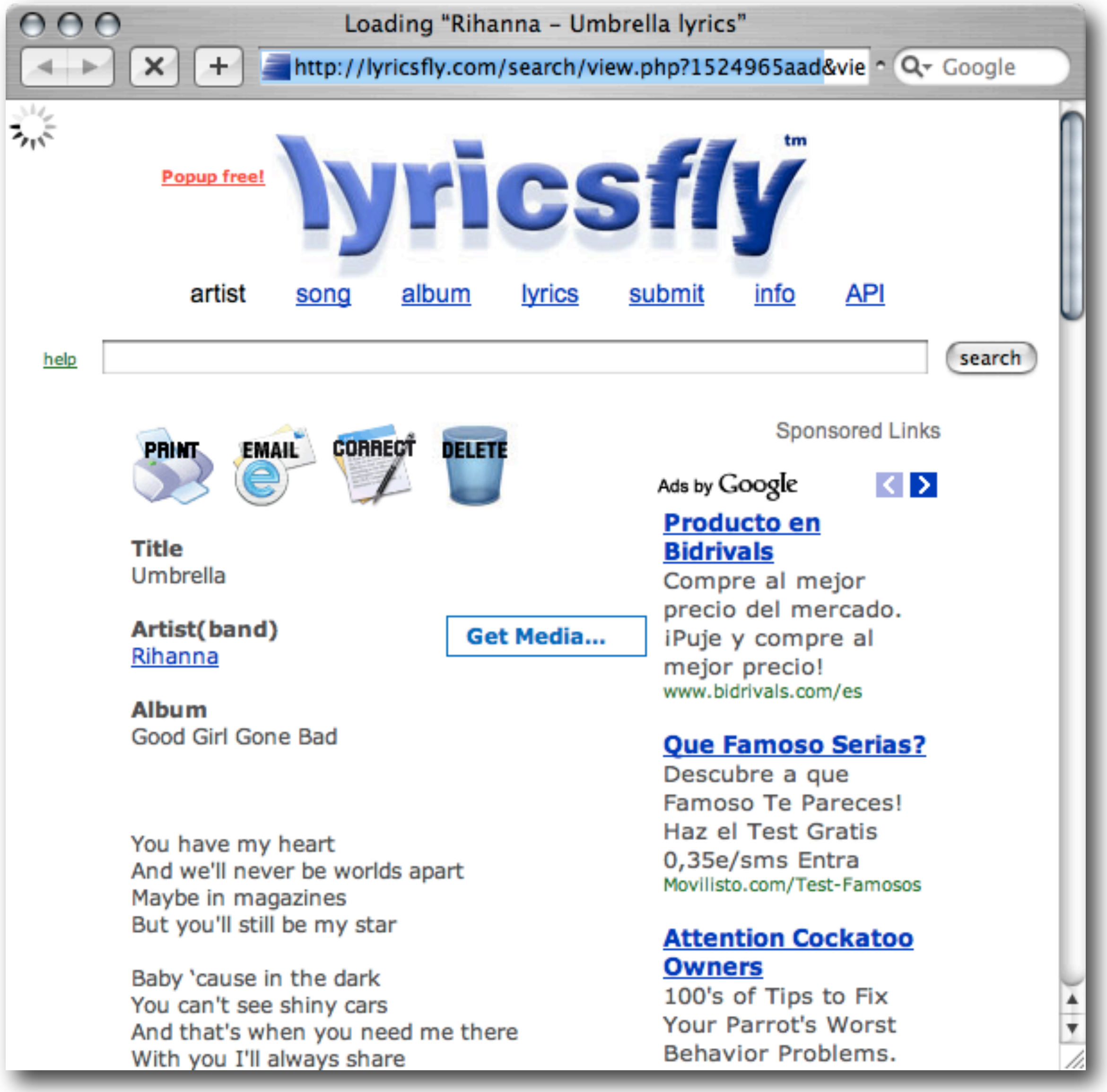
[Make Your Own Karaoke]
Remove Vocals from CD's & mp3 Songs Easy! Only $10 Money Back Guarantee
www.antbytes.com

Be an Indepe
Build career as
Create and ma
www.MI.edu

Artist	Title	Album
Rihanna	Don't Stop the Music	
Rihanna	Umbrella [Seamus Haji & Paul Emanuel Remix]	
Rihanna	SOS	A Girl Like
Rihanna	A Million Miles Away	A Girl Like
Rihanna	We Ride	A Girl Like
Rihanna	Kisses Don't Lie	A Girl Like
Rihanna	Girls Like Me	A Girl Like

An online song lyrics database

Lyricsfly provides a web API to retrieve lyrics


Loading "Rihanna – Umbrella lyrics"

http://lyricsfly.com/search/view.php?1524965aad&vie

Title
Umbrella

Artist(band)
Rihanna

Album
Good Girl Gone Bad

You have my heart
And we'll never be worlds apart
Maybe in magazines
But you'll still be my star

Baby 'cause in the dark
You can't see shiny cars
And that's when you need me there
With you I'll always share

Producto en Bidrivals
Compre al mejor precio del mercado. iPuje y compre al mejor precio!
www.bidrivals.com/es

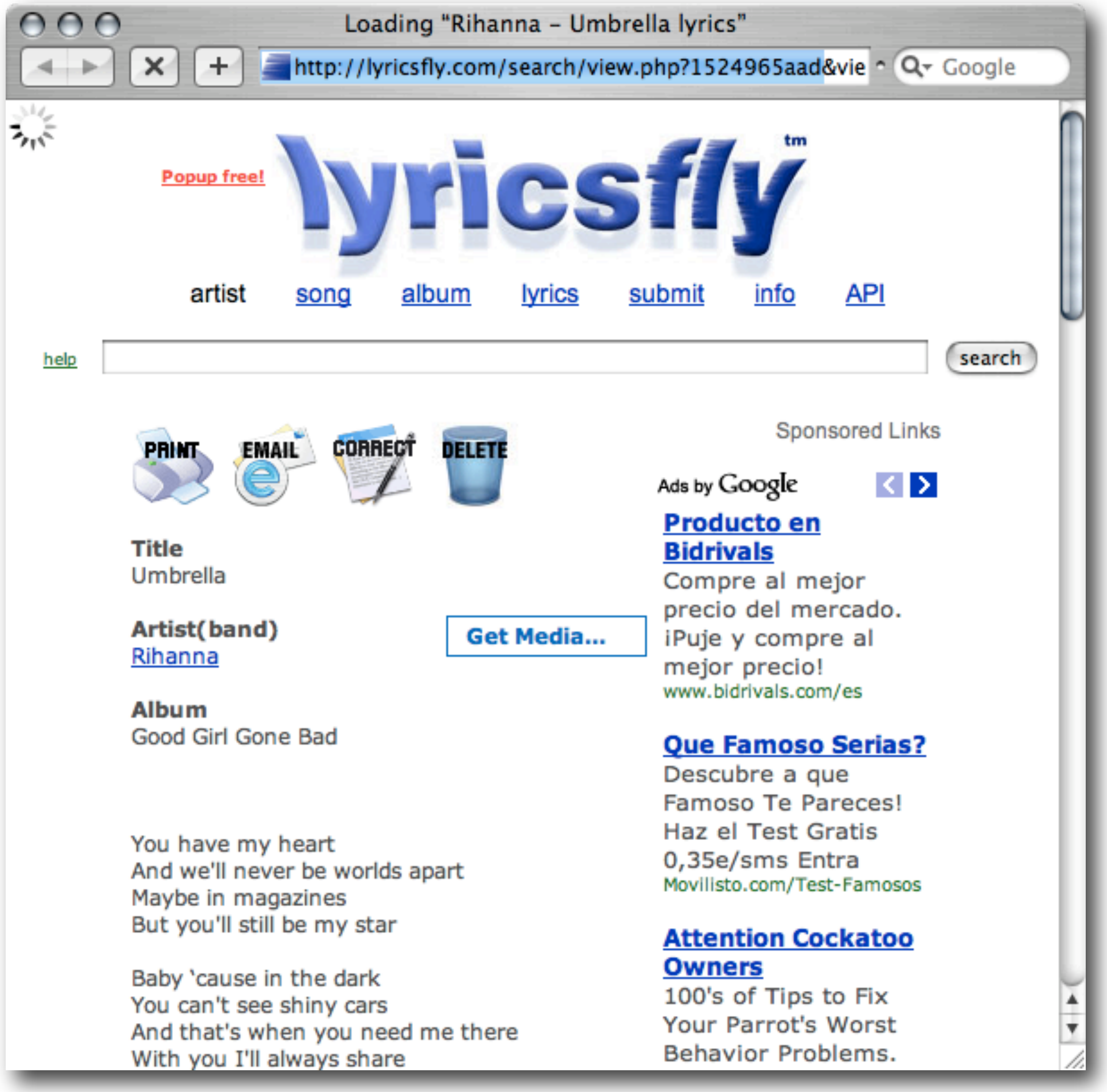
Que Famoso Serias?
Descubre a que Famoso Te Pareces! Haz el Test Gratis 0,35e/sms Entra
Movilisto.com/Test-Famosos

Attention Cockatoo Owners
100's of Tips to Fix Your Parrot's Worst Behavior Problems.

lyricsfly.com/search/view.php?1524965aad&view=578812

An online song lyrics database

Lyricsfly provides a web API to retrieve lyrics



Title
Umbrella

Artist(band)
Rihanna

Album
Good Girl Gone Bad

You have my heart
And we'll never be worlds apart
Maybe in magazines
But you'll still be my star

Baby 'cause in the dark
You can't see shiny cars
And that's when you need me there
With you I'll always share

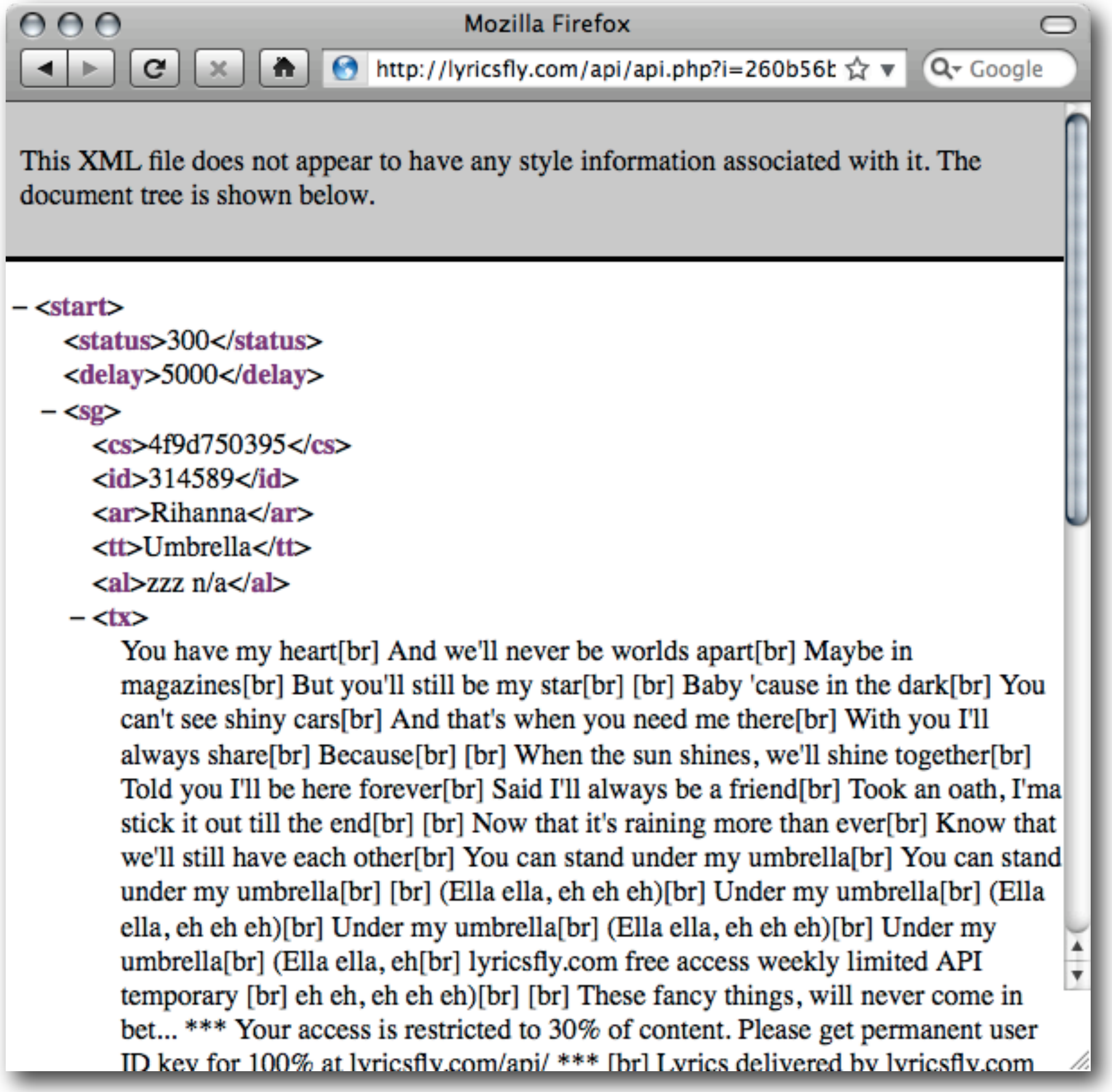
Sponsored Links

Ads by Google

Producto en Bidrivals
Compre al mejor precio del mercado. iPuje y compre al mejor precio!
www.bidrivals.com/es

Que Famoso Serias?
Descubre a que Famoso Te Pareces! Haz el Test Gratis 0,35e/sms Entra
Movilisto.com/Test-Famosos

Attention Cockatoo Owners
100's of Tips to Fix Your Parrot's Worst Behavior Problems.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <start>
    <status>300</status>
    <delay>5000</delay>
  – <sg>
      <cs>4f9d750395</cs>
      <id>314589</id>
      <ar>Rihanna</ar>
      <tt>Umbrella</tt>
      <al>zzz n/a</al>
    – <tx>
        You have my heart[br] And we'll never be worlds apart[br] Maybe in
        magazines[br] But you'll still be my star[br] [br] Baby 'cause in the dark[br] You
        can't see shiny cars[br] And that's when you need me there[br] With you I'll
        always share[br] Because[br] [br] When the sun shines, we'll shine together[br]
        Told you I'll be here forever[br] Said I'll always be a friend[br] Took an oath, I'ma
        stick it out till the end[br] [br] Now that it's raining more than ever[br] Know that
        we'll still have each other[br] You can stand under my umbrella[br] You can stand
        under my umbrella[br] [br] (Ella ella, eh eh eh)[br] Under my umbrella[br] (Ella
        ella, eh eh eh)[br] Under my umbrella[br] (Ella ella, eh eh eh)[br] Under my
        umbrella[br] (Ella ella, eh[br] lyricsfly.com free access weekly limited API
        temporary [br] eh eh, eh eh eh)[br] [br] These fancy things, will never come in
        bet... *** Your access is restricted to 30% of content. Please get permanent user
        ID key for 100% at lyricsfly.com/api/ *** [br] Lyrics delivered by lyricsfly.com
```

lyricsfly.com/search/view.php?1524965aad&view=578812

lyricsfly.com/api/api.php?i=*KEY*&a=Rihanna&t=Umbrella

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

```
$ echo '$lyricsfly_key = "PASTE YOUR KEY
HERE"
```

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

```
$ irb
```

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

```
$ irb
require 'net/http'
require 'rexml/document'
require 'lyricsfly_key'
```

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

```
$ irb
require 'net/http'
require 'rexml/document'
require 'lyricsfly_key'
url = "http://lyricsfly.com/api/api.php?
a=Rihanna&t=Umbrella&i=#{$lyricsfly_key}"
result = Net::HTTP.get_response(URI.parse(url))
```

Creating a Ruby script

To retrieve the lyrics for “Umbrella” (Rihanna):

```
$ irb
require 'net/http'
require 'rexml/document'
require 'lyricsfly_key'
url = "http://lyricsfly.com/api/api.php?
a=Rihanna&t=Umbrella&i=#{$lyricsfly_key}"
result = Net::HTTP.get_response(URI.parse(url))
response = REXML::Document.new
(result.body).elements['//tx']
puts response.text
```

Retrieving multiple lyrics

Retrieving multiple lyrics

The code is included in the file **c/lyricsfly_1.py**:

```
$ ruby lyricsfly_1.rb
```

```
=» You have my heart[br]
And we'll never be ...
```

Retrieving multiple lyrics

The code is included in the file **c/lyricsfly_1.py**:

```
$ ruby lyricsfly_1.rb
```

```
=» You have my heart[br]
And we'll never be ...
```

The script **c/lyricsfly_2.rb** allows to specify the artist name and track title:

```
$ ruby lyricsfly_2.rb
"John Lennon" Imagine
```

```
=» Imagine there's no
Heaven

It's easy if you try

No Hell below us ...
```

Lyrics-based analysis

Mayer, Neumayer, Rauber, *Rhyme and Style Features for Musical Genre Categorisation By Song Lyrics*, 2008

Textual features of lyrics are **related to the genre**

Hip-hop lyrics have more ‘**?**’ than Country ones

Evaluated on 29 Hip-hop and 41 Country songs

Does this hold with larger data sets?

Repeating the experiment

$$\begin{tikzcd}
\text{‘\textit{Country}’ and ‘\textit{Hip-hop}’} \arrow[d] & \\
\boxed{\text{music web API \#1}} \arrow[d] & \\
\text{List \textbf{songs} by genre} \arrow[d] & \\
\boxed{\text{music web API \#2}} \arrow[d] & \\
\text{List \textbf{lyrics} by genre} \arrow[r] & \text{Count ‘\textbf{?}’ by genre}
\end{tikzcd}$$

Repeating the experiment

'Country' and *'Hip-hop'*

↓

music web API #1

↓

List **songs** by genre

↓

music web API #2

↓

List **lyrics** by genre → Count '**?**' by genre

lyricsfly™

Repeating the experiment

'*Country*' and '*Hip-hop*'

↓

music web API #1

↓

List **songs** by genre

↓

music web API #2

↓

List **lyrics** by genre → Count '**?**' by genre

jamendo
open your ears

lyricsfly tm

Repeating the experiment

$$\begin{tikzcd}
\text{‘\textit{Country}’ and ‘\textit{Hip-hop}’} \arrow[d] & \\
\boxed{\text{music web API \#1}} \arrow[d] & \\
\text{List \textbf{songs} by genre} \arrow[d] & \\
\boxed{\text{music web API \#2}} \arrow[d] & \\
\text{List \textbf{lyrics} by genre} \arrow[r] & \text{Count ‘\textbf{?}’ by genre}
\end{tikzcd}$$

last.fm

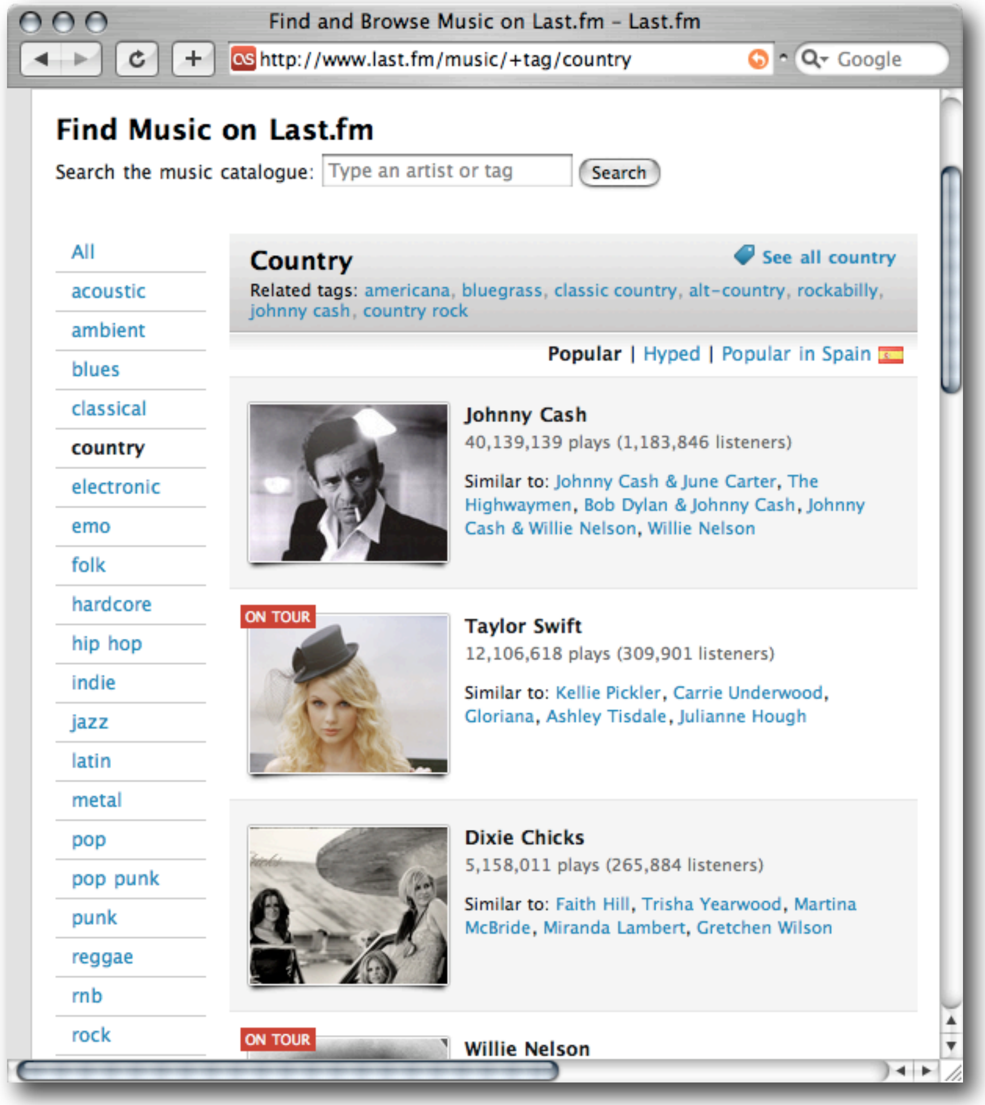
lyricsfly tm

Retrieving songs by genre

Last.fm has 4M songs classified by tags/genres

Retrieving songs by genre

Last.fm has 4M songs classified by tags/genres


Find and Browse Music on Last.fm - Last.fm

http://www.last.fm/music/+tag/country

Find Music on Last.fm

Search the music catalogue: Type an artist or tag Search

All
acoustic
ambient
blues
classical
country
electronic
emo
folk
hardcore
hip hop
indie
jazz
latin
metal
pop
pop punk
punk
reggae
rnb
rock

Country See all country

Related tags: americana, bluegrass, classic country, alt-country, rockabilly, johnny cash, country rock

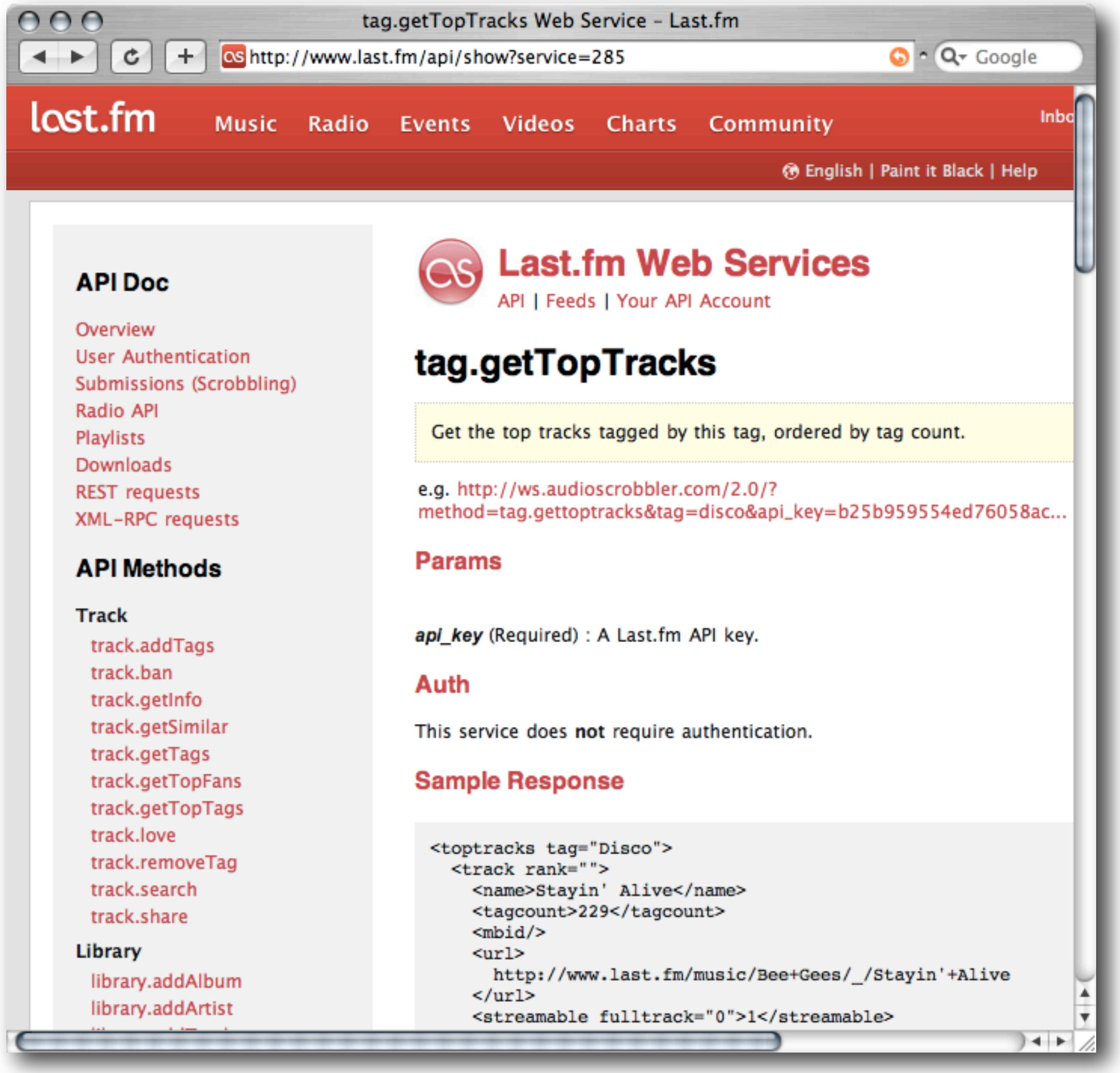
Popular | Hyped | Popular in Spain

Johnny Cash
40,139,139 plays (1,183,846 listeners)
Similar to: Johnny Cash & June Carter, The Highwaymen, Bob Dylan & Johnny Cash, Johnny Cash & Willie Nelson, Willie Nelson

ON TOUR
Taylor Swift
12,106,618 plays (309,901 listeners)
Similar to: Kellie Pickler, Carrie Underwood, Gloriana, Ashley Tisdale, Julianne Hough

Dixie Chicks
5,158,011 plays (265,884 listeners)
Similar to: Faith Hill, Trisha Yearwood, Martina McBride, Miranda Lambert, Gretchen Wilson

ON TOUR
Willie Nelson


tag.getTopTracks Web Service - Last.fm

http://www.last.fm/api/show?service=285

last.fm Music Radio Events Videos Charts Community

English | Paint it Black | Help

API Doc

Overview
User Authentication
Submissions (Scrobbling)
Radio API
Playlists
Downloads
REST requests
XML-RPC requests

API Methods

Track
track.addTags
track.ban
track.getInfo
track.getSimilar
track.getTags
track.getTopFans
track.getTopTags
track.love
track.removeTag
track.search
track.share

Library
library.addAlbum
library.addArtist

Last.fm Web Services

API | Feeds | Your API Account

tag.getTopTracks

Get the top tracks tagged by this tag, ordered by tag count.

e.g. http://ws.audioscrobbler.com/2.0/?method=tag.gettoptracks&tag=disco&api_key=b25b959554ed76058ac...

Params

api_key (Required) : A Last.fm API key.

Auth

This service does **not** require authentication.

Sample Response

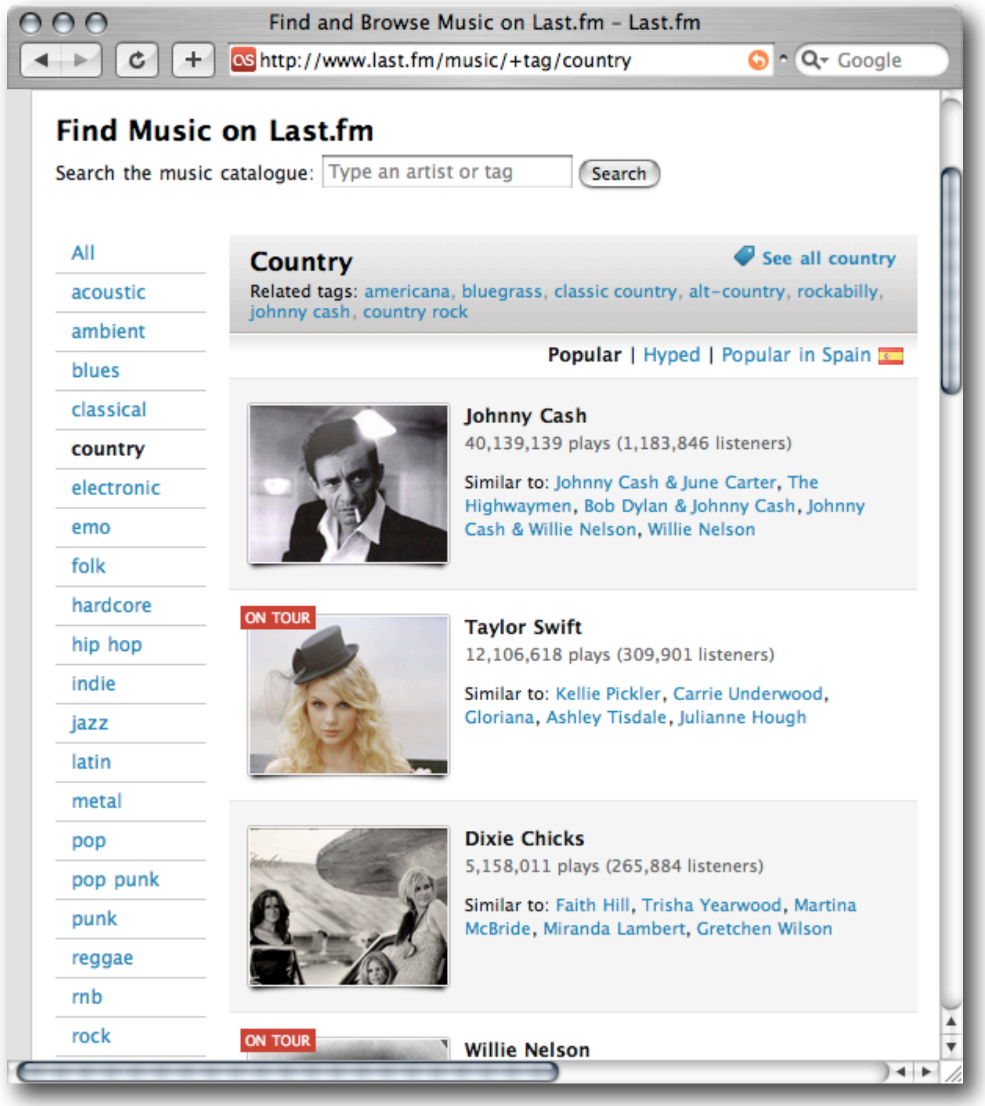
```
<toptracks tag="Disco">
  <track rank="">
    <name>Stayin' Alive</name>
    <tagcount>229</tagcount>
    <mbid/>
    <url>
      http://www.last.fm/music/Bee+Gees/_/Stayin'+Alive
    </url>
    <streamable fulltrack="0">1</streamable>
```

last.fm/music/+tag/country

last.fm/api/show?service=285

Retrieving songs by genre

Last.fm has 4M songs classified by tags/genres


Find and Browse Music on Last.fm – Last.fm
http://www.last.fm/music/+tag/country
Google

Find Music on Last.fm

Search the music catalogue: Type an artist or tag Search

All
acoustic
ambient
blues
classical
country
electronic
emo
folk
hardcore
hip hop
indie
jazz
latin
metal
pop
pop punk
punk
reggae
rnb
rock

Country — See all country

Related tags: americana, bluegrass, classic country, alt-country, rockabilly, johnny cash, country rock

Popular | Hyped | Popular in Spain

Johnny Cash
40,139,139 plays (1,183,846 listeners)
Similar to: Johnny Cash & June Carter, The Highwaymen, Bob Dylan & Johnny Cash, Johnny Cash & Willie Nelson, Willie Nelson

ON TOUR
Taylor Swift
12,106,618 plays (309,901 listeners)
Similar to: Kellie Pickler, Carrie Underwood, Gloriana, Ashley Tisdale, Julianne Hough

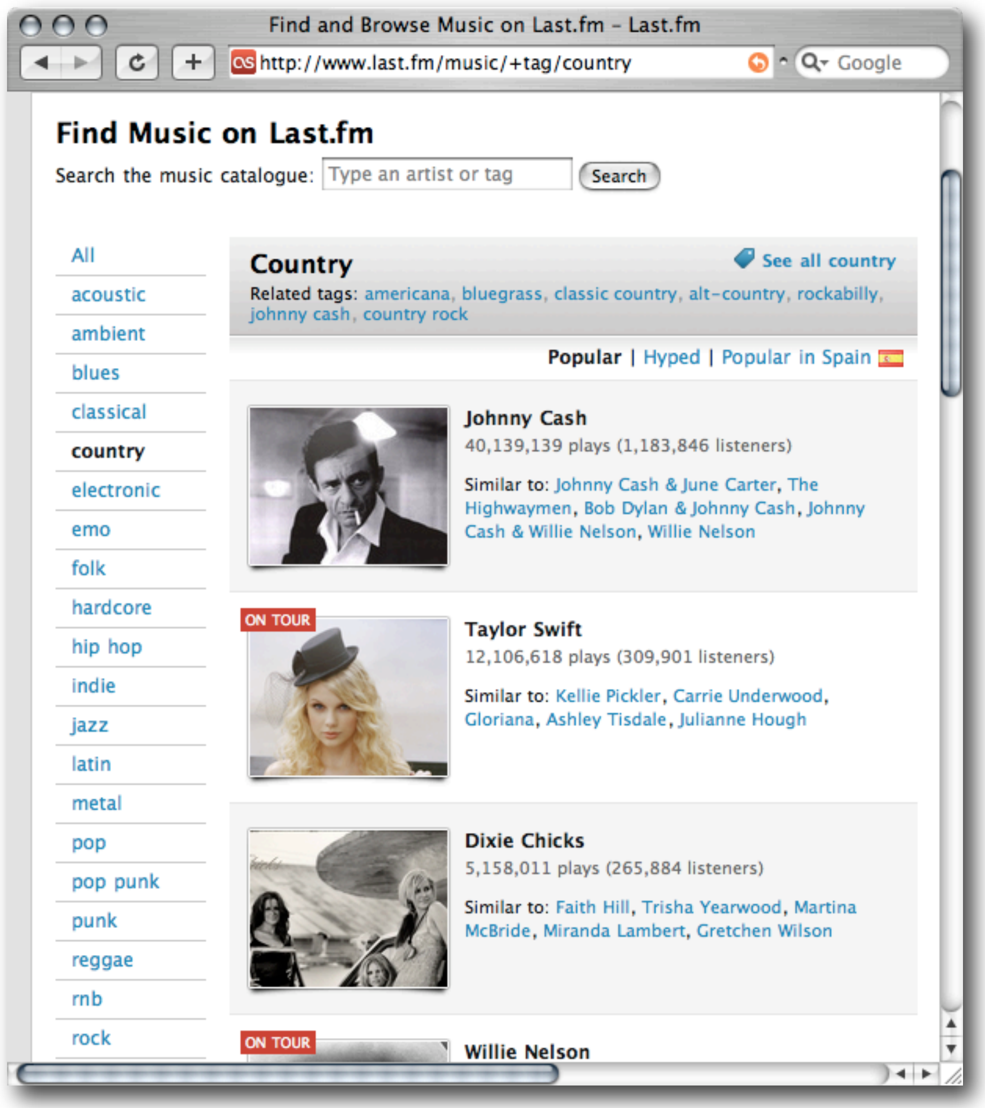
Dixie Chicks
5,158,011 plays (265,884 listeners)
Similar to: Faith Hill, Trisha Yearwood, Martina McBride, Miranda Lambert, Gretchen Wilson

ON TOUR
Willie Nelson

last.fm/music/+tag/country

Retrieving songs by genre

Last.fm has 4M songs classified by tags/genres


Find and Browse Music on Last.fm – Last.fm

http://www.last.fm/music/+tag/country

Find Music on Last.fm

Search the music catalogue: Type an artist or tag Search

All
acoustic
ambient
blues
classical
country
electronic
emo
folk
hardcore
hip hop
indie
jazz
latin
metal
pop
pop punk
punk
reggae
rnb
rock

Country

See all country

Related tags: americana, bluegrass, classic country, alt-country, rockabilly, johnny cash, country rock

Popular | Hyped | Popular in Spain

Johnny Cash
40,139,139 plays (1,183,846 listeners)
Similar to: Johnny Cash & June Carter, The Highwaymen, Bob Dylan & Johnny Cash, Johnny Cash & Willie Nelson, Willie Nelson

ON TOUR
Taylor Swift
12,106,618 plays (309,901 listeners)
Similar to: Kellie Pickler, Carrie Underwood, Gloriana, Ashley Tisdale, Julianne Hough

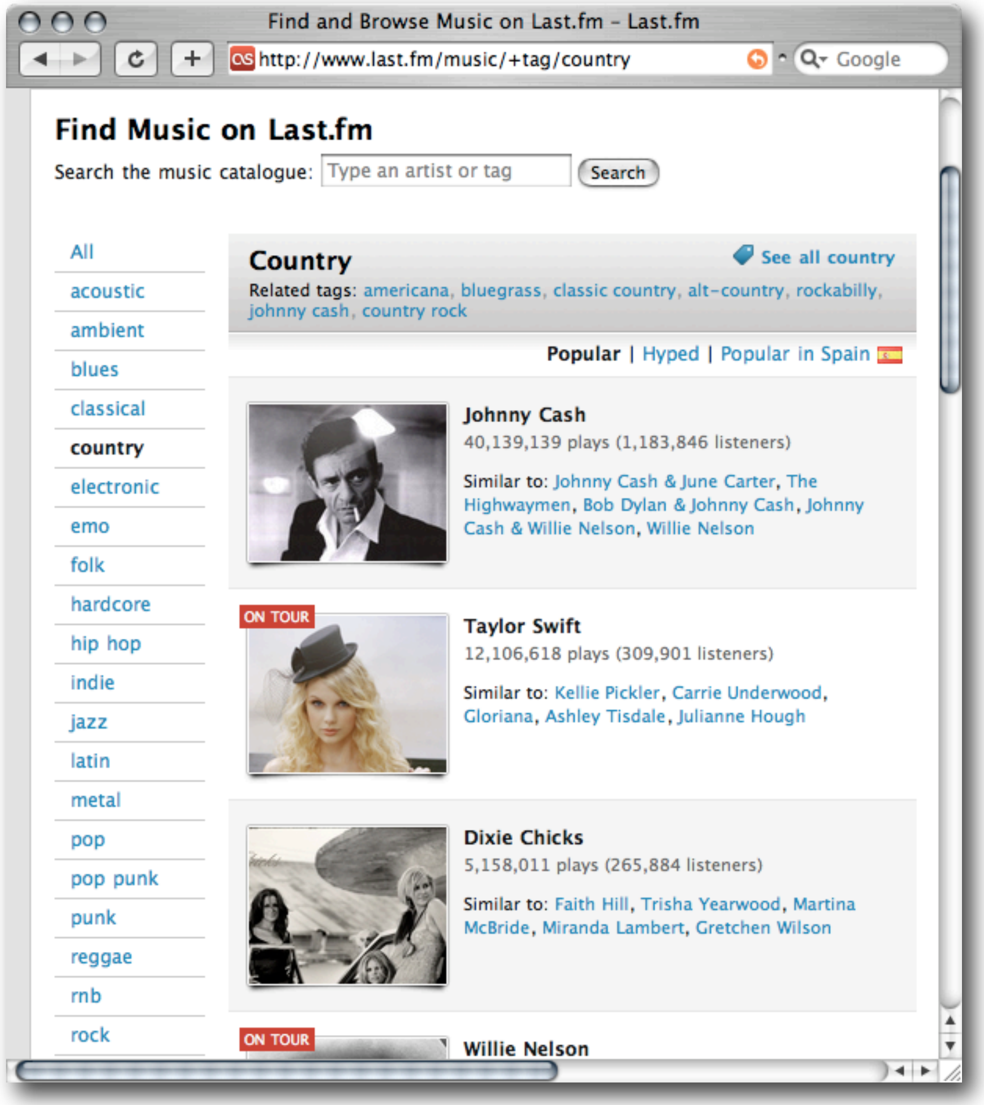
Dixie Chicks
5,158,011 plays (265,884 listeners)
Similar to: Faith Hill, Trisha Yearwood, Martina McBride, Miranda Lambert, Gretchen Wilson

ON TOUR
Willie Nelson

last.fm/music/+tag/country

Retrieving songs by genre

Last.fm has 4M songs classified by tags/genres



Find and Browse Music on Last.fm – Last.fm

http://www.last.fm/music/+tag/country

Find Music on Last.fm

Search the music catalogue: Type an artist or tag [Search]

All · acoustic · ambient · blues · classical · **country** · electronic · emo · folk · hardcore · hip hop · indie · jazz · latin · metal · pop · pop punk · punk · reggae · rnb · rock

Country — See all country

Related tags: americana, bluegrass, classic country, alt-country, rockabilly, johnny cash, country rock

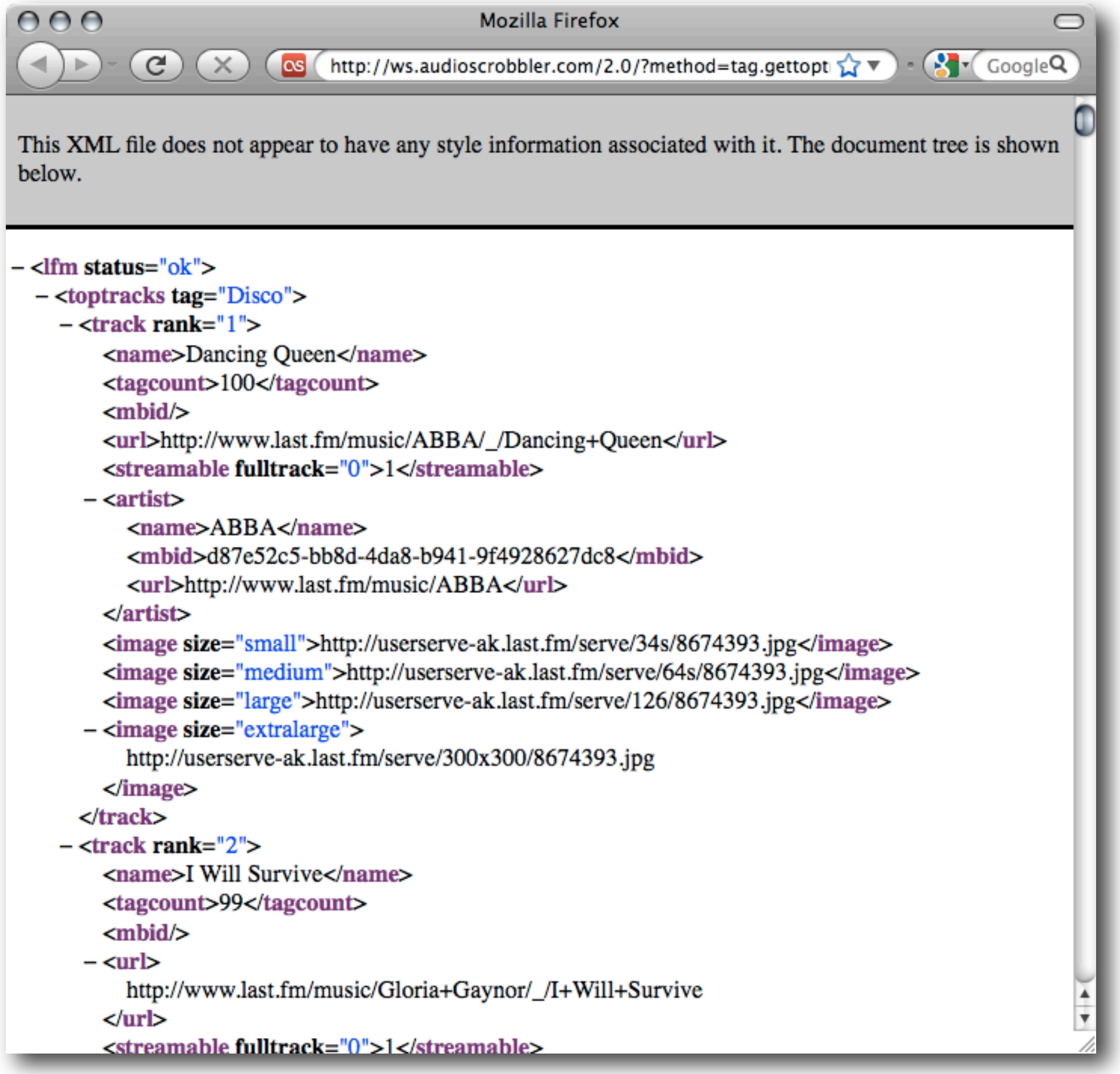
Popular | Hyped | Popular in Spain

Johnny Cash
40,139,139 plays (1,183,846 listeners)
Similar to: Johnny Cash & June Carter, The Highwaymen, Bob Dylan & Johnny Cash, Johnny Cash & Willie Nelson, Willie Nelson

ON TOUR
Taylor Swift
12,106,618 plays (309,901 listeners)
Similar to: Kellie Pickler, Carrie Underwood, Gloriana, Ashley Tisdale, Julianne Hough

Dixie Chicks
5,158,011 plays (265,884 listeners)
Similar to: Faith Hill, Trisha Yearwood, Martina McBride, Miranda Lambert, Gretchen Wilson

ON TOUR
Willie Nelson



Mozilla Firefox

http://ws.audioscrobbler.com/2.0/?method=tag.gettopt

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <lfm status="ok">
  – <toptracks tag="Disco">
    – <track rank="1">
        <name>Dancing Queen</name>
        <tagcount>100</tagcount>
        <mbid/>
        <url>http://www.last.fm/music/ABBA/_/Dancing+Queen</url>
        <streamable fulltrack="0">1</streamable>
      – <artist>
          <name>ABBA</name>
          <mbid>d87e52c5-bb8d-4da8-b941-9f4928627dc8</mbid>
          <url>http://www.last.fm/music/ABBA</url>
        </artist>
        <image size="small">http://userserve-ak.last.fm/serve/34s/8674393.jpg</image>
        <image size="medium">http://userserve-ak.last.fm/serve/64s/8674393.jpg</image>
        <image size="large">http://userserve-ak.last.fm/serve/126/8674393.jpg</image>
      – <image size="extralarge">
          http://userserve-ak.last.fm/serve/300x300/8674393.jpg
        </image>
      </track>
    – <track rank="2">
        <name>I Will Survive</name>
        <tagcount>99</tagcount>
        <mbid/>
      – <url>
          http://www.last.fm/music/Gloria+Gaynor/_/I+Will+Survive
        </url>
        <streamable fulltrack="0">1</streamable>
```

last.fm/music/+tag/country

ws.audioscrobbler.com/2.0/?method=tag.gettoptracks&tag=disco& api_key=*KEY*

Combining two music web APIs

The code is included in the file **c/lyricsfly_3.rb**:

```
require 'net/http'
require 'rexml/document'
require "#{File.dirname(__FILE__)}/lyricsfly_key"
require "#{File.dirname(__FILE__)}/lastfm_key"

def get_lyrics(artist_and_title)
  artist,title = artist_and_title.collect{|arg|
    arg.gsub(/[^a-zA-Z0-9]/,'%25')}
  url =  "http://lyricsfly.com/api/api.php?"
  url += "a=#{artist}&t=#{title}&i=#{$lyricsfly_key}"
  result = Net::HTTP.get_response(URI.parse(url))
  response = REXML::Document.new(result.body).elements['//tx']
  response.text.gsub("[br]", "") unless response.nil?
end
```

Combining two music web APIs

```
def get_artists_and_titles(genre)
 url =  "http://ws.audioscrobbler.com/2.0/?method="
 url += "tag.gettoptracks&tag=#{genre}&api_key=#{$lastfm_key}"
 result = Net::HTTP.get_response(URI.parse(url))
 response = REXML::Document.new(result.body)
 response.elements.collect('//track') do |track| [
   track.elements['artist'].elements['name'].text,
track.elements['name'].text ] end unless response.nil?
end

ARGV.each do |genre|
  tracks = get_artists_and_titles(genre)
  lyrics = tracks.collect{|track| get_lyrics(track)}.compact
  qm = lyrics.inject(0.0) {|qm, lyric| qm + lyric.count("?")}
  p "#{genre} avg question marks: %.2f" % (qm/lyrics.length)
end
```

Finally: `$ ruby lyricsfly_3.rb country hip-hop`

Lessons learnt

Hip-hop lyrics have more “**?**” than Country ones

Any **programming language** with libraries to retrieve pages and parse XML can do the work

Data from different web APIs can be **aggregated**

A **mash-up application** can uncover hidden musical relationships among different domains

From instance to concept

There is no limit to the **chain** of API calls

To connect even more resources, **unique identifiers** work better than ambiguous *names*

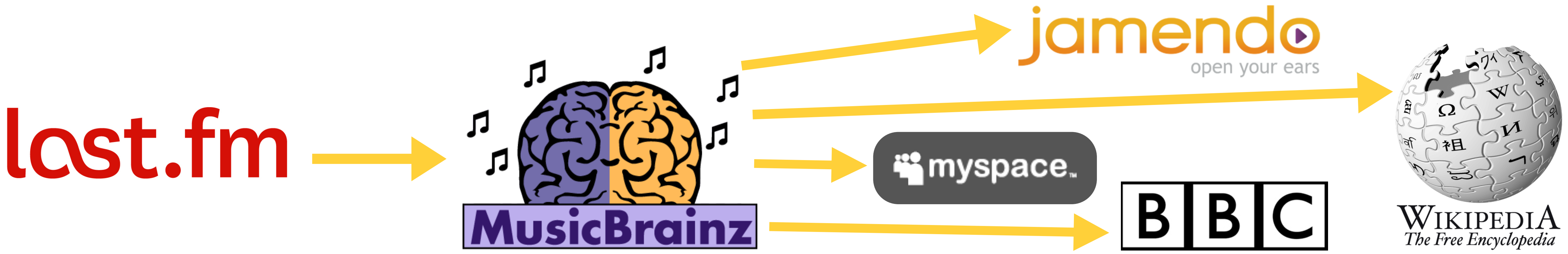
Many web sites identify musical objects through a specific set of **Musicbrainz IDs** which allow to easily match the same item in multiple places

From instance to concept

There is no limit to the **chain** of API calls

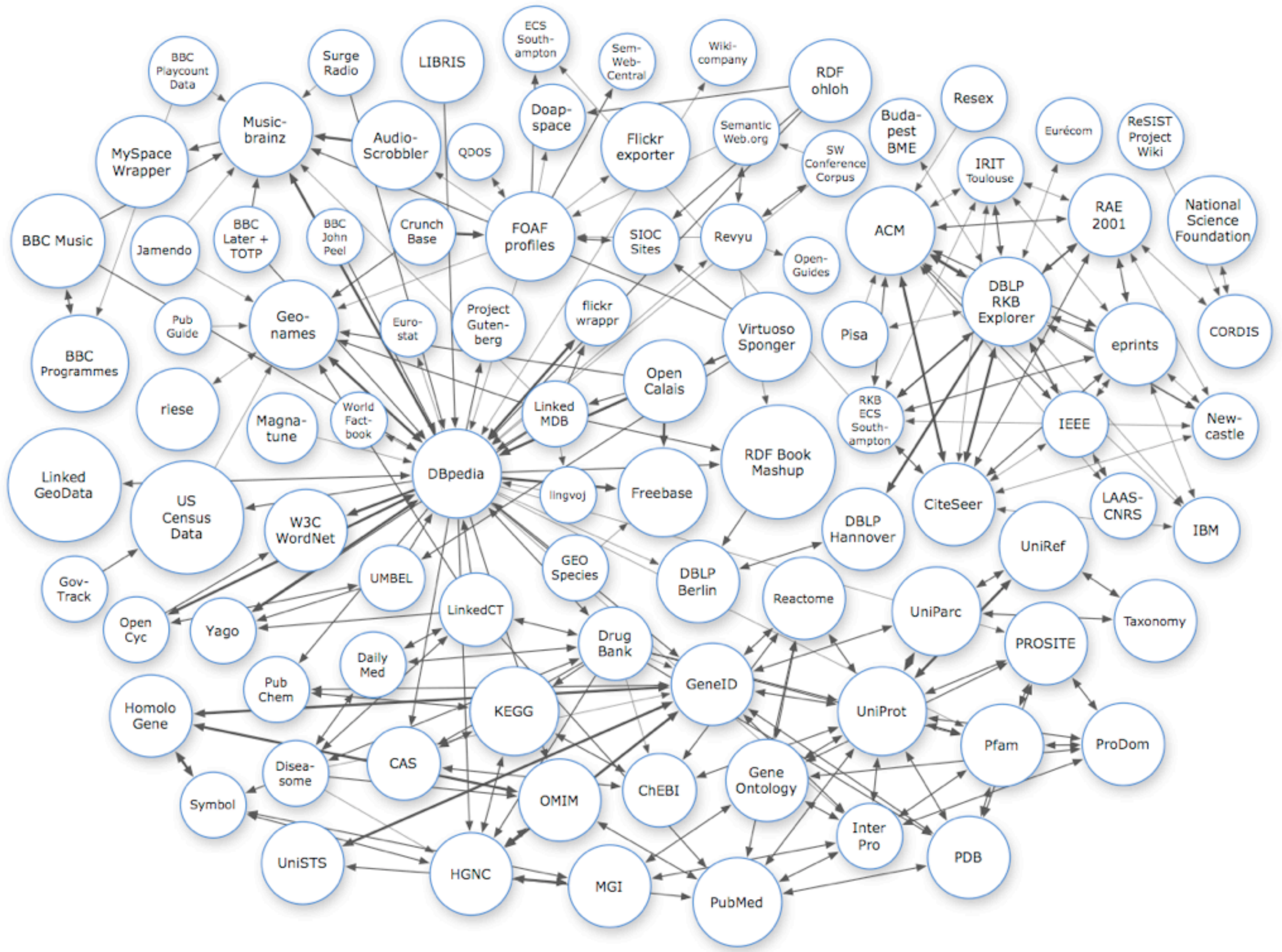
To connect even more resources, **unique identifiers** work better than ambiguous *names*

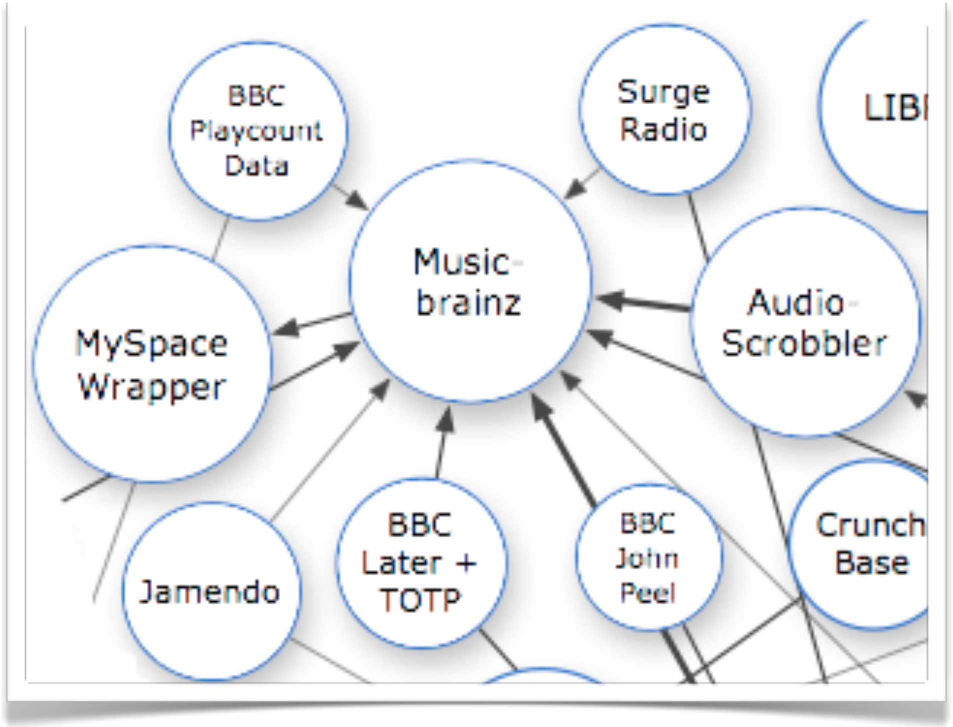
Many web sites identify musical objects through a specific set of **Musicbrainz IDs** which allow to easily match the same item in multiple places



Music and web ontologies

The **Linking Open Data** project is a prominent attempt at expressing and **connecting objects** of different domains using **semantic web** technology

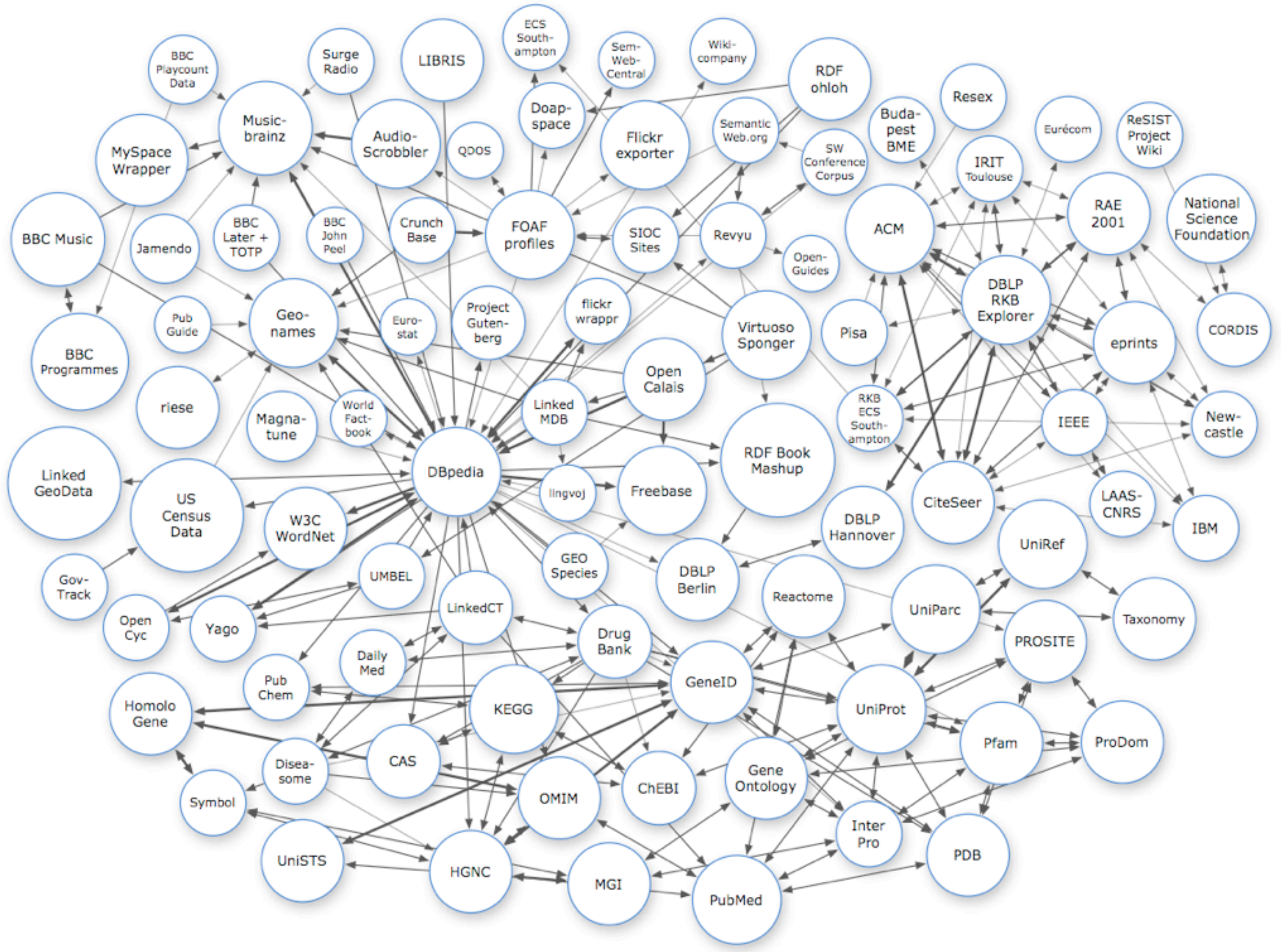


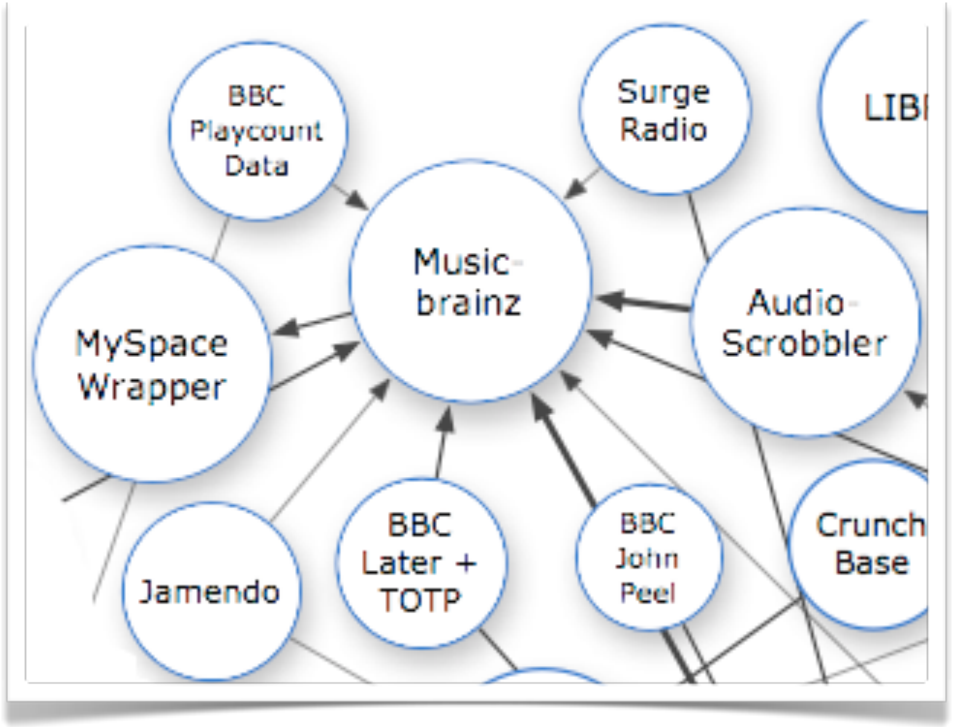


BBC myspace last.fm jamendo open your ears MusicBrainz

Music and web ontologies

The **Linking Open Data** project is a prominent attempt at expressing and **connecting objects** of different domains using **semantic web** technology





BBC

myspace

last.fm

jamendo open your ears

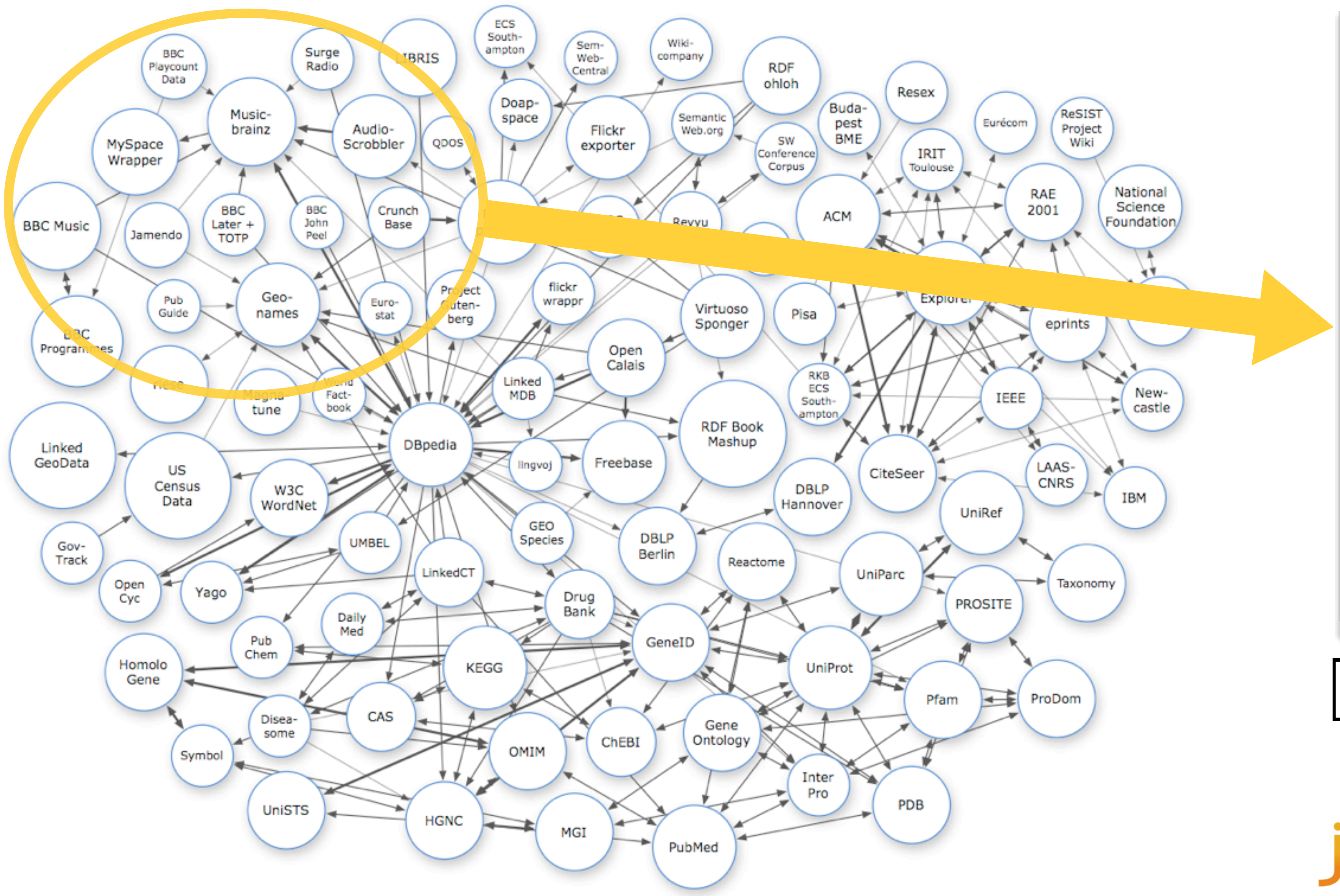
MusicBrainz

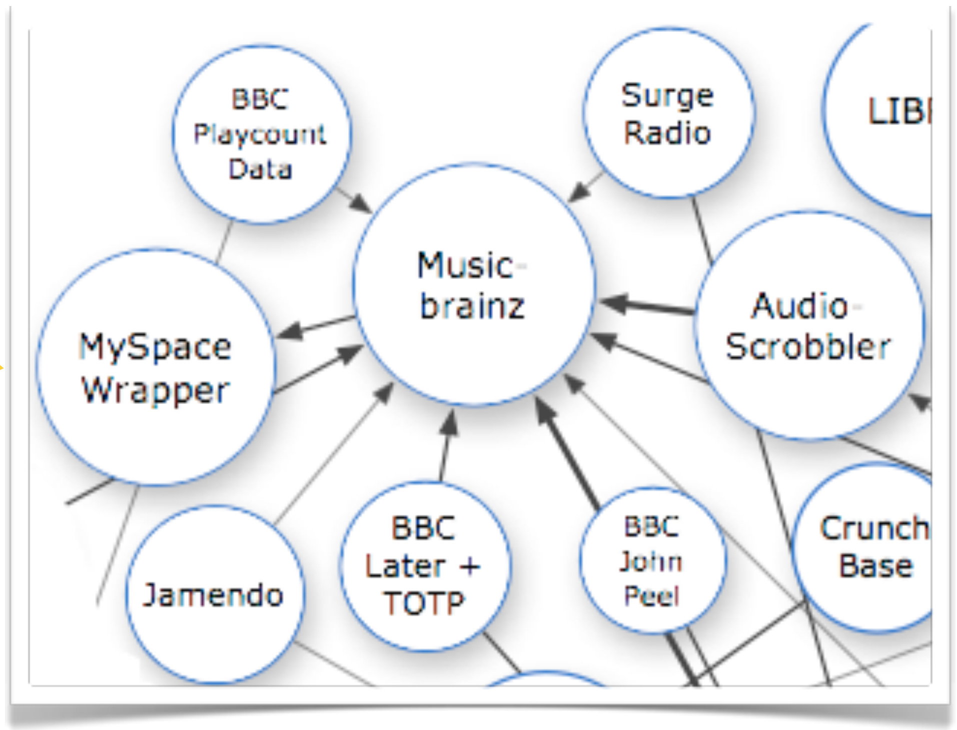
linkeddata.org

musicontology.com

Music and web ontologies

The **Linking Open Data** project is a prominent attempt at expressing and **connecting objects** of different domains using **semantic web** technology





BBC myspace last.fm jamendo open your ears MusicBrainz

linkeddata.org   sameas.org   musicontology.com

QUESTIONS?

First break
10 minutes

```
$ wget http://peak.telecommunity.com/dist/ez_setup.py
$ sudo python ez_setup.py
$ easy_install pylast
```

#3

PERFORMING AUDIO ANALYSIS

The web as a source of tools

The web as a source of tools

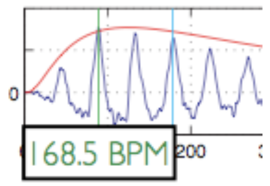
How do you extract **acoustic features** of a song?

The web as a source of tools

How do you extract **acoustic features** of a song?

1. Write your own code:

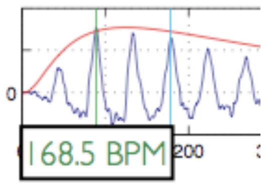
MARSYAS
MUSIC ANALYSIS, RETRIEVAL AND SYNTHESIS FOR AUDIO SIGNALS



The web as a source of tools

How do you extract **acoustic features** of a song?

1. Write your own code: MARSYAS MUSIC ANALYSIS, RETRIEVAL AND SYNTHESIS FOR AUDIO SIGNALS



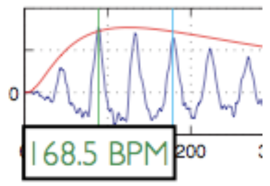
2. Use a software package: python

The web as a source of tools

How do you extract **acoustic features** of a song?

1. Write your own code:

MARSYAS
MUSIC ANALYSIS, RETRIEVAL AND SYNTHESIS FOR AUDIO SIGNALS



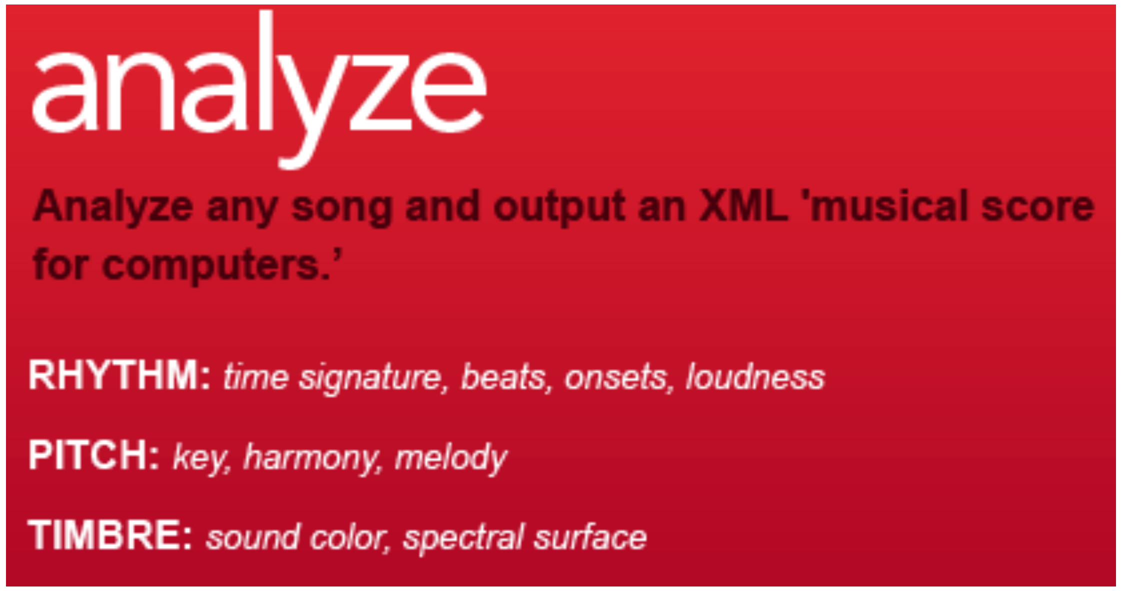
2. Use a software package:

python

3. Retrieve from a web site:

the echonest

echonest.com/analyze



analyze

Analyze any song and output an XML 'musical score for computers.'

RHYTHM: *time signature, beats, onsets, loudness*

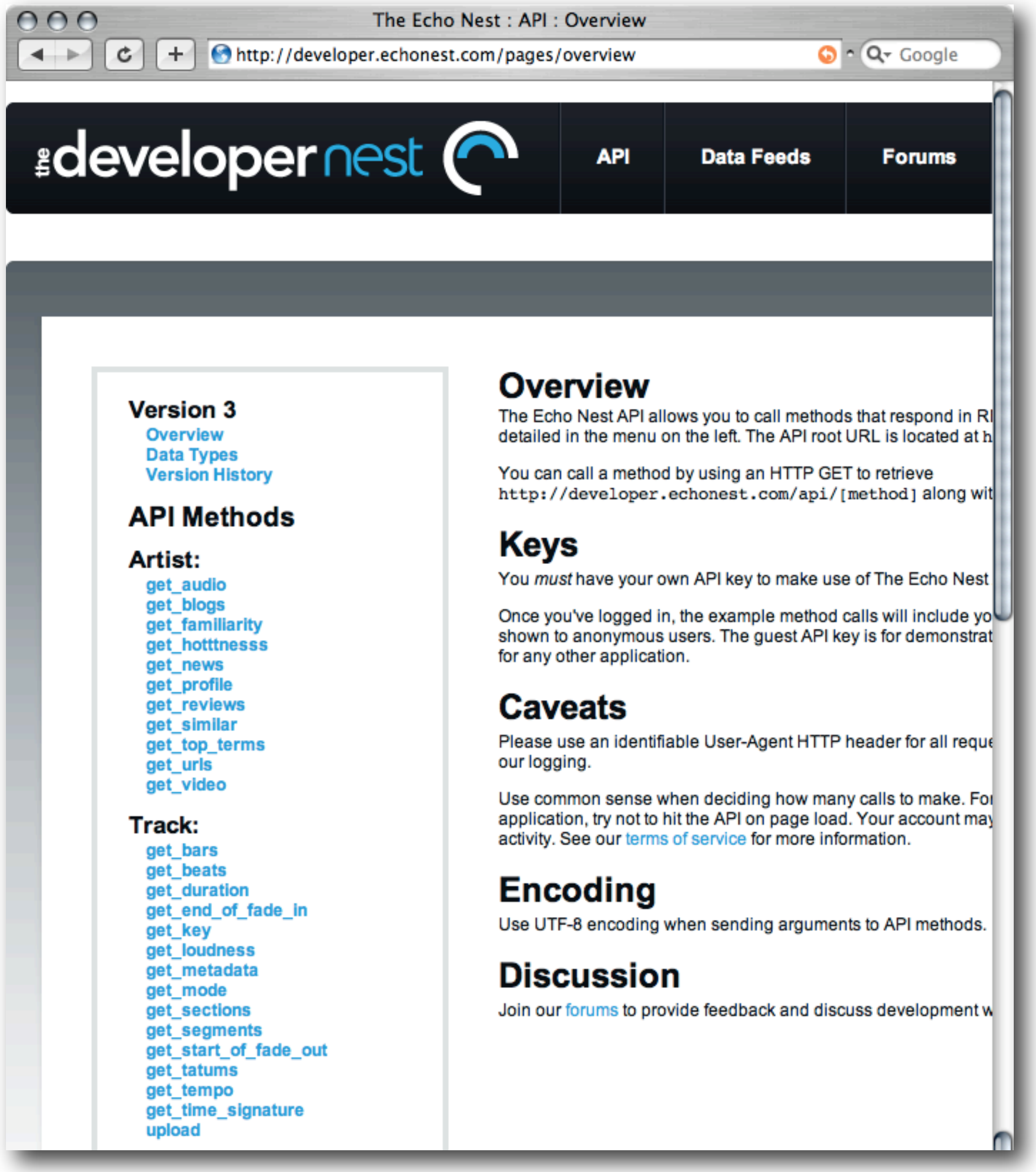
PITCH: *key, harmony, melody*

TIMBRE: *sound color, spectral surface*

Estimating the tempo of a song

Estimating the tempo of a song

Acoustic analysis performed through a web API:



The Echo Nest : API : Overview
http://developer.echonest.com/pages/overview
Google
the developer nest
API
Data Feeds
Forums

Version 3
Overview
Data Types
Version History

API Methods

Artist:
get_audio
get_blogs
get_familiarity
get_hotttnesss
get_news
get_profile
get_reviews
get_similar
get_top_terms
get_urls
get_video

Track:
get_bars
get_beats
get_duration
get_end_of_fade_in
get_key
get_loudness
get_metadata
get_mode
get_sections
get_segments
get_start_of_fade_out
get_tatums
get_tempo
get_time_signature
upload

Overview

The Echo Nest API allows you to call methods that respond in Rl
detailed in the menu on the left. The API root URL is located at h

You can call a method by using an HTTP GET to retrieve
`http://developer.echonest.com/api/[method]` along wit

Keys

You *must* have your own API key to make use of The Echo Nest

Once you've logged in, the example method calls will include yo
shown to anonymous users. The guest API key is for demonstrat
for any other application.

Caveats

Please use an identifiable User-Agent HTTP header for all reque
our logging.

Use common sense when deciding how many calls to make. Fo
application, try not to hit the API on page load. Your account may
activity. See our terms of service for more information.

Encoding

Use UTF-8 encoding when sending arguments to API methods.

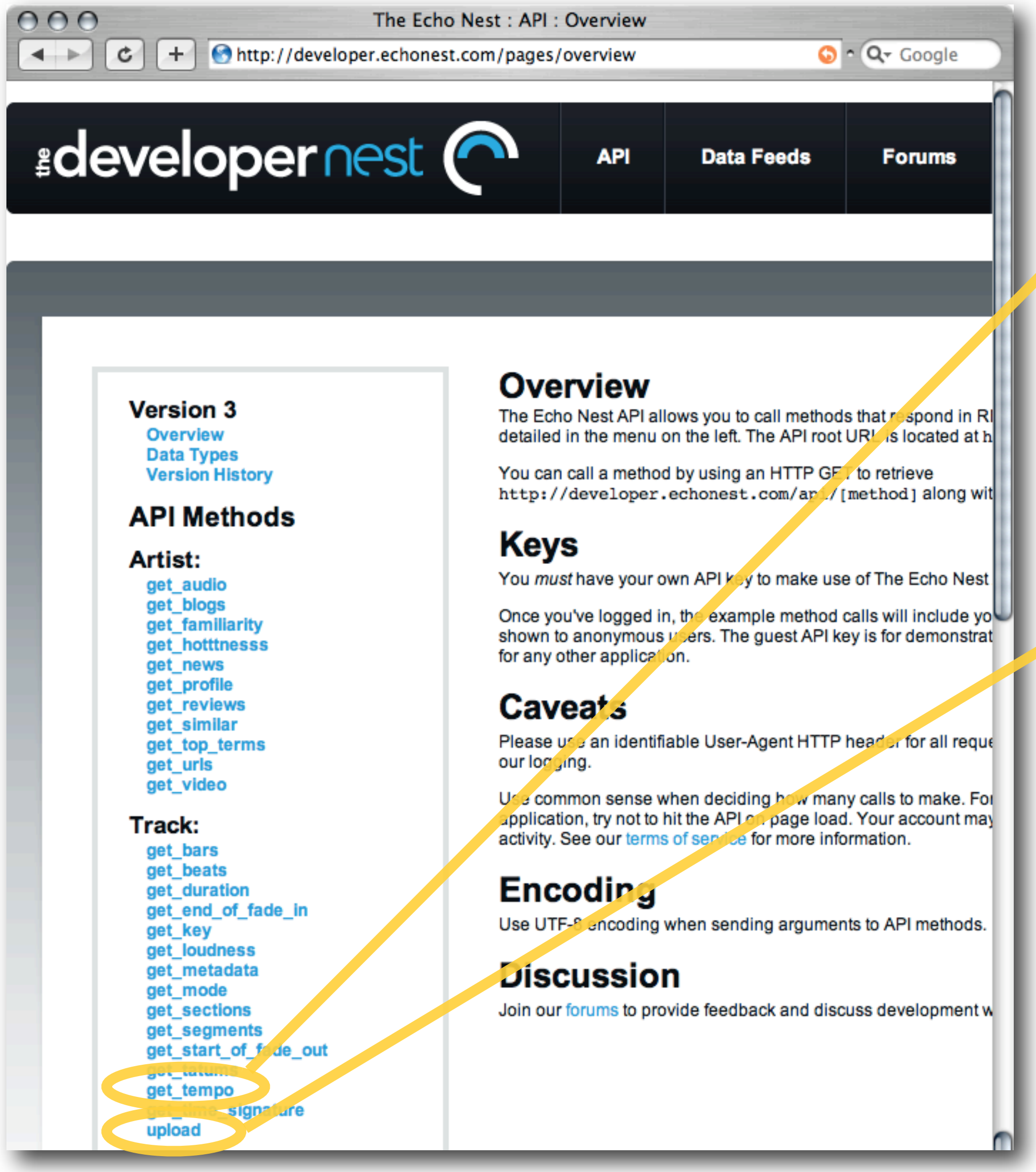
Discussion

Join our forums to provide feedback and discuss development w

developer.echonest.com/
pages/overview

Estimating the tempo of a song

Acoustic analysis performed through a web API:



The Echo Nest : API : Overview
http://developer.echonest.com/pages/overview
Google
the developer nest
API
Data Feeds
Forums

Version 3
- Overview
- Data Types
- Version History

API Methods

Artist:
- get_audio
- get_blogs
- get_familiarity
- get_hotttnesss
- get_news
- get_profile
- get_reviews
- get_similar
- get_top_terms
- get_urls
- get_video

Track:
- get_bars
- get_beats
- get_duration
- get_end_of_fade_in
- get_key
- get_loudness
- get_metadata
- get_mode
- get_sections
- get_segments
- get_start_of_fade_out
- get_tatums
- get_tempo
- get_time_signature
- upload

Overview

The Echo Nest API allows you to call methods that respond in RI
detailed in the menu on the left. The API root URL is located at h

You can call a method by using an HTTP GET to retrieve
`http://developer.echonest.com/api/[method]` along wit

Keys

You *must* have your own API key to make use of The Echo Nest

Once you've logged in, the example method calls will include yo
shown to anonymous users. The guest API key is for demonstrat
for any other application.

Caveats

Please use an identifiable User-Agent HTTP header for all reque
our logging.

Use common sense when deciding how many calls to make. Fo
application, try not to hit the API on page load. Your account may
activity. See our terms of service for more information.

Encoding

Use UTF-8 encoding when sending arguments to API methods.

Discussion

Join our forums to provide feedback and discuss development w

upload: '*Upload a track to The Echo Nest's analyzer for analysis and later retrieval of track information*'

get_tempo: '*Retrieve the overall estimated tempo of a track in beats per minute after previously calling for analysis via upload*'

developer.echonest.com/pages/overview

Authentication is required

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

```
$ irb
```

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

```
$ irb
require 'net/http'
require 'rexml/document'
require 'echonest_key'
```

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

```
$ irb
require 'net/http'
require 'rexml/document'
require 'echonest_key'
song = 'http://ismir2009.benfields.net/m/120bpm.mp3'
url= 'http://developer.echonest.com/api/upload'
result = Net::HTTP.post_form(URI.parse(url),
{'api_key' => $echonest_key, 'version' => '3',
'url' => song})
```

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

```
song_id = REXML::Document.new
(result.body).elements['//track'].attributes['id']
```

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

```
song_id = REXML::Document.new
(result.body).elements['//track'].attributes['id']

url = 'http://developer.echonest.com/api/get_tempo'

url+= "?id=#{song_id}"

url+= "&version=3&api_key=#{$echonest_key}"

result = Net::HTTP.get_response(URI.parse(url))
```

Creating a Ruby script

Estimate tempo for the track **m/120bpm.mp3**:

```
song_id = REXML::Document.new
(result.body).elements['//track'].attributes['id']

url = 'http://developer.echonest.com/api/get_tempo'

url+= "?id=#{song_id}"

url+= "&version=3&api_key=#{$echonest_key}"

result = Net::HTTP.get_response(URI.parse(url))

tempo = REXML::Document.new(result.body).elements
['//tempo'].text

puts "The estimated tempo is #{tempo} BPM"
```

Complete audio analysis

Complete audio analysis

The code is included in the file **c/echonest_1.rb**:

```
$ ruby echonest_1.rb
```

```
=» The estimated tempo
is 120.013 BPM
```

Complete audio analysis

The code is included in the file **c/echonest_1.rb**:

```
$ ruby echonest_1.rb
```

```
=» The estimated tempo
is 120.013 BPM
```

The script **c/echonest_2.rb** allows to specify the track location and estimates more features:

```
$ ruby echonest_2.rb
http://
ismir2009.benfields.net
/m/120bpm.mp3
```

```
=» "time_signature"=> 4,
"mode"=> 1, "key"=> 5,
"tempo"=> 120
```

developer.echonest.com/forums/thread/9

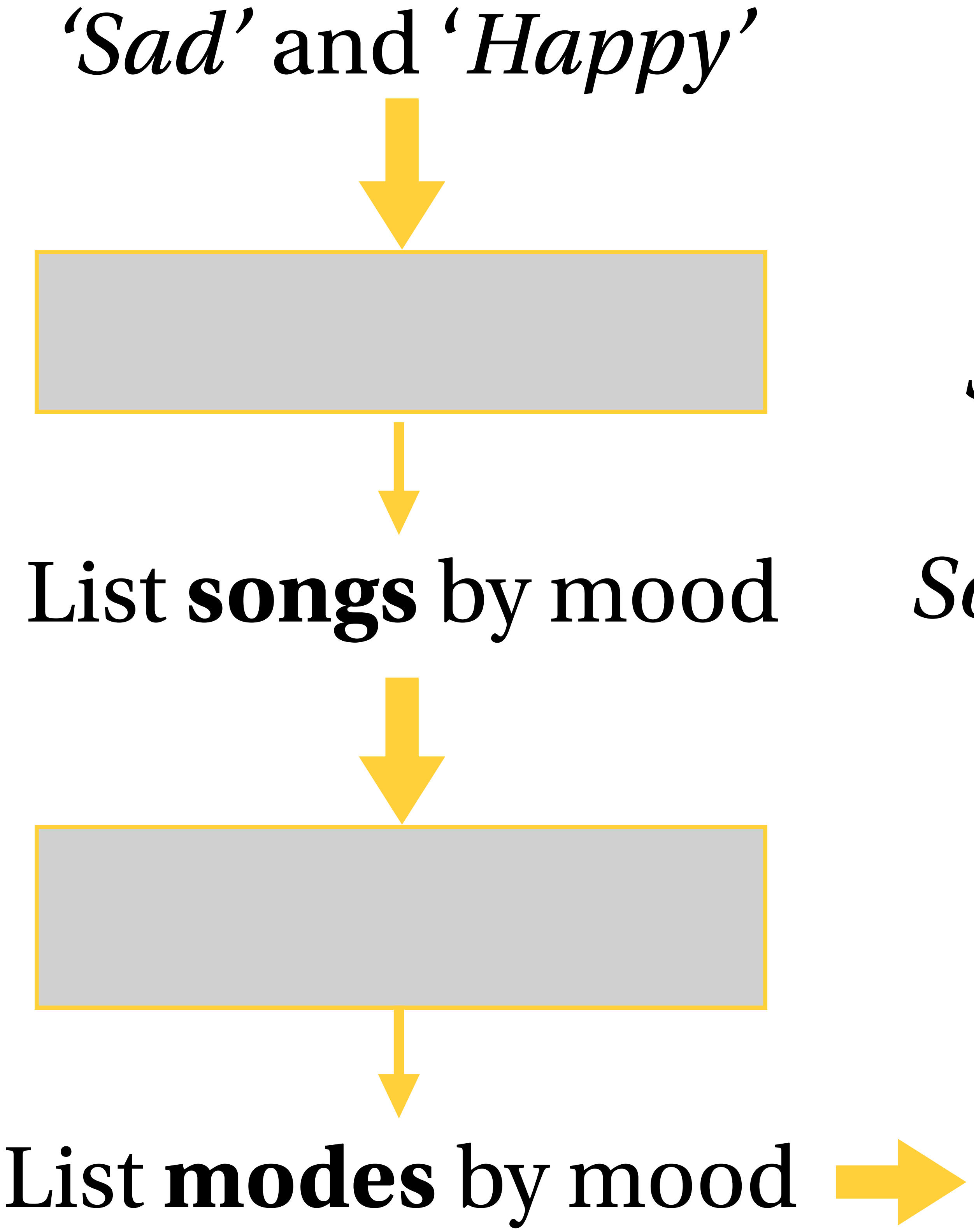
Minor/major vs. sad/happy

Songs in minor are sad

Songs in major are happy

Would you agree?

Minor/major vs. sad/happy



Songs in minor are sad

Songs in major are happy

Would you agree?

Compare **minor/major**

Minor/major vs. sad/happy

‘*Sad*’ and ‘*Happy*’

jamendo
open your ears

List **songs** by mood

the echonest

List **modes** by mood → Compare **minor/major**

Songs in minor are sad

Songs in major are happy

Would you agree?

Running the experiment

Running the experiment

The code is included in the file **c/echonest_3.rb**:

```
$ ruby echonest_3.rb sad happy
```

```
=» sad songs are 0.25 major, 0.75 minor
   happy songs are 1.00 major, 0.00 minor
```

Running the experiment

The code is included in the file **c/echonest_3.rb**:

```
$ ruby echonest_3.rb sad happy
```

```
=» sad songs are 0.25 major, 0.75 minor
   happy songs are 1.00 major, 0.00 minor
```

Repeating the experiment with more songs can serve as a proper **evaluation** of the statement

Running the experiment

The code is included in the file **c/echonest_3.rb**:

```
$ ruby echonest_3.rb sad happy
```

```
=» sad songs are 0.25 major, 0.75 minor
   happy songs are 1.00 major, 0.00 minor
```

Repeating the experiment with more songs can serve as a proper **evaluation** of the statement

Do not submit too many **simultaneous** queries!

Advanced echonest-ing

Even DJs can use web-based tools to **remix**

The echonest python wrapper is available here:

code.google.com/p/echo-nest-remix/

See it in action at donkdj.com (an auto-remixer)

Lessons learnt

The web makes available both musical data and tools for **acoustic analysis**

Symbolic analysis not available… yet?

The future of music software is on the **web**

QUESTIONS?

#4

REVEALING TRENDS

What is trendy?

“Trend” is related to a specific **time** and **context**

Anything that is **rapidly** becoming famous in **your** enviroment (your friends, your location, ...)

Mavens are the first to pick up on nascent trends

Music example: which **artists** should you now be listening to, to keep up with the **latest trends**?

Trendy artist of the month

Trendy artist of the month

your friends

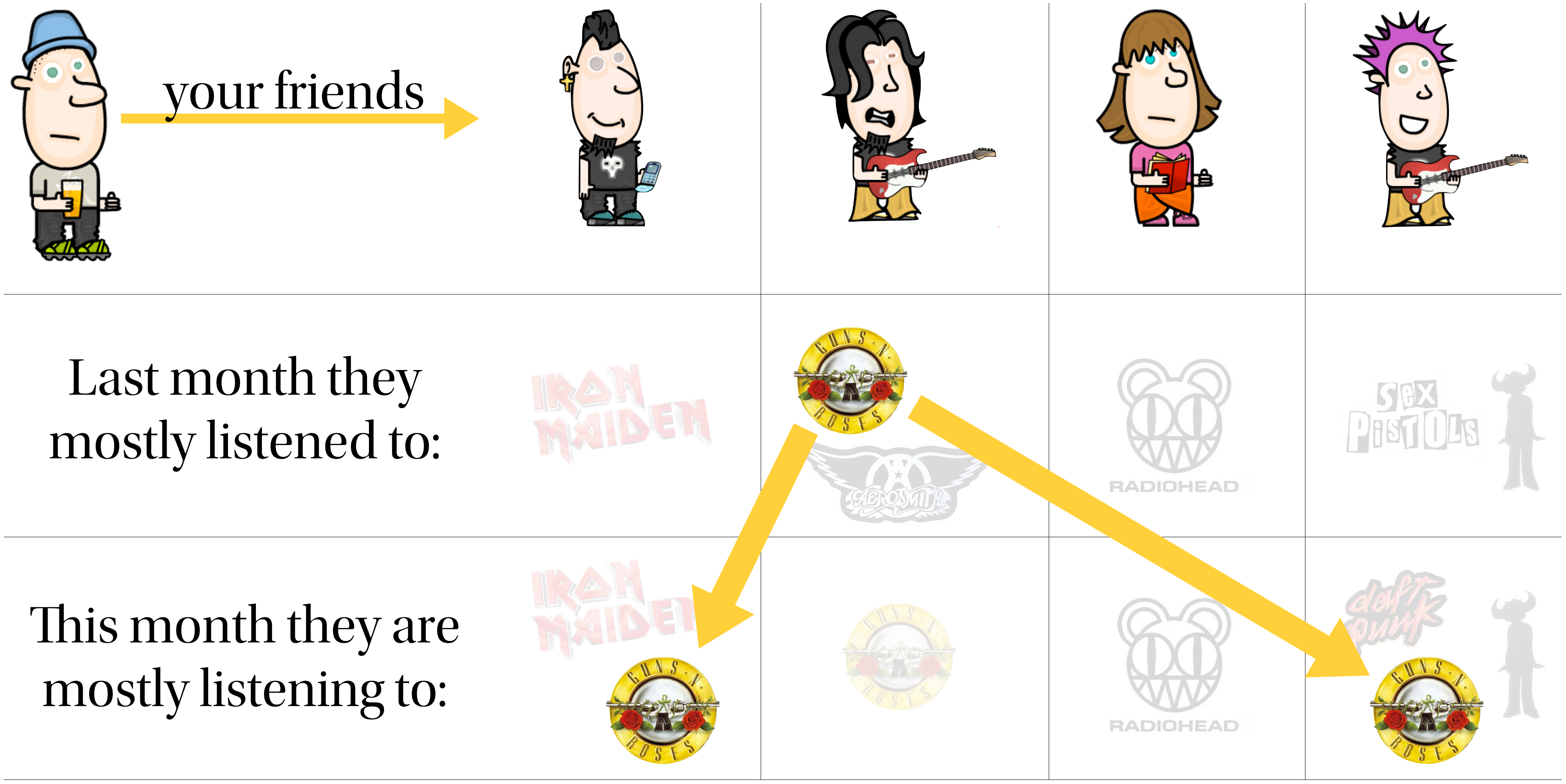
Trendy artist of the month

your friends				
Last month they mostly listened to:	IRON MAIDEN	GUNS N' ROSES AEROSMITH	RADIOHEAD	SEX PISTOLS

Trendy artist of the month

your friends →				
Last month they mostly listened to:	IRON MAIDEN	GUNS N ROSES AEROSMITH	RADIOHEAD	Sex Pistols
This month they are mostly listening to:	IRON MAIDEN GUNS N ROSES	GUNS N ROSES	RADIOHEAD	daft punk GUNS N ROSES

Trendy artist of the month


your friends
Last month they mostly listened to:
This month they are mostly listening to:
RADIOHEAD
RADIOHEAD

This kind of information can be retrieved from music-related web communities such as **Last.fm**

Hiding API calls with wrappers

Last.fm provides for each user the **list of friends** and the **most played** artists in a given period

These data can be retrieved via Last.fm API or, more easily, using the Python **wrapper** *pylast*, available at code.google.com/p/pylast which can be installed through *easy_install*:

```
$ wget http://peak.telecommunity.com/dist/ez_setup.py
$ sudo python ez_setup.py
$ easy_install pylast
```

Retrieving lists of friends

API wrappers **abstract** the functions that make HTTP calls to send and receive information

Using the *pylast* wrapper, the code to obtain lists of friends from Last.fm is **compact** and **clear**:

Retrieving lists of friends

API wrappers **abstract** the functions that make HTTP calls to send and receive information

Using the *pylast* wrapper, the code to obtain lists of friends from Last.fm is **compact** and **clear**:

```
$ python
```

Retrieving lists of friends

API wrappers **abstract** the functions that make HTTP calls to send and receive information

Using the *pylast* wrapper, the code to obtain lists of friends from Last.fm is **compact** and **clear**:

```
$ python
import pylast
from lastfm_key import lastfm_key
```

Retrieving lists of friends

API wrappers **abstract** the functions that make HTTP calls to send and receive information

Using the *pylast* wrapper, the code to obtain lists of friends from Last.fm is **compact** and **clear**:

```
$ python
import pylast
from lastfm_key import lastfm_key
api = pylast.get_lastfm_network(lastfm_key)
friends = api.get_user("claudiob").get_friends()
print "Last.fm friends: %s" % friends
```

Extracting trendy artists of the month

Extracting trendy artists of the month

To reveal which artists a user should listen to:

1. Retrieve the list of friends of that user

2. Retrieve the most played artists by the friends during this and the previous month, printing those who have ‘grown’ more in this period while excluding artists the user is already aware of

Extracting trendy artists of the month

To reveal which artists a user should listen to:

1. Retrieve the list of friends of that user

2. Retrieve the most played artists by the friends during this and the previous month, printing those who have 'grown' more in this period while excluding artists the user is already aware of

The code is included in the file **c/lastfm_2.py**:

```
$ python lastfm_2.py claudiob
```

```
=» Trendy artists for claudiob:
1) Amy Winehouse, already known by daddyrho, recently
discovered by kobra_cccpozzi, pilomatic, econ-luca, ...
```

Lessons learnt

Social data from the web helps uncover trends

Data for trends imply a **temporal** dimension, a **context** (friends, geographical location, ...) and a **class** of objects (artists, tracks, ...) to observe

More **transparent** and 'human' than using collaborative filtering for recommendations

API **wrappers** shorten and clear up the code

QUESTIONS?

Last break
10 minutes

Last break
10 minutes

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz

$ gunzip < scapi.tar.gz | tar xvf -
```

Last break
10 minutes

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
```

Last break
10 minutes

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
$ sudo python setup.py install
```

Last break
10 minutes

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
$ sudo python setup.py install
$ easy_install python-igraph
```

#5

CAPTURING SOCIAL DATA

Social networks for musicians

Web sites for **musician-to-musician** networking

1. Grant access to an artist's public **music**
2. Record **relationships** among musicians in the same network
3. Provide **social data** in the domain of music

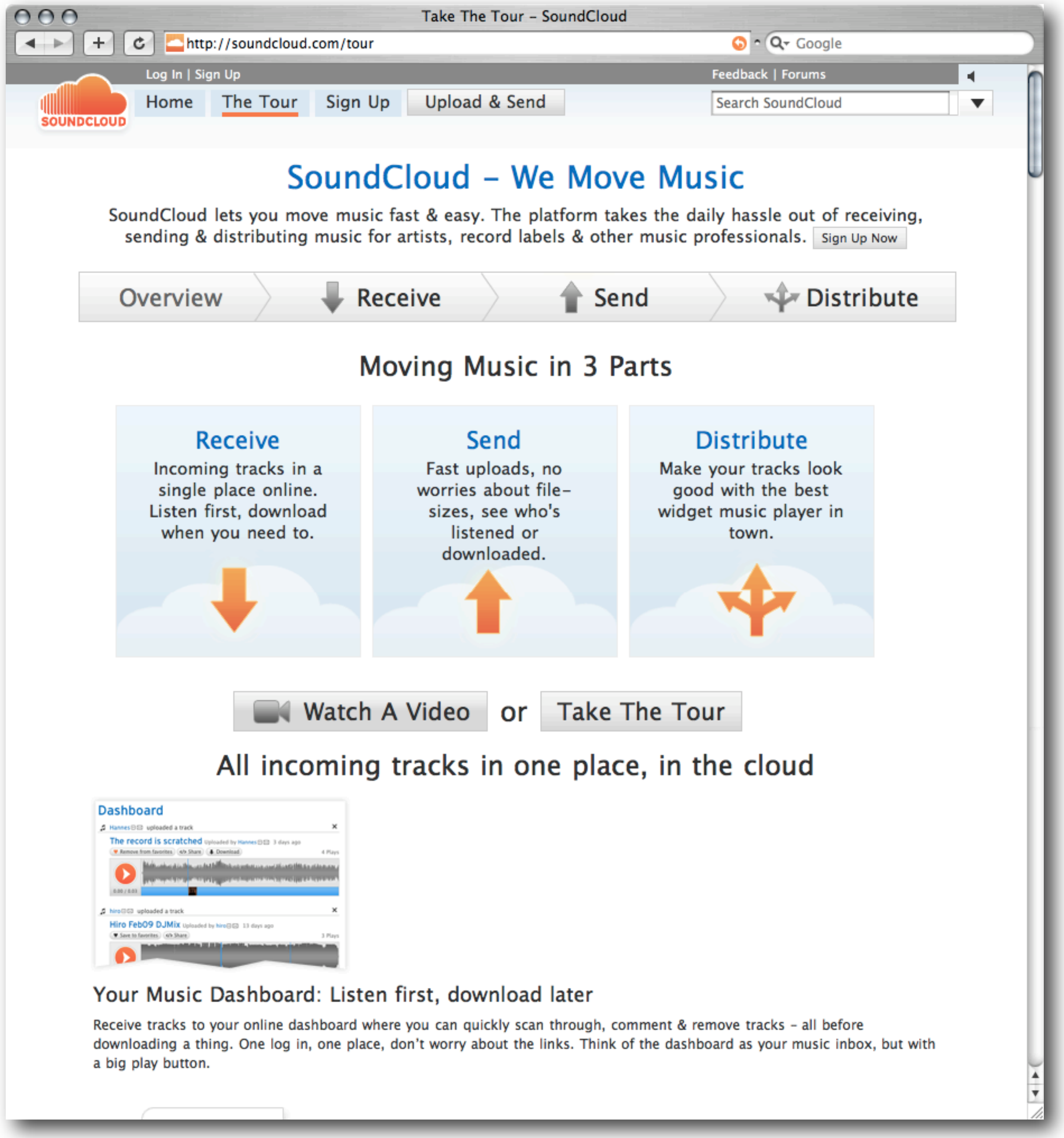
Can be very useful for music informatics

A different kind of musical resource

SoundCloud, an advanced music-sharing platform

A different kind of musical resource

SoundCloud, an advanced music-sharing platform



Take The Tour – SoundCloud

http://soundcloud.com/tour

Log In | Sign Up   Feedback | Forums

Home   The Tour   Sign Up   Upload & Send   Search SoundCloud

SoundCloud – We Move Music

SoundCloud lets you move music fast & easy. The platform takes the daily hassle out of receiving, sending & distributing music for artists, record labels & other music professionals. Sign Up Now

Overview   Receive   Send   Distribute

Moving Music in 3 Parts

Receive Incoming tracks in a single place online. Listen first, download when you need to.

Send Fast uploads, no worries about file-sizes, see who's listened or downloaded.

Distribute Make your tracks look good with the best widget music player in town.

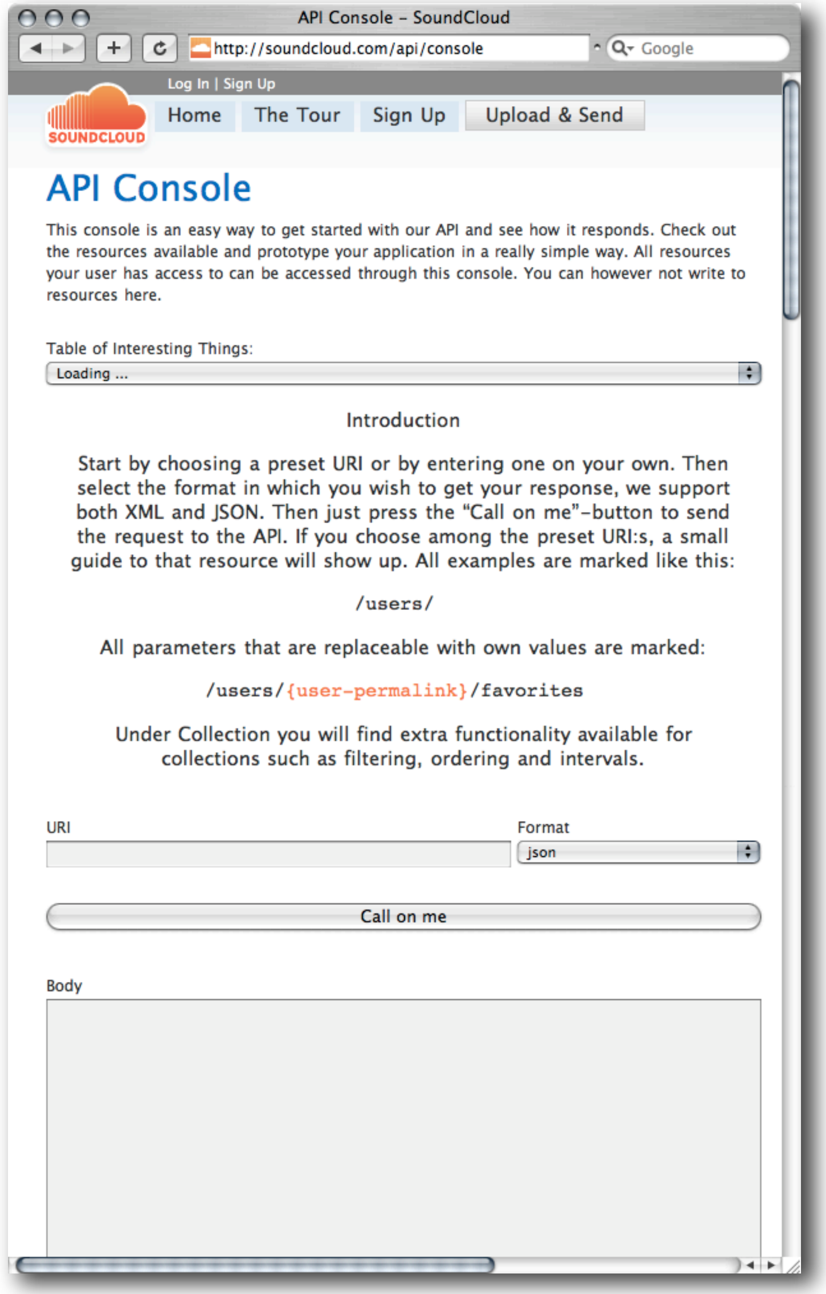
Watch A Video or Take The Tour

All incoming tracks in one place, in the cloud

Dashboard

Your Music Dashboard: Listen first, download later

Receive tracks to your online dashboard where you can quickly scan through, comment & remove tracks – all before downloading a thing. One log in, one place, don't worry about the links. Think of the dashboard as your music inbox, but with a big play button.



API Console – SoundCloud

http://soundcloud.com/api/console

Log In | Sign Up

Home   The Tour   Sign Up   Upload & Send

API Console

This console is an easy way to get started with our API and see how it responds. Check out the resources available and prototype your application in a really simple way. All resources your user has access to can be accessed through this console. You can however not write to resources here.

Table of Interesting Things: Loading ...

Introduction

Start by choosing a preset URI or by entering one on your own. Then select the format in which you wish to get your response, we support both XML and JSON. Then just press the "Call on me"-button to send the request to the API. If you choose among the preset URI:s, a small guide to that resource will show up. All examples are marked like this:

/users/

All parameters that are replaceable with own values are marked:

/users/{user-permalink}/favorites

Under Collection you will find extra functionality available for collections such as filtering, ordering and intervals.

URI   Format: json

Call on me

Body

soundcloud.com   soundcloud.com/api/console

Retrieving lists of followers

Retrieving lists of followers

Install the Python **wrapper** for SoundCloud API:

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz

$ gunzip < scapi.tar.gz | tar xvf -
```

Retrieving lists of followers

Install the Python **wrapper** for SoundCloud API:

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
```

Retrieving lists of followers

Install the Python **wrapper** for SoundCloud API:

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
$ sudo python setup.py install
```

Retrieving lists of followers

Install the Python **wrapper** for SoundCloud API:

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/
master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
$ sudo python setup.py install
```

Code is included in the file **c/soundcloud_1.py**:

```
import scapi
from soundcloud_oauth import init_scope
root = init_scope()
user = root.users("bfields")
for friend in user.followings():
```

Retrieving lists of followers

Install the Python **wrapper** for SoundCloud API:

```
$ wget http://github.com/soundcloud/python-api-wrapper/tarball/master -O scapi.tar.gz
$ gunzip < scapi.tar.gz | tar xvf -
$ cd soundcloud-python-api-wrapper-d34be69
$ sudo python setup.py install
```

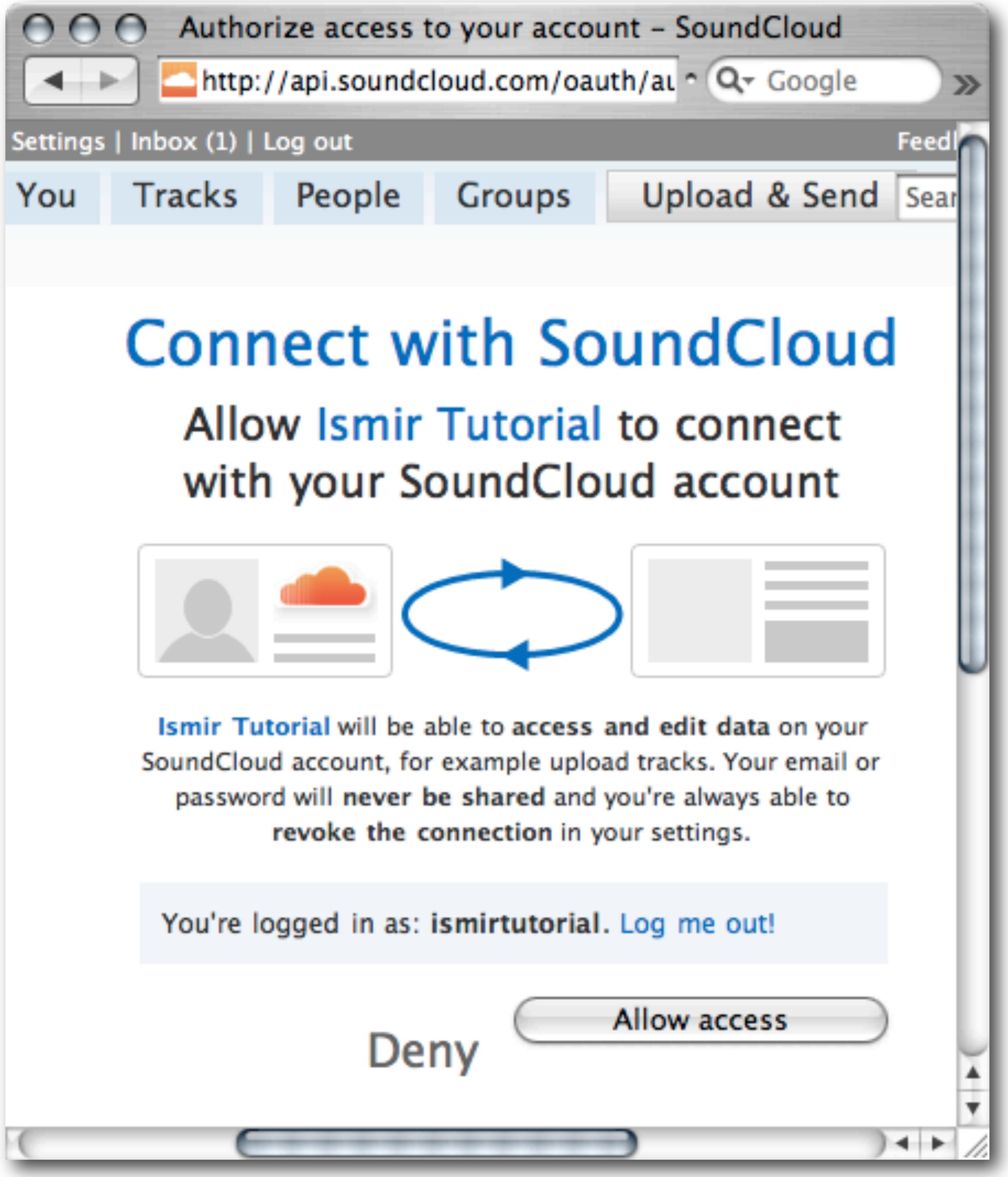
Code is included in the file **c/soundcloud_1.py**:

```
import scapi
from soundcloud_oauth import init_scope
root = init_scope()
user = root.users("bfields")
for friend in user.followings():
    print "Following %s" % friend["username"]
```

A different type of authentication

A different type of authentication

SoundCloud authenticates with **OAuth** protocol:


Authorize access to your account – SoundCloud
http://api.soundcloud.com/oauth/au
Google
Settings | Inbox (1) | Log out
Feed
You
Tracks
People
Groups
Upload & Send
Sear
Deny
Allow access

Connect with SoundCloud

Allow Ismir Tutorial to connect with your SoundCloud account

Ismir Tutorial will be able to **access and edit data** on your SoundCloud account, for example upload tracks. Your email or password will **never be shared** and you're always able to **revoke the connection** in your settings.

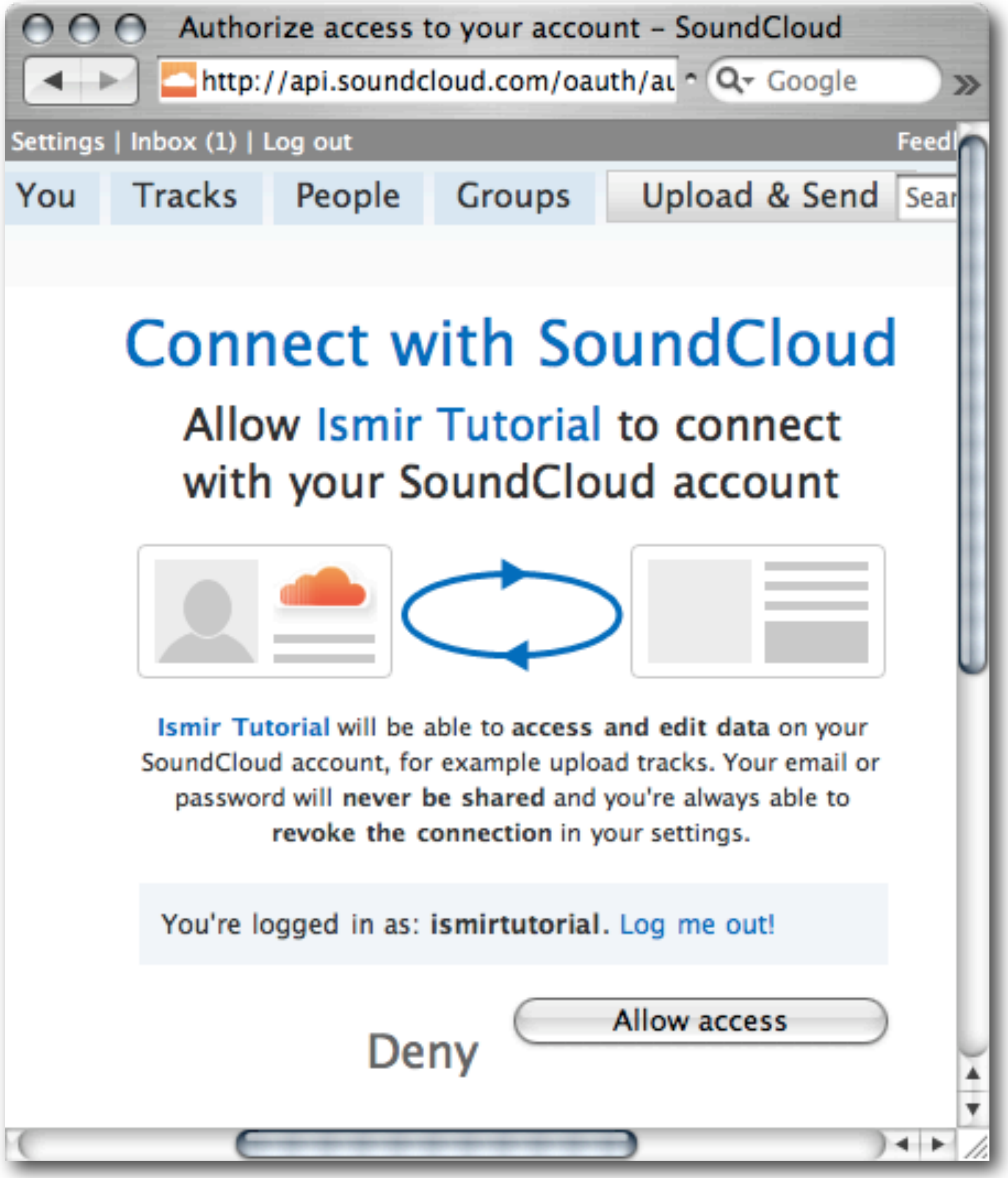
You're logged in as: **ismirtutorial**. Log me out!

c/soundcloud_1.py

runs the full protocol

A different type of authentication

SoundCloud authenticates with **OAuth** protocol:


Authorize access to your account – SoundCloud

http://api.soundcloud.com/oauth/au

Settings | Inbox (1) | Log out

Feed

You | Tracks | People | Groups | Upload & Send

Connect with SoundCloud

Allow Ismir Tutorial to connect with your SoundCloud account

Ismir Tutorial will be able to access and edit data on your SoundCloud account, for example upload tracks. Your email or password will never be shared and you're always able to revoke the connection in your settings.

You're logged in as: ismirtutorial. Log me out!

Deny

Allow access

c/soundcloud_1.py
runs the full protocol

c/soundcloud_2.py
includes a valid token only for this application

```
$ python soundcloud_2.py
```

=» bfields is following: Forss, atl, stunna, ...

Plotting networks of friends

```
$ easy_install python-igraph
```

Plotting networks of friends

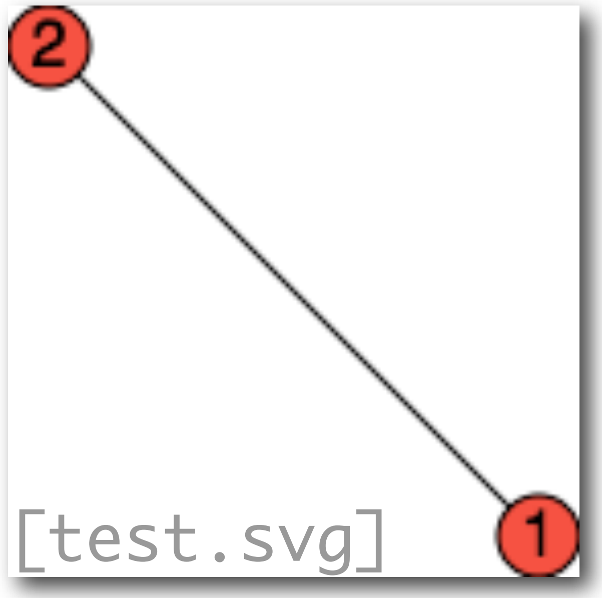
Install the **igraph** library from igraph.sf.net/download.html and the Python **wrapper**:

```
$ easy_install python-igraph
```

and test by drawing a simple graph into a file:

```
$ python
import igraph
g = igraph.Graph(n=2, edges=[(0,1)])
g.write_svg("test.svg", g.layout("kk"))
```

=»



Plotting networks of friends

Plotting networks of friends

The code included in the file **c/soundcloud_3.py**:

1. Adds a *vertex* for a given seed **user**
2. Gets from SoundCloud the list of people the **user** "follows"
3. For each of these persons:
   - Recursively restart from 1. until the desired level of **depth**
   - Adds an *edge* to connect the person to the seed **user**

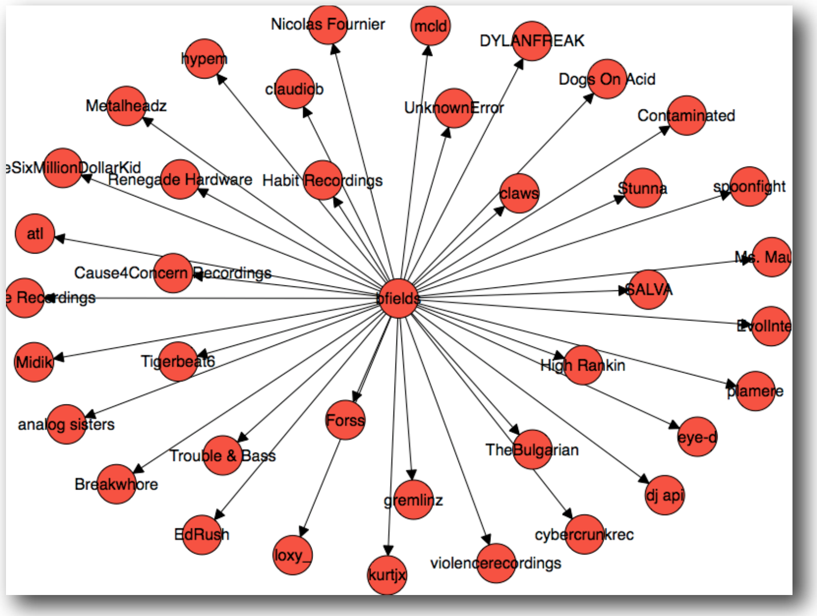
Plotting networks of friends

The code included in the file **c/soundcloud_3.py**:

1. Adds a *vertex* for a given seed **user**
2. Gets from SoundCloud the list of people the **user** “follows”
3. For each of these persons:
   - Recursively restart from 1. until the desired level of **depth**
   - Adds an *edge* to connect the person to the seed **user**

```
$ python soundcloud_3.py bfields 2
```

=»



Plotting networks of friends

The code included in the file **c/soundcloud_3.py**:

1. Adds a *vertex* for a given seed **user**
2. Gets from SoundCloud the list of people the **user** "follows"
3. For each of these persons:
   - Recursively restart from 1. until the desired level of **depth**
   - Adds an *edge* to connect the person to the seed **user**

Plotting networks of friends

The code included in the file **c/soundcloud_3.py**:

1. Adds a *vertex* for a given seed **user**
2. Gets from SoundCloud the list of people the **user** “follows”
3. For each of these persons:
   - Recursively restart from 1. until the desired level of **depth**
   - Adds an *edge* to connect to the person to the seed **user**

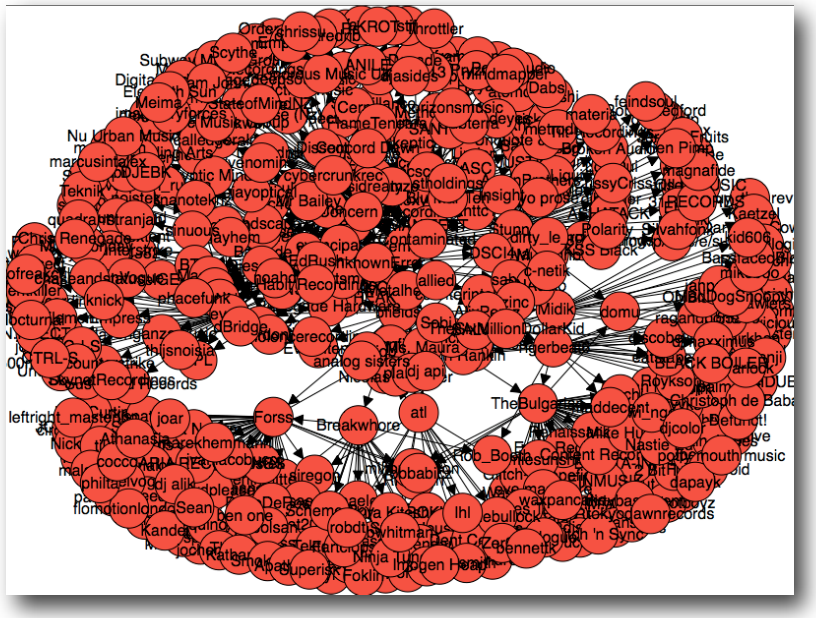
Plotting networks of friends

The code included in the file **c/soundcloud_3.py**:

1. Adds a *vertex* for a given seed **user**
2. Gets from SoundCloud the list of people the **user** "follows"
3. For each of these persons:
   - Recursively restart from 1. until the desired level of **depth**
   - Adds an *edge* to connect to the person to the seed **user**

```
$ python soundcloud_3.py bfields 3
```

=»



Plotting as a web service

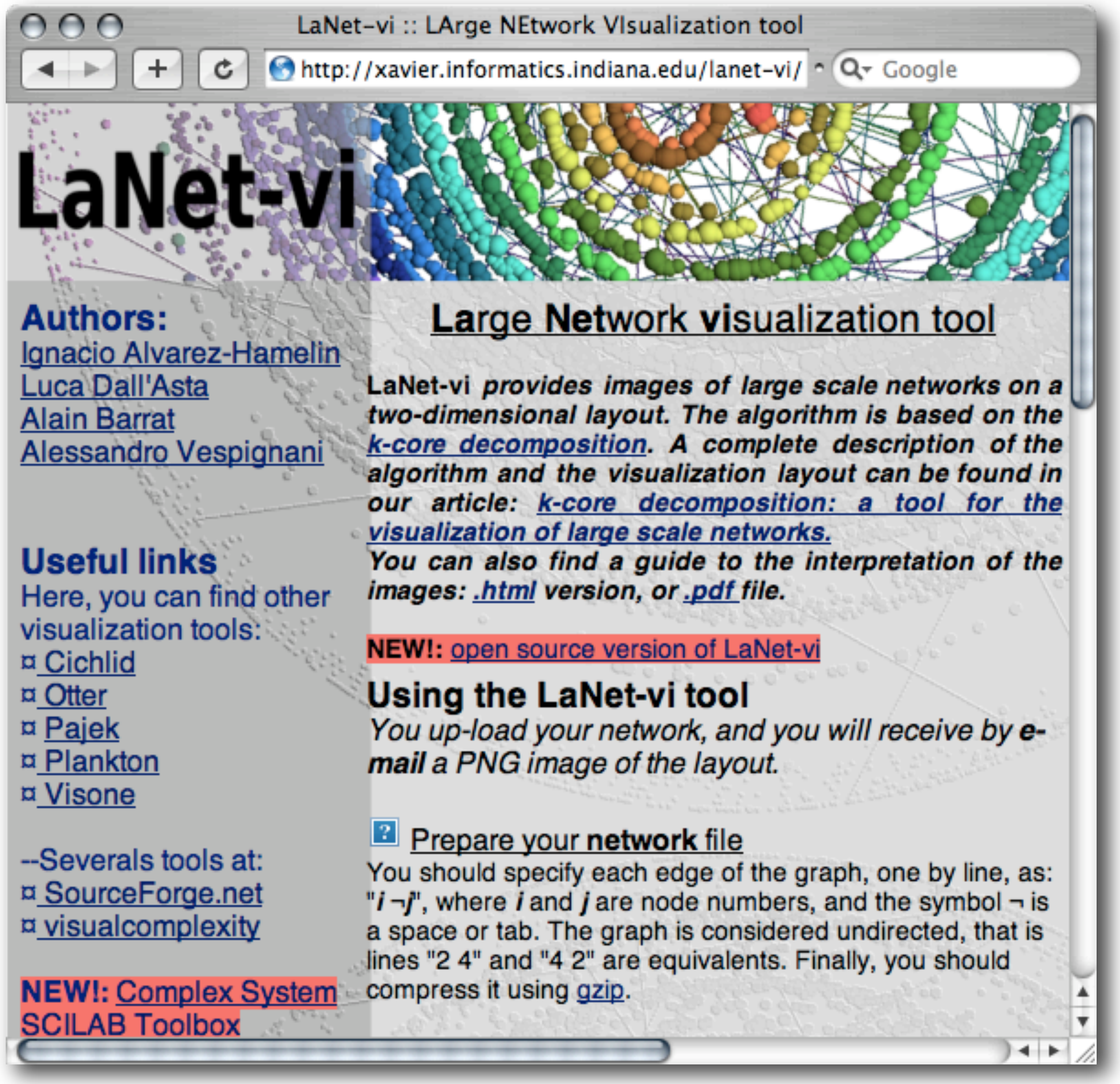
Plotting as a web service

How to plot a snapshot of the full network?

c/soundcloud_4.py
collects the full network
of SoundCloud friends
(takes a long time!)

Plotting as a web service

How to plot a snapshot of the full network?



LaNet-vi :: LArge NEtwork VIsualization tool

http://xavier.informatics.indiana.edu/lanet-vi/

LaNet-vi

Authors:
Ignacio Alvarez-Hamelin
Luca Dall'Asta
Alain Barrat
Alessandro Vespignani

Useful links
Here, you can find other visualization tools:
¤ Cichlid
¤ Otter
¤ Pajek
¤ Plankton
¤ Visone

--Severals tools at:
¤ SourceForge.net
¤ visualcomplexity

NEW!: Complex System SCILAB Toolbox

Large Network visualization tool

LaNet-vi provides images of large scale networks on a two-dimensional layout. The algorithm is based on the k-core decomposition. A complete description of the algorithm and the visualization layout can be found in our article: k-core decomposition: a tool for the visualization of large scale networks. You can also find a guide to the interpretation of the images: .html version, or .pdf file.

NEW!: open source version of LaNet-vi

Using the LaNet-vi tool
*You up-load your network, and you will receive by **e-mail** a PNG image of the layout.*

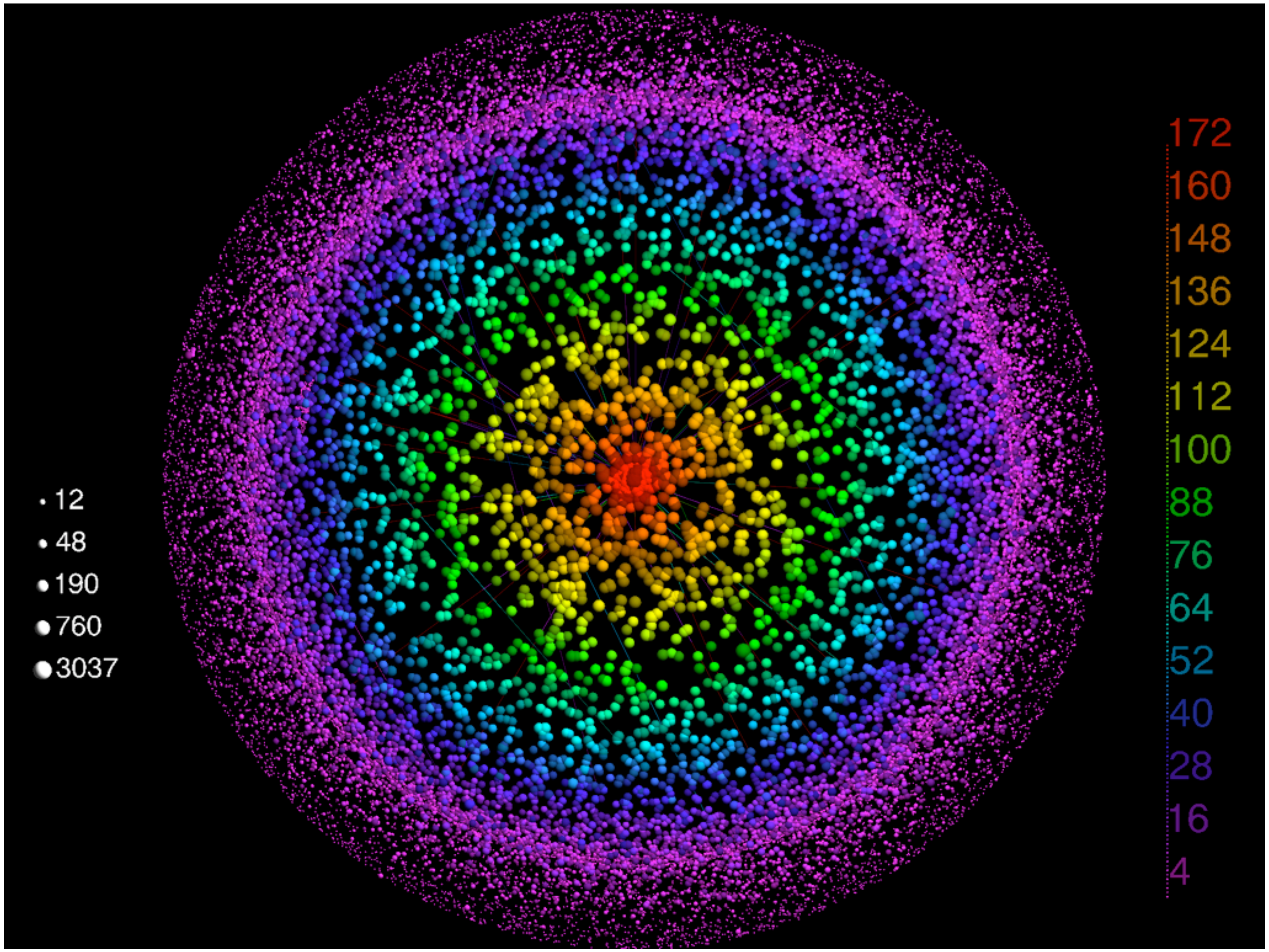
Prepare your network file
You should specify each edge of the graph, one by line, as: "*i* ¬*j*", where *i* and *j* are node numbers, and the symbol ¬ is a space or tab. The graph is considered undirected, that is lines "2 4" and "4 2" are equivalents. Finally, you should compress it using gzip.

xavier.informatics.indiana.edu /lanet-vi/

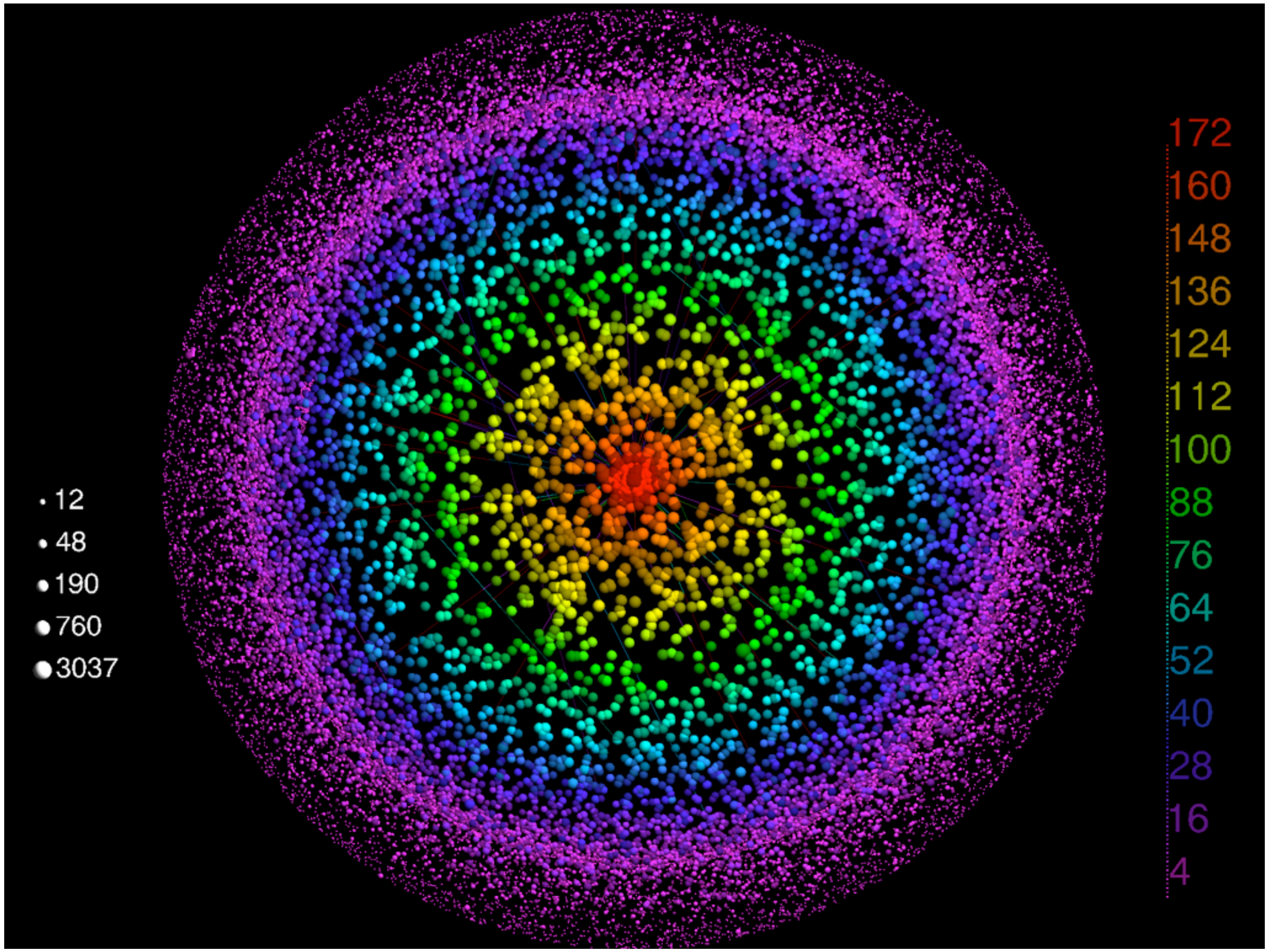
c/soundcloud_4.py collects the full network of SoundCloud friends *(takes a long time!)*

LaNet-vi offers a web service that draws large scale networks of data *(no software required!)*

Plotting as a web service



Plotting as a web service



ismir2009.benfields.net/m/soundcloud_16k.png

Another Application of Graphs

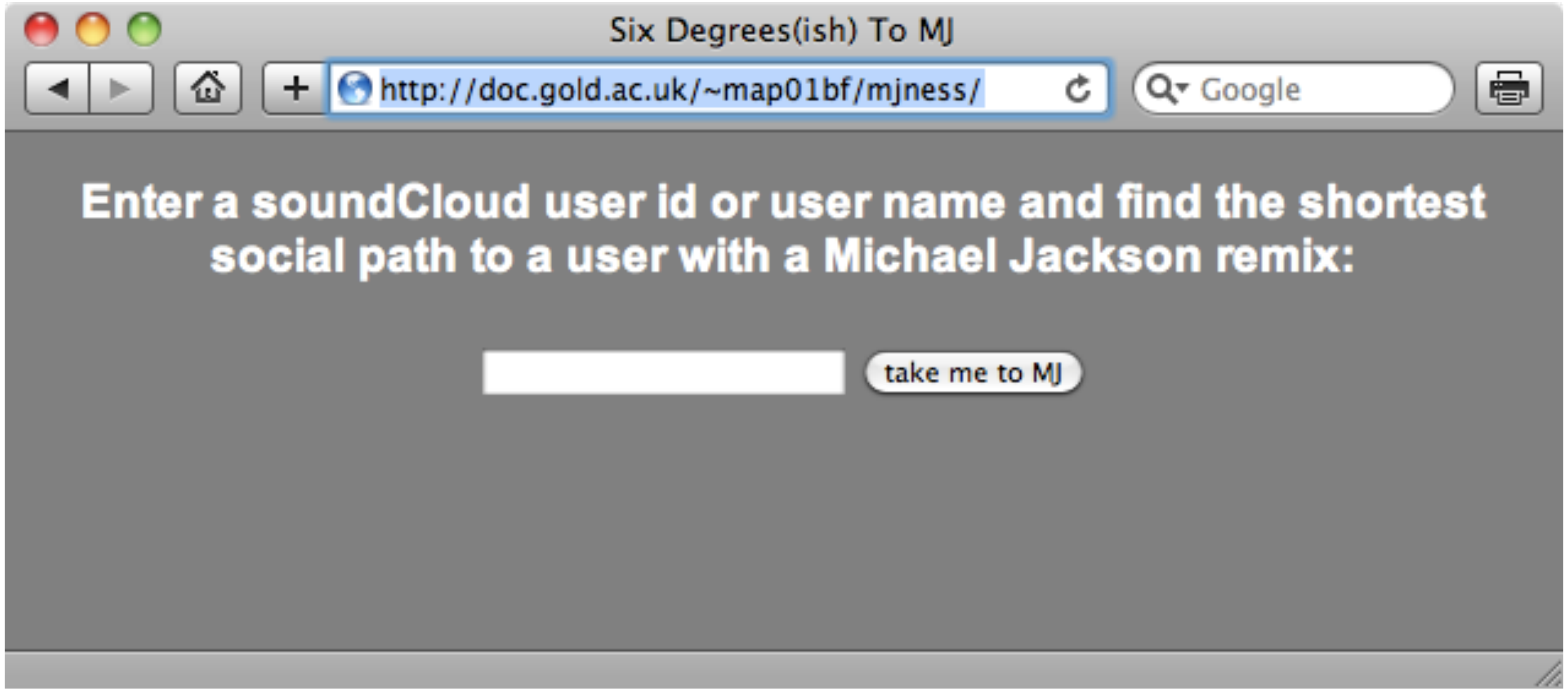
Find the closest Michael Jackson remix to a given user and give an ordered list of artists to get to that user:

http://tr.im/mjness

Another Application of Graphs

Find the closest Michael Jackson remix to a given user and give an ordered list of artists to get to that user:

http://tr.im/mjness


Six Degrees(ish) To MJ

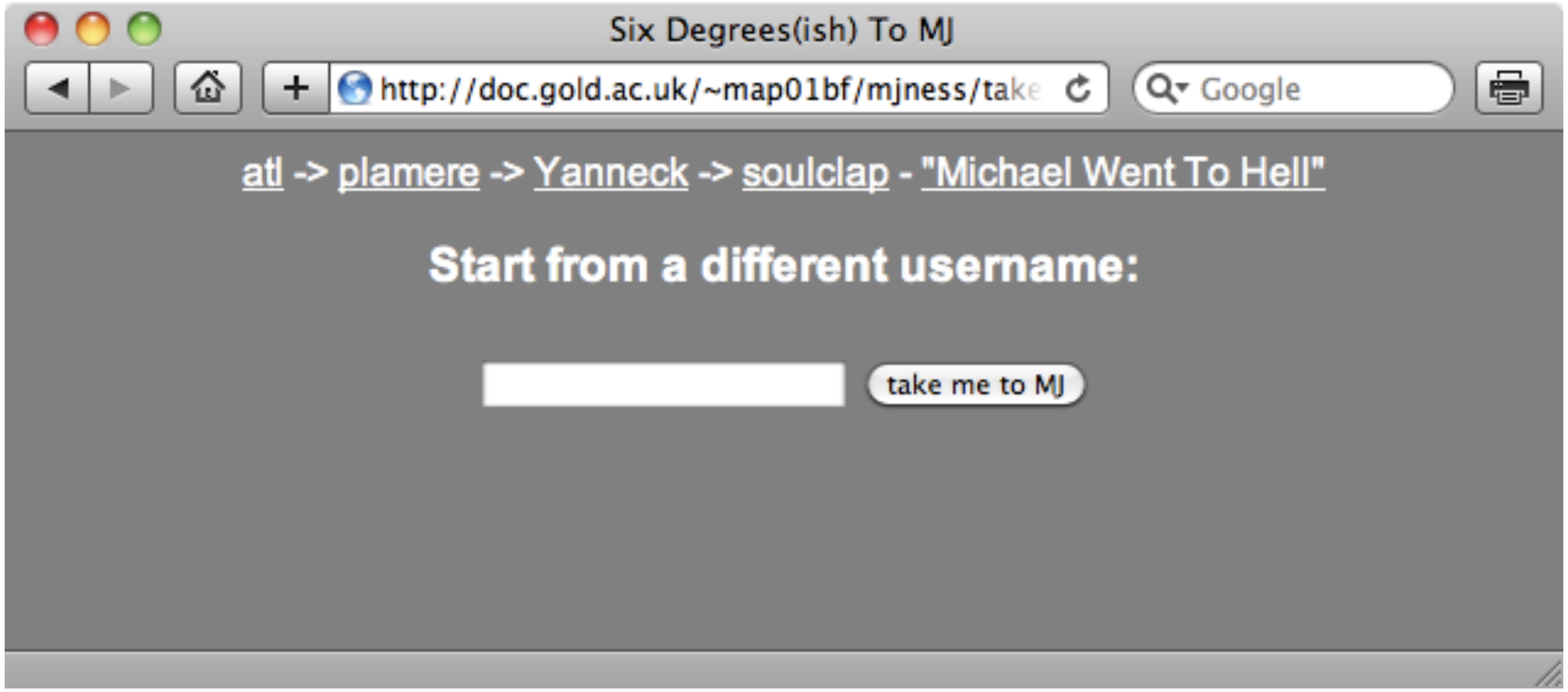
http://doc.gold.ac.uk/~map01bf/mjness/

Enter a soundCloud user id or user name and find the shortest social path to a user with a Michael Jackson remix:

Another Application of Graphs

Find the closest Michael Jackson remix to a given user and give an ordered list of artists to get to that user:

http://tr.im/mjness


Six Degrees(ish) To MJ

http://doc.gold.ac.uk/~map01bf/mjness/take

atl -> plamere -> Yanneck -> soulclap - "Michael Went To Hell"

Start from a different username:

Lessons learnt

Musicians **relate** in online communities with each other (as friends, followers, ...)

Social networks can easily be extracted and plotted, either partially or completely

Plotting can also be outsourced to a **web service**

Researchers can benefit for several applications: **playlist generation**, recommender systems, ...

QUESTIONS?

#6

COLLECTING FEEDBACK

Subjective evaluation

Subjective evaluation

Researchers need feedback in many scenarios

Recommender systems	Automatic composition
Mood-based analysis	...and more

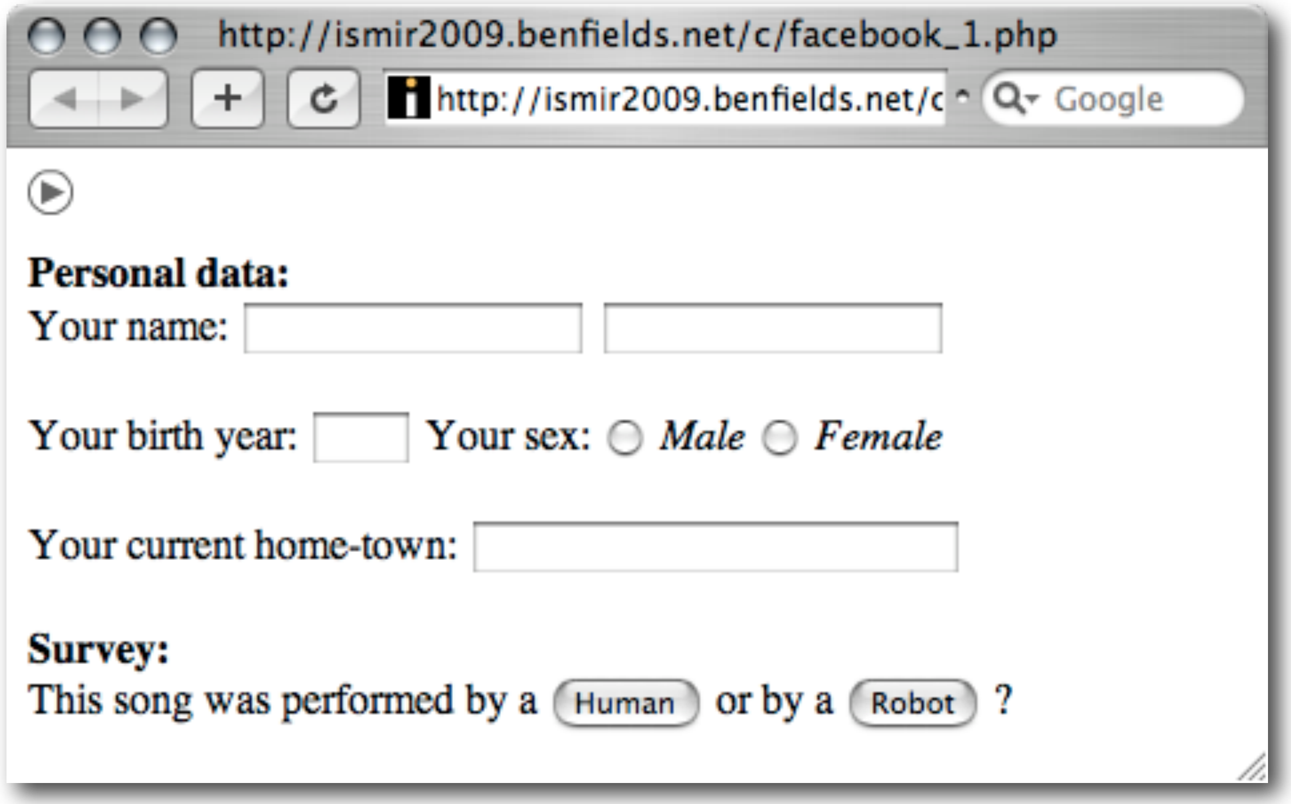
Subjective evaluation

Researchers need feedback in many scenarios

Recommender systems | Automatic composition

Mood-based analysis | ...and more

Setting up a web survey


http://ismir2009.benfields.net/c/facebook_1.php
http://ismir2009.benfields.net/c
Google

Personal data:
Your name:

Your birth year: Your sex: ○ *Male* ○ *Female*

Your current home-town:

Survey:
This song was performed by a Human or by a Robot ?

Subjective evaluation

Researchers need feedback in many scenarios

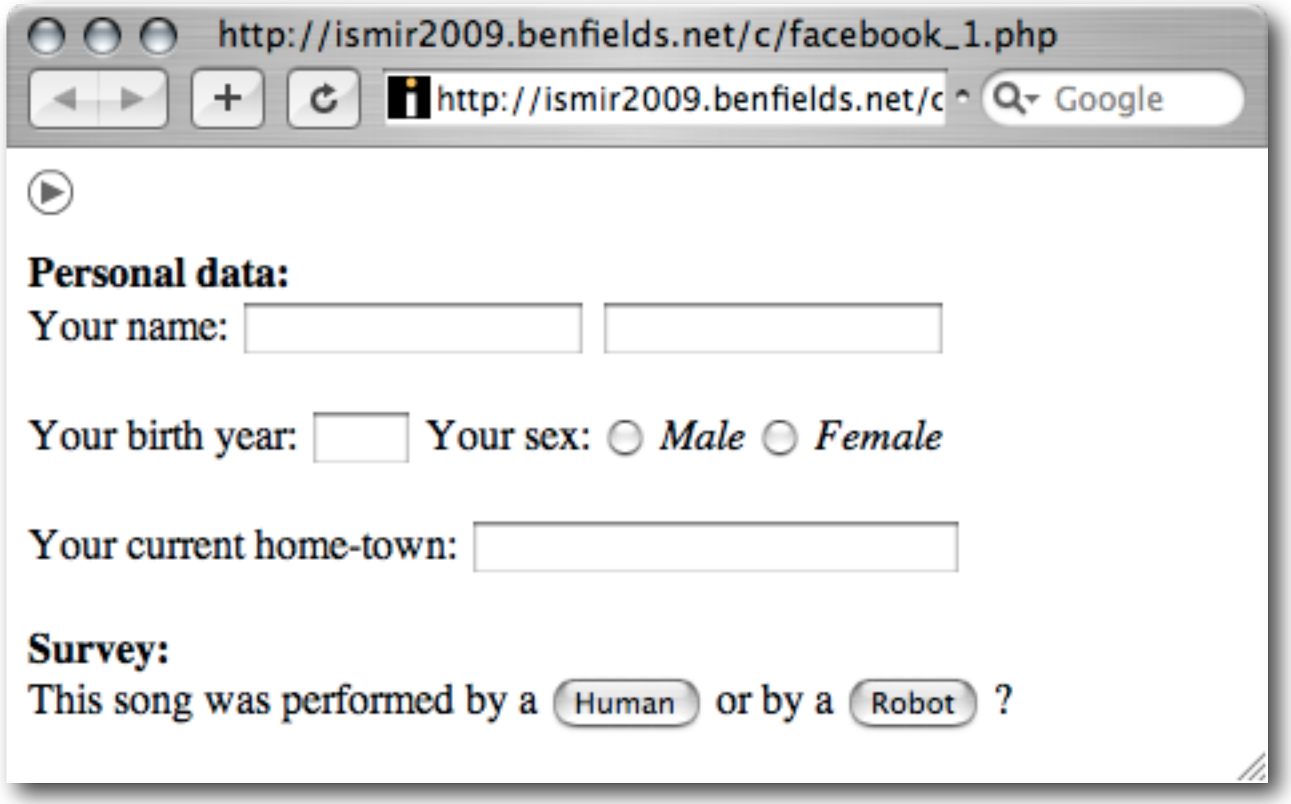
Recommender systems

Automatic composition

Mood-based analysis

...and more

Setting up a web survey **has some drawbacks!**


http://ismir2009.benfields.net/c/facebook_1.php
http://ismir2009.benfields.net/c
Google

Personal data:
Your name:

Your birth year: Your sex: ○ *Male* ○ *Female*

Your current home-town:

Survey:
This song was performed by a Human or by a Robot ?

isolated

hard to share

can vote twice (or never)

requires personal data

Advantages of social networks

Advantages of social networks

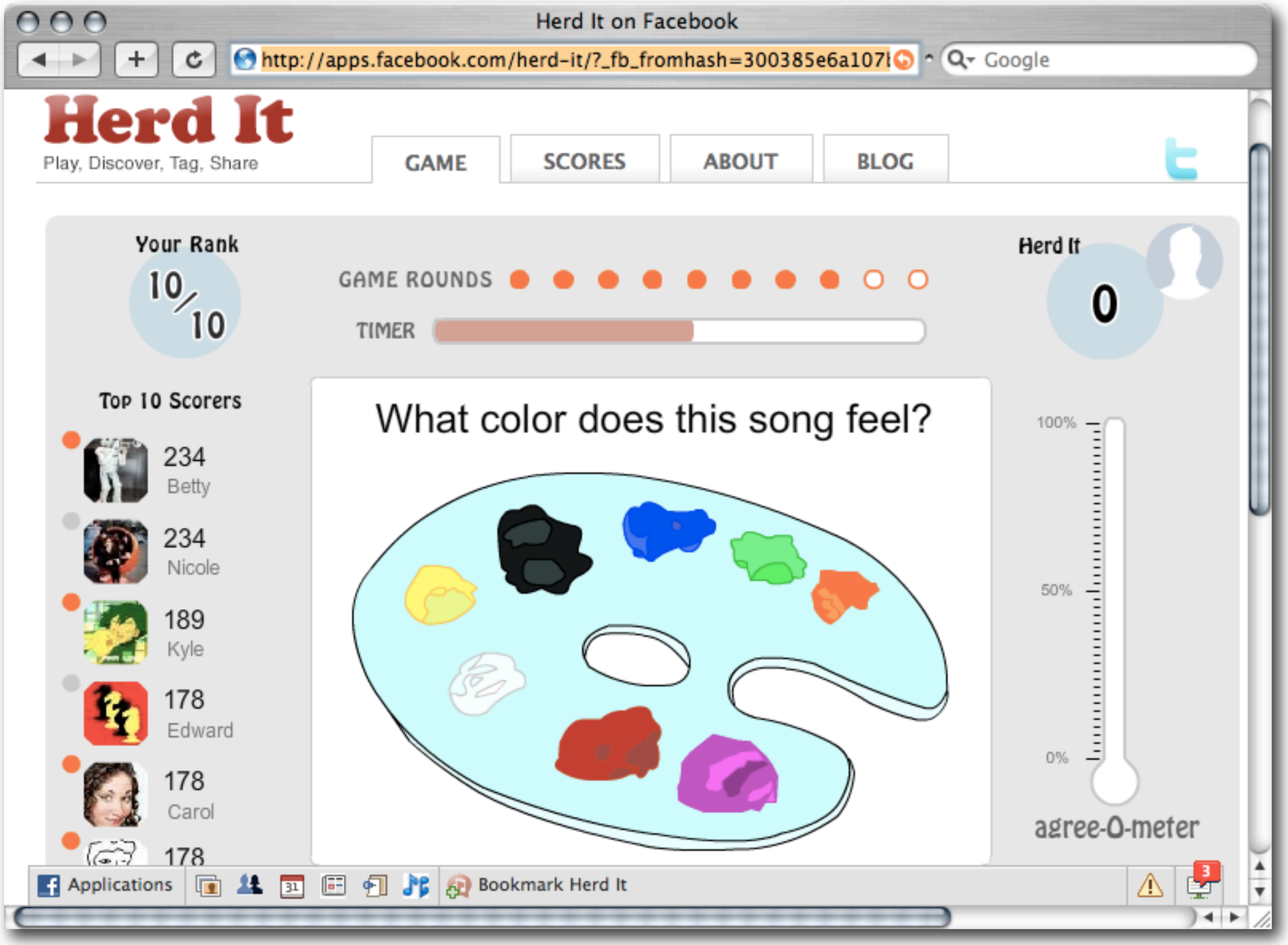
Benefits of publishing the survey on **Facebook**:

Advantages of social networks

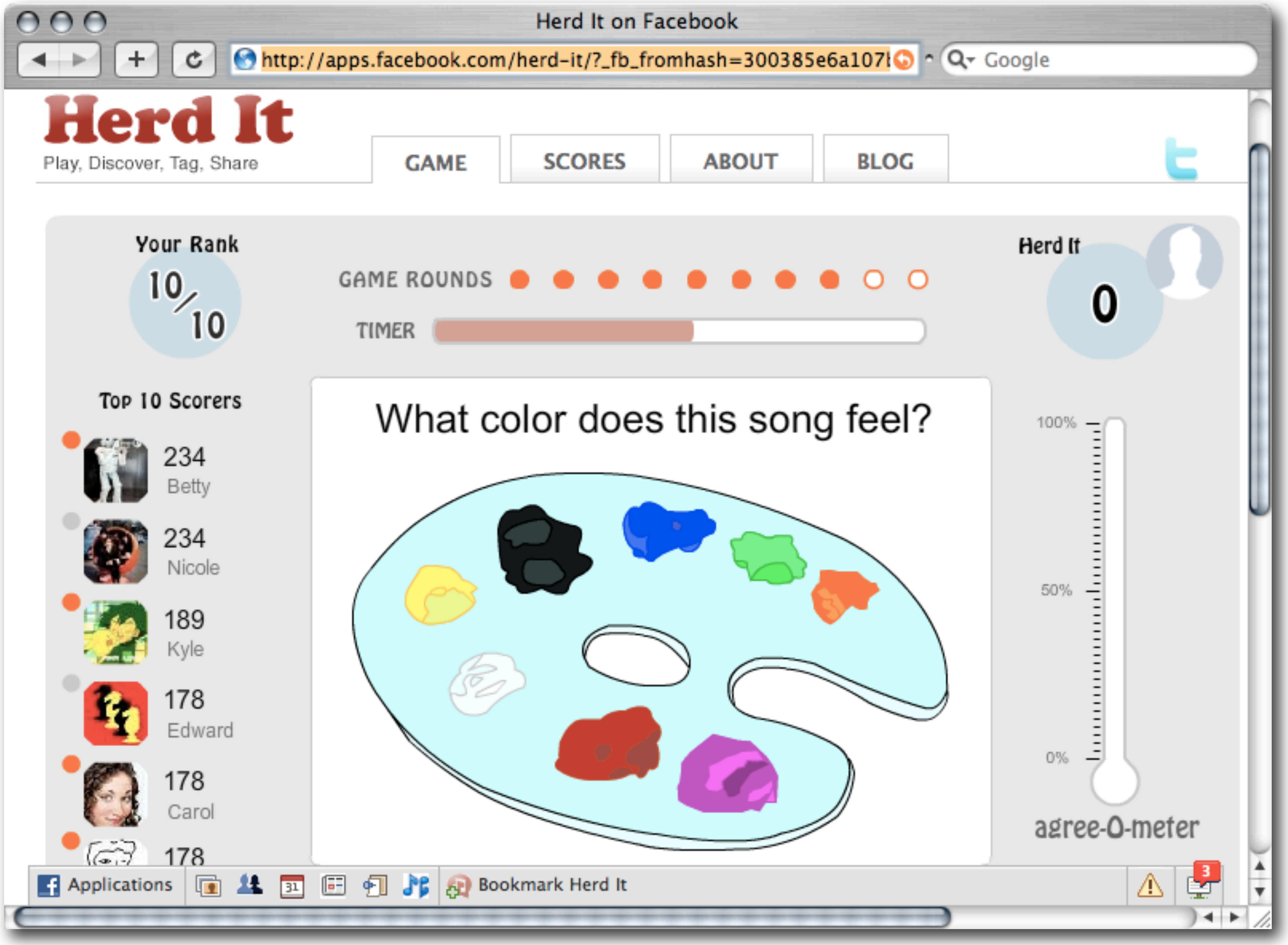
Benefits of publishing the survey on **Facebook**:

1. Collect personal data without filling any form
2. Explore social connections between users
3. Share and publish surveys through networks of friends
4. Potentially reach millions of users in a friendly environment
5. Attract more people with game-styled application

Advantages of social networks



Advantages of social networks



apps.facebook.com/herd-it

Creating a PHP survey

The code is included in the file **c/facebook_1.php**:

Creating a PHP survey

The code is included in the file **c/facebook_1.php:**

```
<?php if(!($vote = @$_GET['vote'])) { ?>

<object type="application/x-shockwave-flash" height="30"
data="mp3player.swf?autoplay=true&amp;song_url=http://
ismir2009.benfields.net/m/saxex.mp3"></object>

<form>
<b>Personal data:</b><br /> Your name:
  <input type="text" name="first_name" size="17">
  <input type="text" name="last_name" size="17"> Your birth year:
  <input type="text" name="birth_year" size="4"> Your sex:
  <input type="radio" name="sex" value="male"> Male
  <input type="radio" name="sex" value="female"> Female
  Your current home-town: <input type="text" name="city" size=25>
<b>Survey:</b><br />
  This song was performed by a
  <input type="submit" value='Human' name="vote" /> or by a
  <input type="submit" value='Robot' name="vote" />?</form>
```

Creating a PHP survey

The code is included in the file **c/facebook_1.php**:

Creating a PHP survey

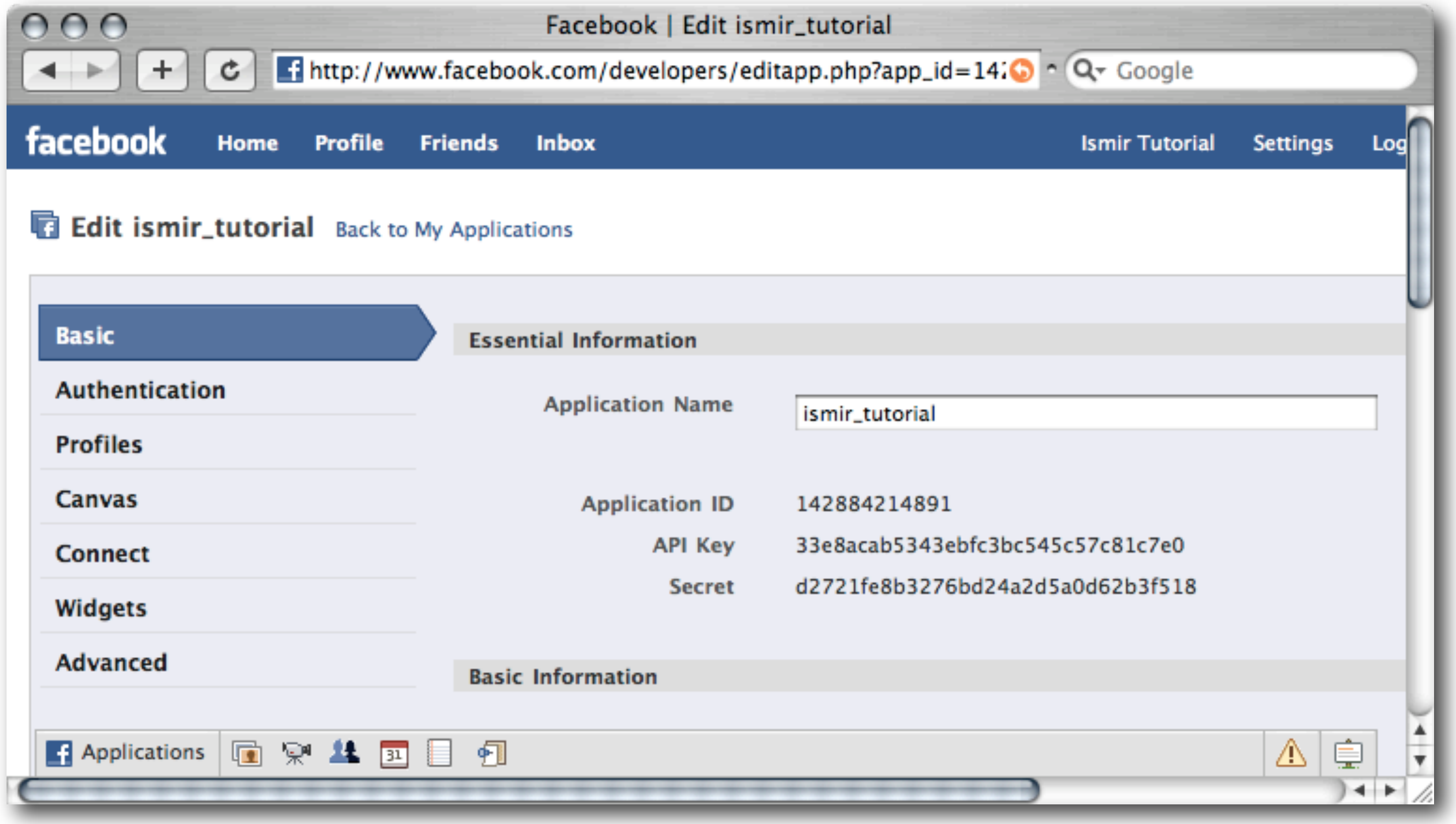
The code is included in the file **c/facebook_1.php:**

```
<?php } else {
 $info = $_GET;

 $data_file = "/tmp/fb_survey.txt";
 $text  = "|". date("Y.m.d H:i:s");
 $text .= "|". $vote;
 $text .= "|". $info['first_name'] ." ". $info['last_name'];
 $text .= "|". $info['sex'];
 $text .= "|". $info['birth_year'];
 $text .= "|". $info['city'];
 $file = fopen($data_file, "a");
 fwrite($file, $text ."\n"); fclose($file);
 $file = file_get_contents($data_file, 'r');
 $human = substr_count($file, '|Human|');
 $robot = substr_count($file, '|Robot|');
 echo "Votes so far: Human ". $human ." - Robot ". $robot;
 echo "<pre>". $file ."</pre>";
} ?>
```

Creating a Facebook application

1. Create a Facebook account
2. Navigate to facebook.com/developers/apps.php
3. Add the Developer application to the Facebook profile
4. Set up a new application


Facebook | Edit ismir_tutorial

http://www.facebook.com/developers/editapp.php?app_id=14

Edit ismir_tutorial Back to My Applications

Essential Information

Application Name	ismir_tutorial
Application ID	142884214891
API Key	33e8acab5343ebfc3bc545c57c81c7e0
Secret	d2721fe8b3276bd24a2d5a0d62b3f518

Basic Information

Setting up the environment

Setting up the environment

5. Copy and paste the application keys to **c/facebook_key.php**:

```
 <?php
$app_id     = "142884214891";
$api_key    = "33e8acab5343ebfc3bc545c57c81c7e0";
$secret_key = "d2721fe8b3276bd24a2d5a0d62b3f518";
?>
```

6. Download and unzip the Facebook PHP client library from facebook.com/developers/apps.php to the **c/** folder

7. Add the following lines at the top of **c/facebook_1.php**:

```
 <?php
require_once('facebook_key.php');
require_once("facebook-platform/php/facebook.php");
$facebook = new Facebook($api_key, $secret);
$user = $facebook->require_login(); ?>
```

Creating a Facebook survey in PHP

8. Modify the rest of **c/facebook_1.php** as follows:

```
<?php if(!($vote = @$_GET['vote'])) { ?>

<object type="application/x-shockwave-flash" height="30"
data="mp3player.swf?autoplay=true&amp;song_url=http://
ismir2009.benfields.net/m/saxex.mp3"></object>

<form>
<b>Personal data:</b><br /> Your name:
  <input type="text" name="first_name" size="17">
  <input type="text" name="last_name" size="17"> Your birth year:
  <input type="text" name="birth_year" size="4"> Your sex:
  <input type="radio" name="sex" value="male"> Male
  <input type="radio" name="sex" value="female"> Female
  Your current home-town: <input type="text" name="city" size=25>
<b>Survey:</b><br />
  This song was performed by a
  <input type="submit" value='Human' name="vote" /> or by a
  <input type="submit" value='Robot' name="vote" />?</form>
```

Creating a Facebook survey in PHP

8. Modify the rest of **c/facebook_1.php** as follows:

```
<?php if(!($vote = @$_GET['vote'])) { ?>

<fb:mp3 src="http://ismir2009.benfields.net/m/saxex.mp3"
title="Autumn Leaves" album="Autumn Leaves" artist="Human or
Robot?" />


<form>
<b>Personal data:</b><br /> Your name:
  <input type="text" name="first_name" size="17">
  <input type="text" name="last_name" size="17"> Your birth year:
  <input type="text" name="birth_year" size="4"> Your sex:
  <input type="radio" name="sex" value="male"> Male
  <input type="radio" name="sex" value="female"> Female
  Your current home-town: <input type="text" name="city" size=25>
<b>Survey:</b><br />
  This song was performed by a
  <input type="submit" value='Human' name="vote" /> or by a
  <input type="submit" value='Robot' name="vote" />?</form>
```

Creating a Facebook survey in PHP

8. Modify the rest of **c/facebook_1.php** as follows:

```
<?php if(!($vote = @$_GET['vote'])) { ?>

<fb:mp3 src="http://ismir2009.benfields.net/m/saxex.mp3"
title="Autumn Leaves" album="Autumn Leaves" artist="Human or
Robot?" />

<form>
<b>Personal data:</b><br /> Your name:
  <input type="text" name="first_name" size="17">
  <input type="text" name="last_name" size="17"> Your birth year:
  <input type="text" name="birth_year" size="4"> Your sex:
  <input type="radio" name="sex" value="male"> Male
  <input type="radio" name="sex" value="female"> Female
  Your current home-town: <input type="text" name="city" size=25>
<b>Survey:</b><br />
  This song was performed by a
  <input type="submit" value='Human' name="vote" /> or by a
  <input type="submit" value='Robot' name="vote" />?</form>
```

Creating a Facebook survey in PHP

9. Modify the rest of **c/facebook_1.php** as follows:

```
<?php } else {
 $info = $_GET;










 $data_file = "/tmp/fb_survey.txt";
 $text  = "|". date("Y.m.d H:i:s");
 $text .= "|". $vote;
 [...]
 echo "Votes so far: Human ". $human ." - Robot ". $robot;
 echo "<pre>". $file ."</pre>";
} ?>
```

Creating a Facebook survey in PHP

9. Modify the rest of **c/facebook_1.php** as follows:

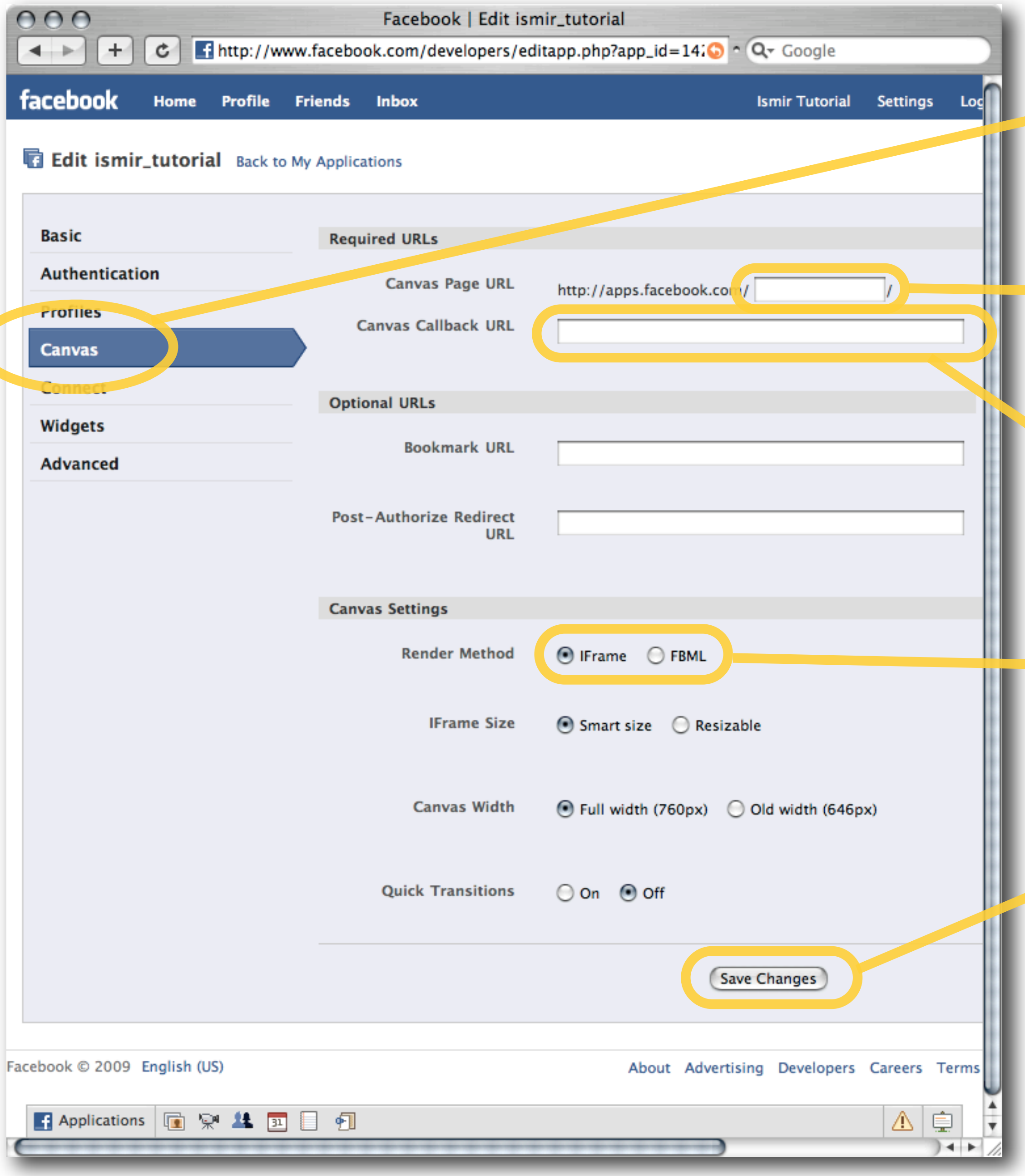
```
<?php } else {
 $info = reset($facebook->api_client->users_getStandardInfo
($user, array('first_name', 'last_name', 'sex', 'birthday',
'current_location')));
 $info['birth_year'] = isset($info['birthday']) ?
   date("Y", strptime($info['birthday'], "%m %d, %Y")) : '';
 $info['city'] = isset($info['current_location']) ?
   $info['current_location']['city'] : '';

 $data_file = "/tmp/fb_survey.txt";
 $text  = "|". date("Y.m.d H:i:s");
 $text .= "|". $vote;
 [...]
 echo "Votes so far: Human ". $human ." - Robot ". $robot;
 echo "<pre>". $file ."</pre>";
} ?>
```

Deploying to Facebook

Deploying to Facebook

Update the code from the file **c/facebook_2.php** to a web server and update Facebook settings:



Facebook | Edit ismir_tutorial

http://www.facebook.com/developers/editapp.php?app_id=14

facebook Home Profile Friends Inbox Ismir Tutorial Settings Log

Edit ismir_tutorial Back to My Applications

Basic
Authentication
Profiles
Canvas
Connect
Widgets
Advanced

Required URLs

Canvas Page URL: http://apps.facebook.com/

Canvas Callback URL:

Optional URLs

Bookmark URL:

Post-Authorize Redirect URL:

Canvas Settings

Render Method: [✓ IFrame] FBML

IFrame Size: [✓ Smart size] Resizable

Canvas Width: [✓ Full width (760px)] Old width (646px)

Quick Transitions: On [✓ Off]

Save Changes

Facebook © 2009 English (US) About Advertising Developers Careers Terms

Applications

Click on the **canvas** tab

Fill the desired web **address**

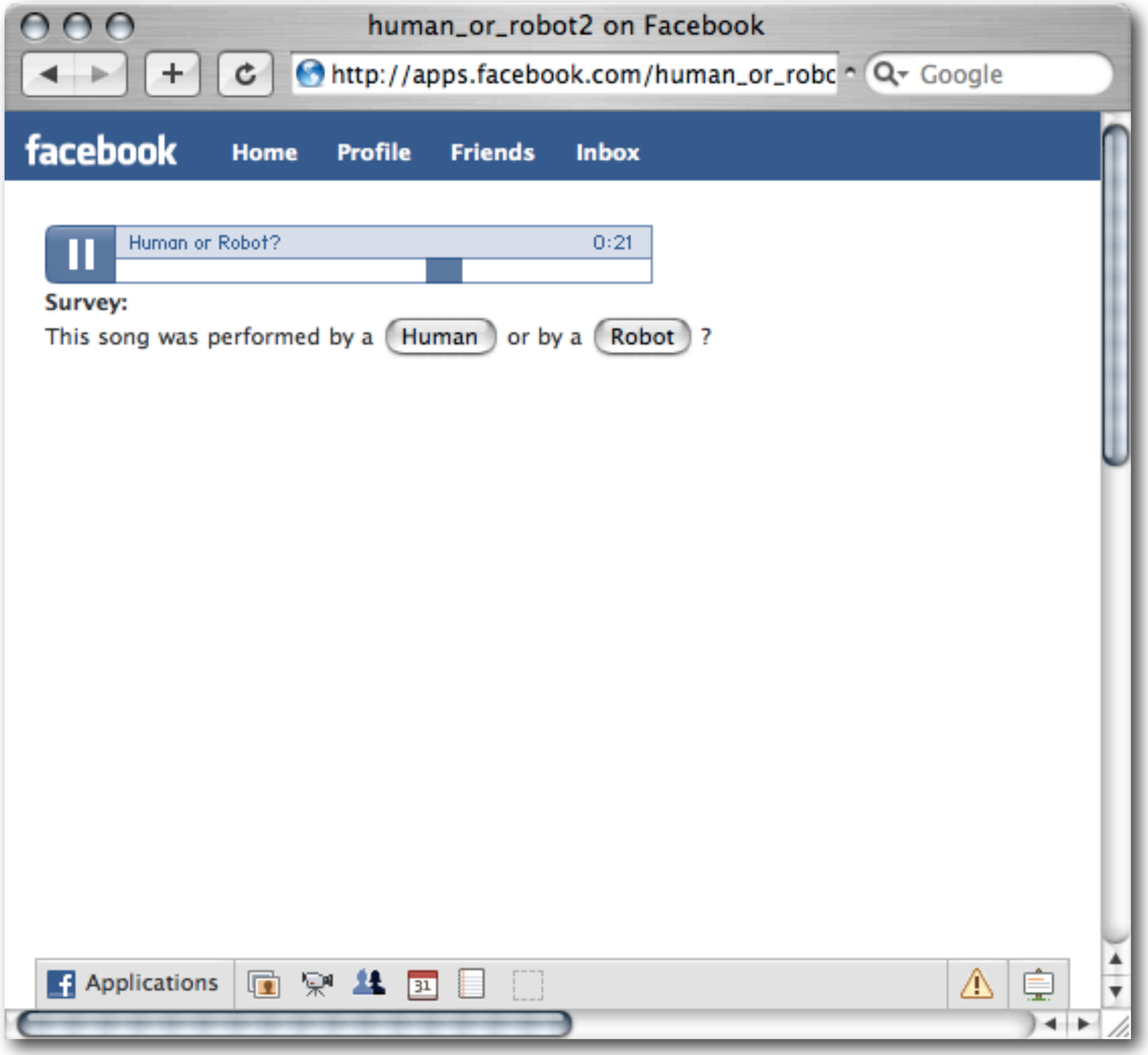
Specify the **location** of the code

Select FBML as **render** method

Save changes

Visit the application page

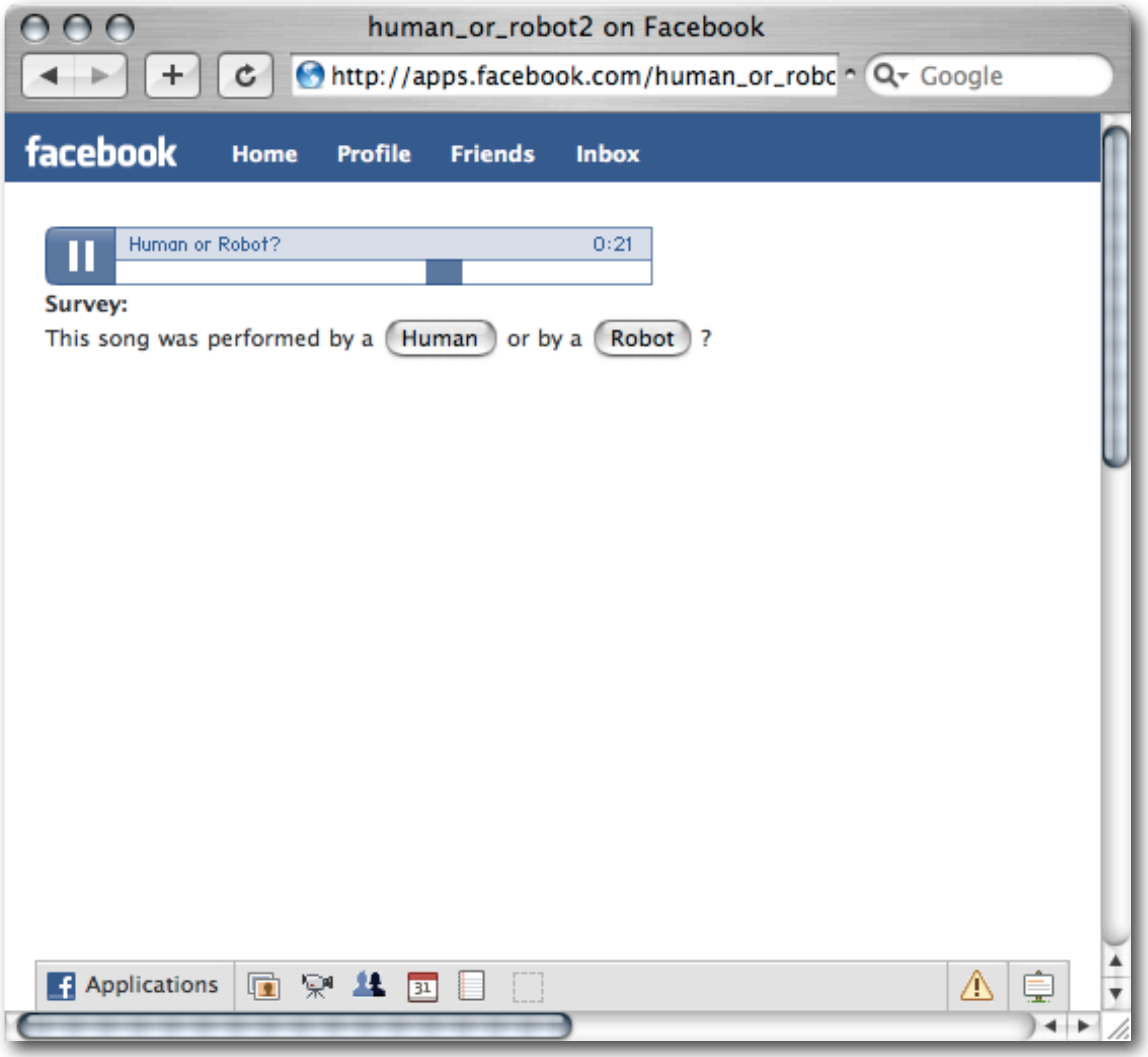
Deploying to Facebook



Survey:

This song was performed by a Human or by a Robot ?

Deploying to Facebook


human_or_robot2 on Facebook

http://apps.facebook.com/human_or_robc

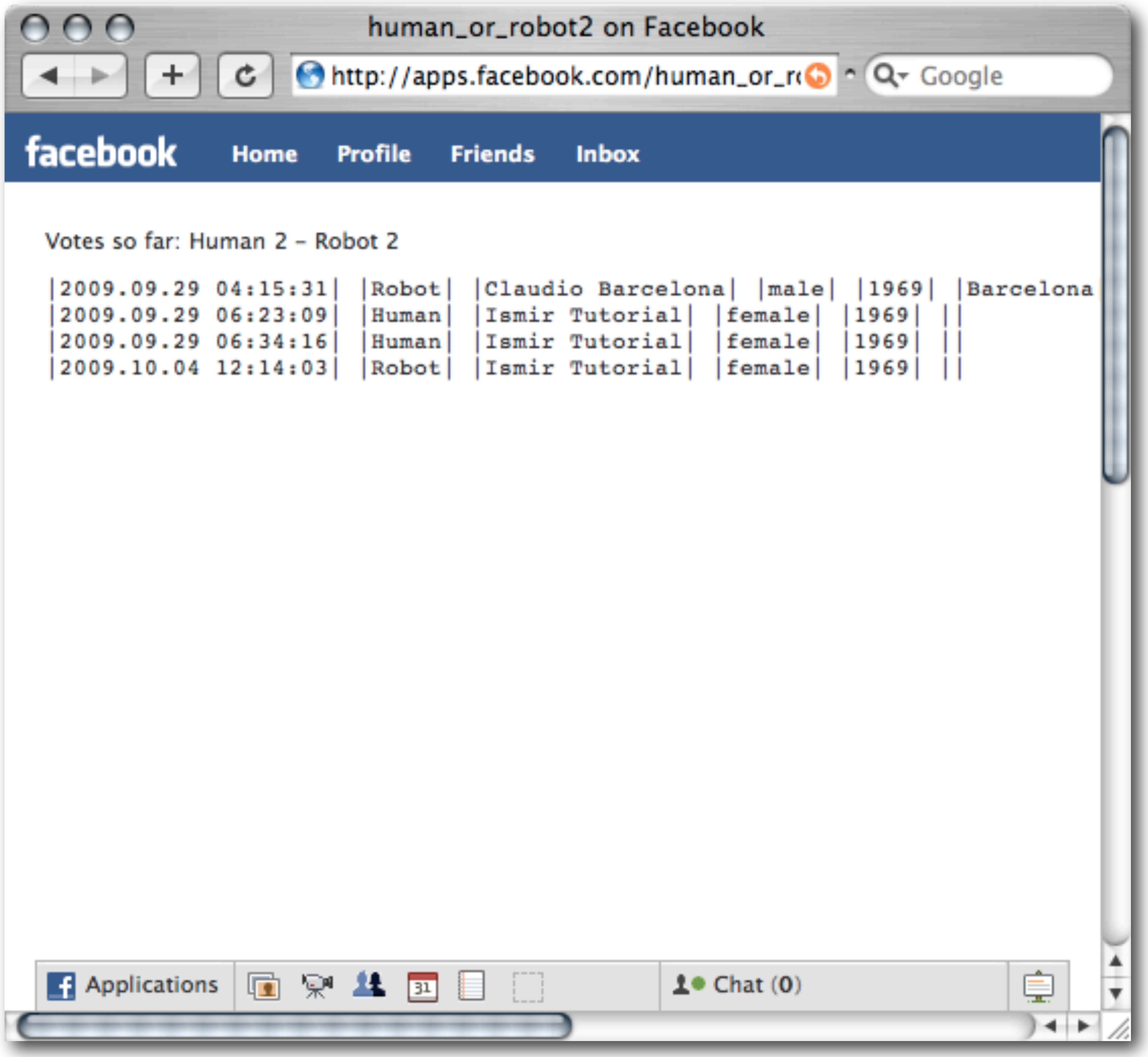
Human or Robot? 0:21

Survey:

This song was performed by a Human or by a Robot ?

apps.facebook.com/*CANVAS_PAGE_URL*

Deploying to Facebook


human_or_robot2 on Facebook
http://apps.facebook.com/human_or_r
Google
facebook
Home
Profile
Friends
Inbox
Applications
Chat (0)

Votes so far: Human 2 - Robot 2

```
|2009.09.29 04:15:31| |Robot| |Claudio Barcelona| |male| |1969| |Barcelona
|2009.09.29 06:23:09| |Human| |Ismir Tutorial| |female| |1969| ||
|2009.09.29 06:34:16| |Human| |Ismir Tutorial| |female| |1969| ||
|2009.10.04 12:14:03| |Robot| |Ismir Tutorial| |female| |1969| ||
```

apps.facebook.com/*CANVAS_PAGE_URL*

Lessons learnt

Developing **Facebook** apps is fast and easy

Users can benefit from the **social** environment

More social **features** can be exploited

QUESTIONS?

CONCLUSIONS

Mining the web for musical data is **easy and fast**

Every researcher can **benefit** from this approach

Web APIs can be **combined** for specific goals

Online **social networks** are practical to collect human experiences and evaluate hypotheses

The web also offers **tools** for analysis and graphs

Mining the web for musical data is **easy and fast**

Every researcher can **benefit** from this approach

Web APIs can be **combined** for specific goals

Online **social networks** are practical to collect human experiences and evaluate hypotheses

The web also offers **tools** for analysis and graphs

Give it a try!

Other interesting web services

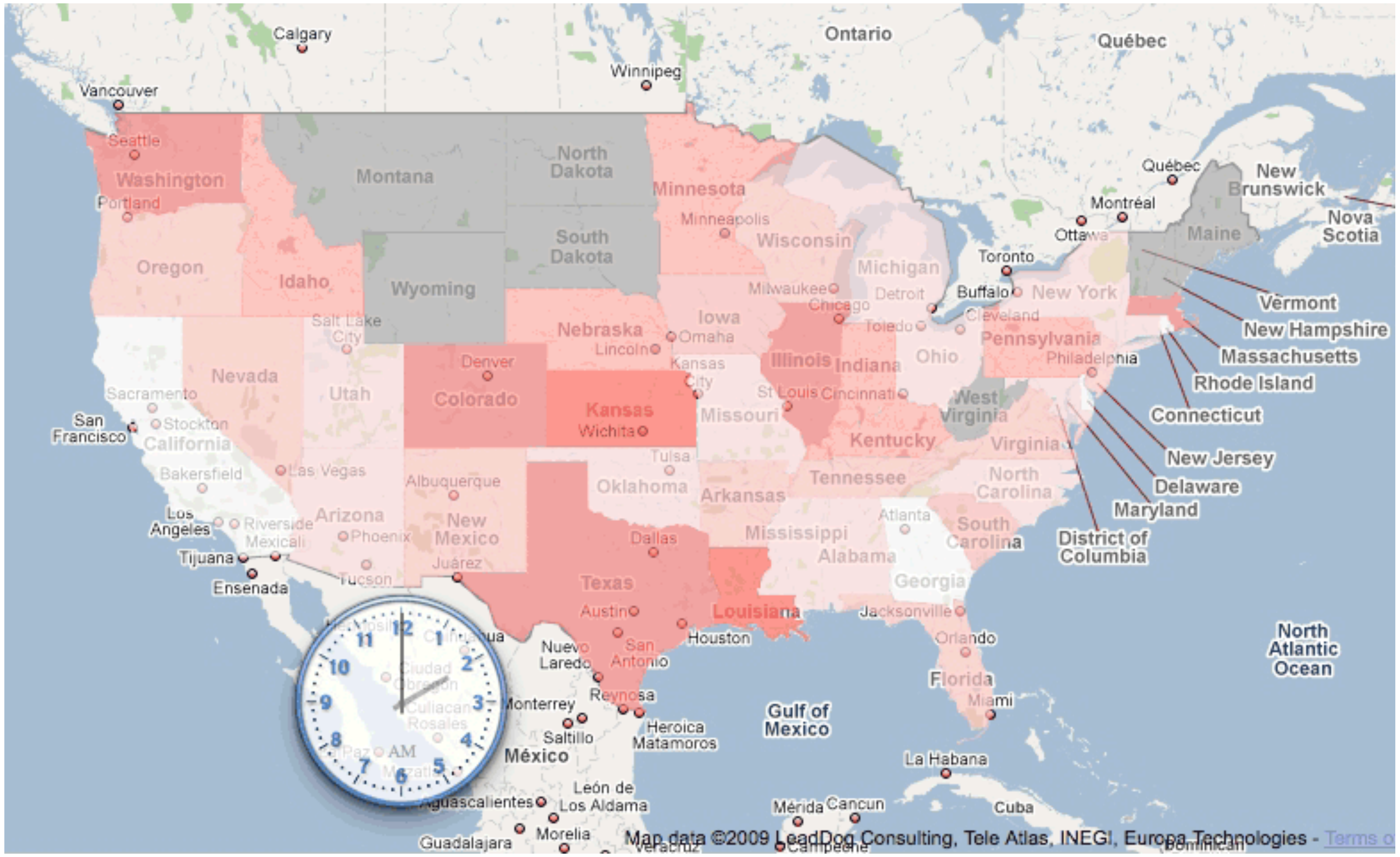
1. Music reviews (the guardian, NY Times)
2. Music TV shows (BBC)
3. Concerts (Songkick)
4. Sales data (7digital, People's music store)
5. Sheet music (Musopen)
6. Distribute composition (Noteflight)
7. Playlists (Art of The Mix, playlist.com, Spotify, iTunes Store)
8. Radio programmes (Yes.com, Shoutcast)

A more advanced mash-up

ismir2009.benfields.net/gmapradio

A more advanced mash-up

ismir2009.benfields.net/gmapradio


Map data ©2009 LeadDog Consulting, Tele Atlas, INEGI, Europa Technologies - Terms o

YES

last.fm

Google

QUESTIONS?