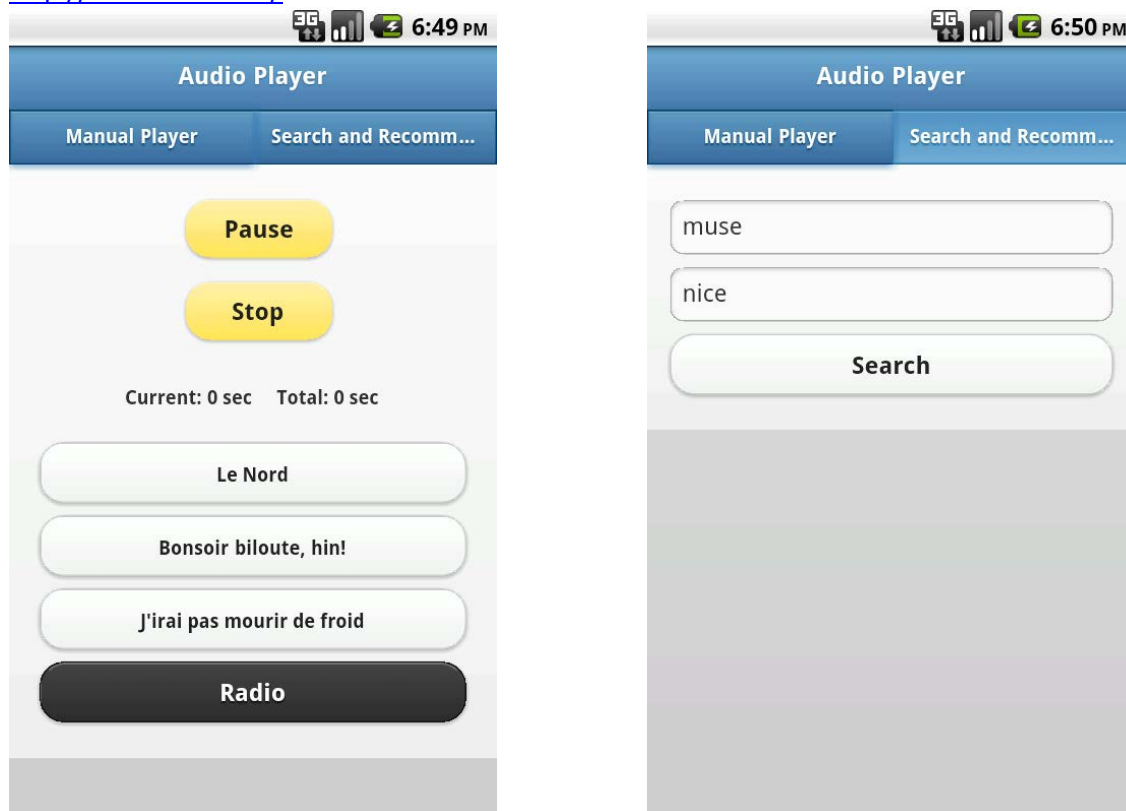


1 Music.fm

In this lab session, we would like to build a music/radio player capable of playing, stopping, pausing the music and/or radio station as well as showing the time elapsed since the music/radio is played. Furthermore, the application will search for nearby concerts using a REST web service provided by setlist.fm. The setlist.fm API has been designed to give you easy access to setlist data in order to enrich your application with popular artists, concerts, and other relevant statistics. See <http://api.setlist.fm/docs/index.html> for API documentation. Check the website as well <http://www.setlist.fm/>



This lab session is based on PhoneGap for Android using Cordova CLI. You may use part of the code developed from the previous lab session (for instance IP location to find the nearby concerts and events).

For the UI, this application makes use of jQueryMobile (Touch-Optimized Web Framework) version 1.2, see <http://jquerymobile.com/>, for HTML5-based user interface system. Feel free to checkout the **overview** section of jquery in <http://jquerymobile.com/demos/1.2.0/>. Ionic is a new framework used to design the UI for the phonegap application (<http://ionicframework.com/>). In most of the cases, ionic makes use of Angularjs framework to improve the responsive of the app (<https://angularjs.org/>) .

The media player application uses the remote css and js files (see the header of index.html: jquery.mobile-1.2.0.min.css, jquery-1.8.2.min.js, and jquery.mobile-1.2.0.min.js). You can also download them locally and compare the performance of the application.

1.1 Setup the environment

- Add the following in your ~/.bashrc+.
 - export JAVA_HOME=/usr/jdk/latest
 - export PATH=\$PATH:/packages/mobserv/android-sdk/tools:/packages/mobserv/android-sdk/platform-tools/
- use local storage for Android emulator
 - Go to your homes : cd ~
 - Remove .android from your homes, i.e. rm -rf ~/.android
 - Create a local directory, mkdir /home/LOCAL_DATA/android.\$USERNAME
 - Create a symbolic link to this new directory: ln -s /home/LOCAL_DATA/android.\$USERNAME .android
- create and run the emulator and create an AVD for API level 22:
 - android avd
 - device: nexus 4
 - RAM: 1024
 - internal storage 100 MiB
 - SD card: 32
 - Then, start the emulator.

1.2 Create a project

- mkdir phonegap
- cd phonegap
- cordova create music.fm fr.eurecom.musicfm MusicFM
- cordova platform add android (to test run: "phonegap cordova platform ls")
 - check the android target in the output, it should be api level 22.
- Overwrite the www directory with /packages/mobserv/phonegap/musicfm/www/
- cordova emulate android

Make sure that the required android permissions are granted as well as the required phonegap features are added to config.xml and plugins are installed. You need

- Device
- Console
- Media capture
- File api

Filter the logs using adb logcat CordovaLog:D *:S

- This will show you only CordovaLog items (the "D" indicates Debug level or above) and the "*:S" indicates to filter out all others.

1.3 PhoneGap Media object

The Media object provides the ability to record and play back audio files on a device.

```
var media = new Media(src, mediaSuccess, [mediaError], [mediaStatus]);
```

Parameters

- **src**: A URI containing the audio content. (*DOMString*)
- **mediaSuccess**: (Optional) The callback that executes after a `Media` object has completed the current play, record, or stop action. (*Function*)
- **mediaError**: (Optional) The callback that executes if an error occurs. (*Function*)
- **mediaStatus**: (Optional) The callback that executes to indicate status changes. (*Function*)

Constants for media status

- `Media.MEDIA_NONE = 0;`
- `Media.MEDIA_STARTING = 1;`
- `Media.MEDIA_RUNNING = 2;`
- `Media.MEDIA_PAUSED = 3;`
- `Media.MEDIA_STOPPED = 4;`

Methods

- **media.getCurrentPosition**: Returns the current position within an audio file.
- **media.getDuration**: Returns the duration of an audio file.
- **media.play**: Start or resume playing audio file.
- **media.pause**: Pause playing audio file.
- **media.release**: Releases the underlying OS'es audio resources.
- **media.seekTo**: Moves the position within the audio file.
- **media.startRecord**: Start recording audio file.
- **media.stopRecord**: Stop recording audio file.
- **media.stop**: Stop playing audio file.

See the examples in <http://plugins.cordova.io/#/package/org.apache.cordova.media> .

In the same page, you can find a full example for each method, that you need to combine to build your music player. If needed, you may also use some Capture methods:

http://docs.phonegap.com/en/edge/cordova_media_capture_capture.md.html#Capture

For the radio stations and music, provide a list of internet radio stations to the user. You could hard-code the URI/URL in your application. you could get inspired by Tunein (<http://tunein.com/>) or simply search for the URL of internet radio stations. Here is some links:

- <http://www.internet-radio.com/>
- http://doc.ubuntu-fr.org/liste_radio_france
- <http://www.thestreamcenter.com/state.asp?state=VA>
- <http://radio-locator.com/>

1.4 Search and recommend concert

This feature is provided by setlist APIs (api.setlist.fm), and in particular you are interested in getting a list of matching setlists. This is obtained through <http://api.setlist.fm/rest/0.1/search/setlist.json>, you need to find the concert for a given artistName and cityName (see table below or <http://api.setlist.fm/docs/rest.0.1.search.setlists.html> for additional help). The response will be a Json object that you need to parse and present to user in a convenient manner.

For example check: <http://api.setlist.fm/rest/0.1/search/setlists.json?artistName=muse&cityName=nice>

1.5 Implement a new feature using setlist API

Imagine a new feature that you can provide using setlist API, and implement it using an extra tab, e.g.

- Combine cityName and year, e.g. paris and 2013
- The venue by category, see <http://www.setlist.fm/venues>
- Top setlist, see <http://www.setlist.fm/setlists>
- ... you decide

1.6 Play music from the sdcard

Add a function to read and play a file from the sdcard. Check out the required permission, feature, and plugin. Make sure that Android emulator has sdcard storage.

1.7 Test

Make sure that the generated ADV includes the audio recording and playback supports.

name	description	type	default
artistMbid	the artist's Musicbrainz Identifier (mbid)	query	
artistTmid	the artist's Ticketmaster Identifier (tmid)	query	
artistName	the artist's name	query	
tour	the tour name	query	
date	the date of the event (format dd-MM-yyyy)	query	
year	the year of the event	query	
lastFm	the event's Last.fm Event ID	query	
lastUpdated	the date and time (UTC) when this setlist was last updated (format yyyyMMddHHmmss) - either edited or reverted. search will return setlists that were updated on or after this date	query	
venueId	the venue id	query	
venueName	the name of the venue	query	
cityId	the city's geoid	query	
cityName	the name of the city	query	
stateCode	the state code	query	
state	the state	query	
countryCode	the country code	query	
p	the number of the result page	query	1
l	desired result language of city and country (experimental feature)	query	en

