



Deezer Applications Hands On

Waves Vienna Music Hack Day

October 1st 2016

Prerequisites

You must have a Deezer account to play music from Deezer, whatever the means. For your convenience, **ask the Deezer team to grant you a full access** during the Hack Day to test fully your hacks. Without this access, you can hear to the first 30 seconds of every track.

> Create a Deezer account : <https://www.deezer.com/>

Deezer music is encrypted when streamed. You **have to** use one of our SDKs to play its content.

Create a developer profile

Go to <http://developers.deezer.com/> to create a Developer's profile. It will be linked to your Deezer account.

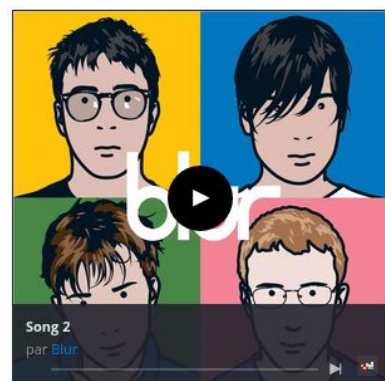
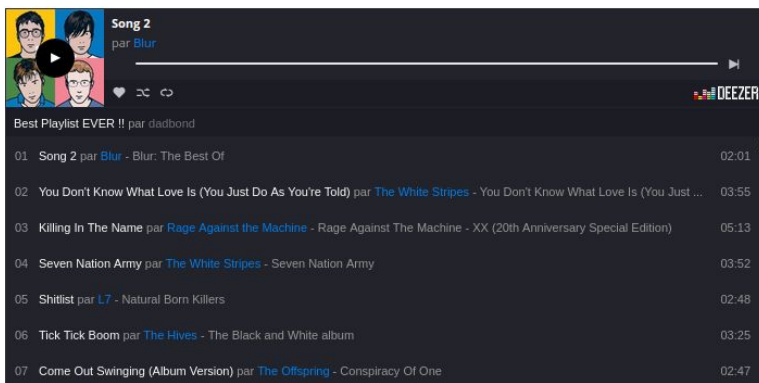
Deezer for Developers is the platform where you create and set up your applications.

Get Deezer Data: the API

To access to the metadata of our huge catalog, the [Deezer API](#) is your best friend. You can search for artists, albums, songs, playlists, podcasts, radios... and get all their metadata. To make your work even easier, use the [API explorer](#) to test your requests beforehand.

The API needs an [OAuth authentication](#) only if you want personal datas, and you must specify which [permissions](#) will be asked to the user.

Play Deezer Music with the Widget Player



The [Deezer Widget Player](#) is the easiest way to play Deezer music. You can set it up with an initial track list, and control it later by the Deezer Javascript SDK. The widget will ask you to connect to Deezer if you are not connected (even for a free account)

Create an application

You need to create an application to use **any** SDK. The SDK will need your application ID to initialize. To do so, go to <http://developers.deezer.com/>, *My Apps* and *Create a new Application*.

The *Domain* is the Internet domain where your application will be running. *localhost* (127.0.0.1) is acceptable.

Using the JS SDK

You need the `and` SDK to control the player for basic controls (play, pause, ...) or to change the current track list for example.

To initialize your player, follow carefully [the given steps here](#). After initialization, you'll be happy to control the player with the `DZ.player` object with [its many controls](#), like `DZ.player.setVolume(70)`. You can try it in your explorer console.

A **channel file** is necessary for initialization. The required `channelUrl` parameter must contain a link to a page only containing a script element pointing to the JavaScript SDK:
`<script src="http://cdn-files.deezer.com/js/min/dz.js"></script>`

You can also use the [DZ.api](#) object to get metadata from the SDK itself:

```
// Get data from user with ID 5
DZ.api('/user/5', function(response) {
    console.log("Name of user id 5", response.name);
});
```

Creating a Web App

Deezer embed your application in its main interface, as many [web apps already existing](#). A web app is like a regular application, but **must be technically validated by the Deezer team** (ask us for today! We never say no). It then has a set of different parameters in the developer's panel, such as the location of the page you want to embed (it can be localhost).

This [demo app](#) gives you precious details about how to create your first app. The JS SDK is your only option here, and to init your player is even simpler than a standalone page, as you don't need to specify a channel (the SDK is already there) or an app ID. The `DZ.player` JS object is available after page loading, and controls the real Deezer player, at the bottom left (web).

Your web app is available at http://www.deezer.com/app/{app_id}

Using your own JS player

You don't have to use the widget player or the Deezer player (web app) to listen to Deezer music. Indeed, you can set up a full personalized player, where your elements will provoke JS to call the then *invisible* Deezer player and react accordingly. You can also implement your own progress bar from callbacks given by the player. [This page explains how to do your own player](#). The [player events](#) will be helpful.

Using iOS and Android SDKs

You will find step by step tutorials about the iOS and Android SDKs on their respective pages:

- > <http://developers.deezer.com/sdk/ios>
- > <http://developers.deezer.com/sdk/android>

Using the native SDK

The recent release of a native SDK is a blessing for all hackers who wish to embed their own creativity in a specific device. The following table gives you compatibilities. The Linux ARM compatibility means you can make it run on a Raspberry Pi.

	i386	X86_64	ARM
Mac OS X ⁽¹⁾		X	
Windows ⁽²⁾	X	X	
Linux ⁽³⁾	X	X	X

⁽¹⁾ Mac OS X 10.8 or upper. ⁽²⁾ Windows7 or upper. ⁽³⁾ Validated with Fedora23 and Raspbian Jessie.

Don't forget your device must have a minimal processor power to be able to decrypt and play the Deezer stream live.

[This page explains everything you need](#) to bootstrap your native Deezer player.

Tell us if you plan to integrate Deezer in your project, and

ENJOY

Anis Khlif: akhlif@deezer.com, Mickaël Arcos: mar@deezer.com - Deezer R&D