



Melomics API Handbook

versión 1.1

April 13, 2013

Melomics Media Inc.

Melomics API Handbook

[Introduction](#)

[Access To API and Authentication](#)

[Understanding API methods responses](#)

[Searching for songs](#)

[Searching for users](#)

[Searching for user's public playlists](#)

[OAuth authentication](#)

[API Calls Table](#)

Introduction

This document describes version 1.0 of Melomics' API, which allows you to access its repository of musical contents (available in Melomics' browser). By using the methods providing by the Melomics API you will be able to browse contents from your app. In addition, Melomics API also provides methods for accessing to the public content of Melomics users such as their public song lists. In order to a better understanding about how this API works some code example are shown and explained. The used format is JSON and the base URL for API endpoints is:

```
http://melomics.com/api/...
```

Access To API and Authentication

To access the version 1.0 of Melomics API you do not need any API key due to the calls can be accomplished in an anonymously way.

Understanding API methods responses

The Melomics API returns data in JSON format, which contains information about songs, users or lists of song (*playlists*). Here is an example of response for *songs* labeled with tag "piano". The call will return a list with all the *songs* matching instrument piano.

Example

```
http://melomics.com/api/songs?q=%23instrument:piano
```

Response

```
{
  "songs": [
    {
      "author": {
        "artistName": "Melomics",
        "avatar":
        "https://secure.gravatar.com/avatar/e92c4472c967de848b7958c6c2f53c0d?d=identicon&rg",
        "description": "<p><strong>melomics</strong> is the first computer that has mastered
        human musical language.</p>\n\n<p>With more than a billion songs, its production constitutes the
        biggest collection of music in the world, and the main contribution to <strong>melomics</strong>
        catalogue.</p>\n",
        "email": "",
        "id": "5527d30c-25af-46e5-8707-0428741afb4a",
        "lang": "en",
        "name": "",
        "payment": {
          "email": "melomics.spain@gmail.com"
        },
        "permalink": "melomics",
        "showEmail": false,
        "surname": "",
        "uri": "@melomics",
        "username": "melomics"
      },
      "buy": false,
      "description": "",
      "download": false,
      "duration": 276000,
      "entities": {
        "midi": {
          "buy": false,
          "download": false,
          "price": -1.0
        },
        "mp3": {
          "buy": false,
          "download": false,

```

```
    "price": -1.0
  },
  "pdf": {
    "buy": false,
    "download": false,
    "price": -1.0
  },
  "xml": {
    "buy": false,
    "download": false,
    "price": -1.0
  }
},
"image": "/mexels/ebb39b729b-78a4-4591-bc13-640d72c47b33",
"like": null,
"locked": false,
"songID": "ebb39b729b-78a4-4591-bc13-640d72c47b33",
"numLikes": 0,
"owner": {
  "artistName": "Melomics",
  "avatar":
"https://secure.gravatar.com/avatar/e92c4472c967de848b7958c6c2f53c0d?d=identicon&rg",
  "description": "<p><strong>melomics</strong> is the first computer that has mastered
human musical language.</p>\n\n<p>With more than a billion songs, its production constitutes the
biggest collection of music in the world, and the main contribution to <strong>melomics</strong>
catalogue.</p>\n",
  "email": "",
  "id": "5527d30c-25af-46e5-8707-0428741afb4a",
  "lang": "en",
  "name": "",
  "payment": {
    "email": "melomics.spain@gmail.com"
  },
  "permalink": "melomics",
  "showEmail": false,
  "surname": "",
  "uri": "@melomics",
  "username": "melomics"
},
"permalink": "8",
"tags": [
  "effect:gliss",
  "effect:trill",
  "tempo:66bpm",
  "instrument:piano",
  "effect:grace"
],
```

```

    "title": "8",
    "totalPrice": -1.0,
    "type": "song",
    "uri": "@melomics/8",
    "visibility": "public"
  },
}
{.....}
{.....}
{.....}
(other listed songs)
],

"more": false
}

```

Song Attributes

Attribute	Description	Example value
author	song author	See Author attributes
buy	song available for sale (boolean)	True
description	description	My favourite song
download	song available for downloading (boolean)	False
duration	duration in milliseconds	276000
entities	available formats for the song	See Entity attributes
image	image associated to the song	/mexels/ebb39b729b-78a4-4591-bc13-640d72c47b33
locked	lock state (boolean)	False
songID	string ID	ebb39b729b-78a4-4591-bc13-640d72c47b33

numLikes	number of like hits	48
owner	song owner	See Owner attributes
permalink	permalink of the song	85
tags	information related to song	See Tag attributes
title	song title	Happy bells
totalPrice	price for the song (dollars)	4.00
type	item type	song
uri	uniform resource identifier for the song	@melomics/8
visibility	visibility for other users	Public

Author Attributes

Attribute	Description	Example value
artistName	author artist name	Melomics
avatar	author avatar url	https://secure.gravatar.com/avatar/e92c4472c967de848b7958c6c2f53c0d?d=identicon&rg
description	author description	Melomics is the first computer that has mastered human musical language.
email	author email	melomics@melomics.com
id	author ID	5527d30c-25af-46e5-8707-0428741afb4a
lang	author language	en
name	author name	Melomics

permalink	permalink of the author	8
showEmail	show or hide author email	False
surname	author surname	Comp
uri	uniform resource identifier for the auth	@melomics
username	author username	Melomics

Entity attributes

Attribute	Description	Example value
midi	.midi format	5527d30c.mid
mp3	.mp3 format	5527d30c.mp3
pdf	.pdf format	5527d30c.pdf
xml	.xml format	5527d30c.xml

midi, mp3, pdf and xml attributes

Attribute	Description	Example value
buy	format available for sale (boolean)	True
download	format available for downloading (boolean)	True
price	price of the format	\$1.00

Owner Attributes

Attribute	Description	Example value
artistName	owner artist name	Melomics
avatar	owner avatar url	https://secure.gravatar.com/avatar/e92c4472c967de848b7958c6c2f53c0d?d=identicon&rg
description	owner description	Melomics hosts the largest repository of professional music.
email	owner email	melomics@melomics.com
id	owner ID	5527d30c-25af-46e5-8707-0428741afb4a
lang	owner language	en
name	owner name	Melomics
permalink	permalink of the owner	8
showEmail	show or hide owner email	False
surname	owner surname	Comp
uri	uniform resource identifier for the author	@melomics
username	owner username	Melomics

Tag attributes

Attribute	Description	Example value
-----------	-------------	---------------

effect	Effect of the song	tremolo
instrument	Musical instrument used to perform the song	piano
style	Musical style for a song	electronica
substyle	Musical substyle for a song	dance
tempo	Represent the speed of the song in bpm	100
Beat	Steady pulse of the song.	quaternary
Dynamics	The loudness or softness of the song	forte
Texture	Complexity of the song	monophonic

Pagination

As shown above, a search usually returns a list of *songs*. You can limit how many items you retrieve as well as the number of items listed per page by using the following parameters:

- **_from**: determines the item from which the search starts to show results. For instance, imagine that a search returns 100 *songs* but you consider that 40 *songs* are sufficient. By using `_from=60` the search will only display the items between 60 and 100.
- **size**: items per page. It can take values from 1 to 10 (default is 10).

Searching for songs

One of the most common things to do with the Melomics API is to search *songs* matching given parameters. These search parameters can be owner, instrument, style, duration, etc. *Songs* can be identified by id as well. Searching for songs can take several forms, which are shown as following:

Searching by id

The easiest way to search for a *song* is by id. You need to add the id at the end of the call.

```
http://melomics.com/api/songs/[id]
```

Example:

```
http://melomics.com/api/songs/jf96ecdeea-1d59-4b3f-8d81-38a664f8afb3
```

Response:

```
{
  "author": {
    "artistName": "Melomics",
    "avatar":
      "https://secure.gravatar.com/avatar/e92c4472c967de848b7958c6c2f53c0d?d=identicon&rg",
    "description": "<p><strong>melomics</strong> is the first computer that has mastered human
musical language.</p>\n\n<p>With more than a billion songs, its production constitutes the biggest
collection of music in the world, and the main contribution to <strong>melomics</strong>
catalogue.</p>\n",
    "email": "",
    "id": "5527d30c-25af-46e5-8707-0428741afb4a",
    "lang": "en",
    "name": "",
    "payment": {
      "email": "melomics.spain@gmail.com"
    },
    "permalink": "melomics",
    "showEmail": false,
    "surname": "",
    "uri": "@melomics",
    "username": "melomics"
  },
  "buy": false,
  "description": "",
  "download": false,
  "duration": 359000,
```

```
"entities": {
  "midi": {
    "buy": false,
    "download": false,
    "price": -1.0
  },
  "mp3": {
    "buy": false,
    "download": false,
    "price": -1.0
  },
  "pdf": {
    "buy": false,
    "download": false,
    "price": -1.0
  },
  "xml": {
    "buy": false,
    "download": false,
    "price": -1.0
  }
},
"image": "/mexels/jf96ecdeea-1d59-4b3f-8d81-38a664f8afb3",
"like": null,
"locked": false,
"songID": "jf96ecdeea-1d59-4b3f-8d81-38a664f8afb3",
"numLikes": 0,
"owner": {
  "artistName": "Melomics",
  "avatar":
"https://secure.gravatar.com/avatar/e92c4472c967de848b7958c6c2f53c0d?d=identicon&rg",
  "description": "<p><strong>melomics</strong> is the first computer that has mastered
human musical language.</p>\n\n<p>With more than a billion songs, its production constitutes
the biggest collection of music in the world, and the main contribution to
<strong>melomics</strong> catalogue.</p>\n",
  "email": "",
  "id": "5527d30c-25af-46e5-8707-0428741afb4a",
  "lang": "en",
  "name": "",
  "payment": {
    "email": "melomics.spain@gmail.com"
  },
  "permalink": "melomics",
  "showEmail": false,
  "surname": "",
  "uri": "@melomics",
  "username": "melomics"
```

```
    },
    "permalink": "7",
    "tags": [
      "effect:trill",
      "effect:tremolo",
      "tempo:120bpm",
      "effect:mordent",
      "tempo:52bpm",
      "instrument:violin",
      "instrument:piano",
      "tempo:68bpm",
      "effect:grace",
      "effect:pizzicato"
    ],
    "title": "7",
    "totalPrice": -1.0,
    "type": "song",
    "uri": "@melomics/7",
    "visibility": "public"
  }
}
```

Searching by parameters

Searching by id is important, but not necessarily that interesting. The power of the Melomics API comes through the possibility of searching for *songs* in some other ways. For instance, you may be looking for *songs* which is performed with a specific instrument or belongs to a specific style.

```
http://melomics.com/api/songs/?q=[QUERY]
```

This call returns a list of *songs* matching the given parameters. By default, results are ordered by relevance measured as the success of matching. The example below searches for *songs* related to instrument piano.

```
http://melomics.com/api/songs/?q=%23instrument:piano
```

You can also use the boolean connectives *and*, *or* and *not*, even parentheses. The example below will return *songs* related to instrument violin excluding those related to piano.

```
http://melomics.com/api/songs/?q=not%20%23instrument:piano%20and%20%23instrument:violin
```

And this call search for *songs* whose owner is Melomics and its duration is between 2:30 and 4:40:

[http://melomics.com/api/songs/?q=%23username:melomics%20and%20\[2:30-4:40\]](http://melomics.com/api/songs/?q=%23username:melomics%20and%20[2:30-4:40])

As commented above many parameters can be used in order to make sure the search is as accurate as possible. Below, the available query parameters (all optional) are described:

Parameter	Description
Owner	User who is the owner of the song
Instrument	Musical instrument used to perform the song such as piano, violin, flute, etc.
Style	Musical style for a song
Substyle	Musical substyle for a song
Tempo	Represent the speed of the song such as 100 bpm, 108 bpm, etc...
Duration	Desired duration in [mm:ss] format of match song. To search songs of a specific duration use [duration]. By default songs are given some slack: They are allowed to be up to 10 seconds shorter or longer than the specified duration. You can also use [duration_1-duration_2] if you want to search in a duration range.
Beat	Steady pulse of the song such as 4/4, quaternary, etc...
Dynamics	The loudness or softness of the song such as f, fff, etc...
Effect	Effect of the song such as slap, tremolo, etc...
Texture	Complexity of the song

Searching for users

```
http://melomics.com/api/users/[username]
```

This call returns information about the user [username].

Searching for user's public *playlists*

An user can group *songs* into lists. These lists receive the name of *playlists*, which can be identified by id and shared to the world. Melomics API provide methods for searching these user's public *playlists*. An user can only add *songs* that he owns to a *playlist* and a *song* can belong to multiple *playlists*.

Listing all the public *playlists* of specific user

```
http://melomics.com/api/users/[username]/playlists
```

This call returns a list with all the public *playlists* for the user [username]. Pagination parameters (`_from`, `size`) can be used to customize the way in which results are displayed.

Searching for a specific public *playlist* of an user

```
http://melomics.com/api/users/[username]/playlists/[playlistID]
```

This call returns a list with *songs* belonging to the *playlist* [playlist] for the user [username]. In this case, pagination parameters (`_from`, `size`) can also be used to customize the way in which results are displayed.

Searching for user's public content

```
http://melomics.com/api/users/[username]/contents/[permalink]
```

This call returns the public content (*playlist* or *playlist* list) that the user [username] owns by using the specific permalink [permalink] for this content.

OAuth authentication

Melomics' API uses the protocol OAuth 1.0 ([RFC 6749](#)) in order to authenticate users. The API keys can be obtained at <https://melomics.com/oauth/developer> (you can follow the link from your profile's page). Since the API keys are unique for each user, the provided description should be general enough to cover your purposes as a developer, and not to reflect a specific application. Once you got the keys, they won't be active until whitelisted.

Temporary credential request (RFC section 2.1)

```
https://melomics.com/api/oauth/initiate
```

Out-of-band (oob) configuration is supported

Resource owner authorization (RFC section 2.2)

```
https://melomics.com/oauth/authorize
```

A callback could be provided to cover rejection and errors:

```
https://melomics.com/oauth/authorize?...&error_callback=[URI]&...
```

In case of error, the browser is redirected to `error_callback` providing a parameter `callback_status=<error code + description>`.

Access token request (RFC section 2.3)

```
https://melomics.com/api/oauth/token
```

Extra requirements:

- The nonce token must be an alphanumeric string with length between 20 and 30 characters:
`[a-zA-Z0-9]{20,30}`
- HMAC-SHA1 signature method is preferred.

API Calls Table

API call	Args	Return	Description
/songs	- q - _from - size	{ songs: [song], more: bool }	Return a list with the songs matching given parameters (i.e: owner, instrument, style, duration, etc ...)
/songs/[id]		song	Return information about the song with given id
/songs/[id]/like		{ like, numLikes }	Return information of “likes” of the song with given id
[POST] /songs/[id]/like	- like: true/false	{ } if ok	Put or remove a song from favorites (OAuth login required)
/users/[username]		user (json): { ... }	Return information about the user [username]
/users/[username]/contents/[permalink]		user (json): { ... }	Return information about the public content [permalink] for the user [username]
/users/[username]/playlists	- q - _from - size	playlists: [{ from to ... }]	Return a list with information about the public playlists for user [username]
/users/[username]/playlists/[playlistID]	- q - _from - size	playlists: [{ from to ... }]	Return information about the public playlist [playlistID] for user [username]