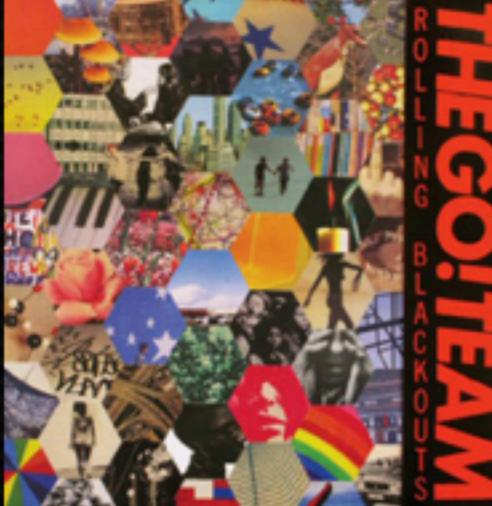


Spotify APIs



Spotify Premium

daft punk

devnevyn		Starred					
		Track	Buy	Artist	Time	Album	Added
What's new		★ Business Car Park 9	(S)	The Black Dog	5:09	Music For Real Airports	Tue Jan 11
Radio		★ Say You Love Me - Original Mix		Slok	8:18	Rain	Mon Jan 10
Play queue		★ Black Dot, White Spider		Seeland	3:14	How to Live	Mon Jan 10
Inbox 1		★ Five Four Child Voice		Fridge	9:21	Happiness	Sun Jan 9
Library		★ Fall		Daft Punk	1:23	TRON: Legacy	Sat Jan 8
Starred	★	★ The Game Has Changed		Daft Punk	3:25	TRON: Legacy	Sat Jan 8
Local files		★ Welcome to Lunar Industries	(S)	Clint Mansell	7:11	Moon	Sat Jan 8
Purchases		★ Cutup Piano And Xylophone		Fridge	2:38	Happiness	Sat Jan 8
Playlists		★ Incident		Joris Voorn	4:06		Sat Jan 8
Double Starred		★ No Revolution	(S)	Joris Voorn	5:22	Futurism	Sat Jan 8
Dark side of the moon		★ Flat Beat	(S)	Mr Oizo	5:25	Flat Beat	Sat Jan 8
Apollo Throwdown		★ Bloody Cash		Marco Carola	7:00	Making Kontakt – Sound...	Sat Jan 8
The Go! Team (Rolling Blackouts)		★ Recognizer		Daft Punk	2:38	TRON: Legacy	Fri Jan 7
Share	f t	★ Always Returning	(S)	Brian Eno	4:04	Apollo	Fri Jan 7
	THE GO! TEAM	★ Weightless	(S)	Brian Eno	4:35	Apollo	Fri Jan 7
	ROLLING BLACKOUTS	★ End Of Line		Daft Punk	2:36	TRON: Legacy	Thu Jan 6
		★ Vortex	(S)	Carbon Based Lifeforms	6:14	World Of Sleepers	Tue Jan 4
		★ 5 23	(S)	Global Communication	5:23	76:14	Tue Jan 4
		★ grid4		Per Borgman	5:59	Gridlike	Mon Jan 3
		★ Tinderbox - Hans Bouffmyhre Remix		Max Cooper	6:59	Sleazebox EP	Mon Jan 3
		★ Trills	(S)	Gui Boratto	7:12	Trills	Mon Jan 3
		★ On the 1am		Hot Toddy	4:12	Late Night Boogie	Mon Jan 3
		★ Cecta (Krazy Fiesta Version)	(S)	Boeoes Kaelstigen	7:19	Commuting Colour Remixes	Mon Jan 3

1:21 3:20

Online

Metadata
API

libspotify

HTTP

C

Metadata
API

Classic

<http://ws.spotify.com/lookup/l/>

<http://ws.spotify.com/search/l/>

add .json for JSON response; otherwise XML

Experimental

<https://ws.spotify.com/oauth/token>

<https://ws.spotify.com/lookup/l/>

<https://ws.spotify.com/search/l/>

https://ws.spotify.com/2.0/user/<canonical_username>/playlist/<playlist_id>

Plug ?oauth_token= onto query.

Playlist query can return XSPF! Use .xspf

<https://ws.spotify.com/search/1/>

```
curl 'http://ws.spotify.com/search/1/album?  
q=album:robyn&oauth_token=vF9dft4qmT'
```

```
<?xml version="1.0" encoding="utf-8"?>
<albums xmlns:opensearch="http://a9.com/-/spec/opensearch/
1.1/" xmlns="http://www.spotify.com/ns/music/1">
  <opensearch:Query role="request" startPage="1"
searchTerms="album:robyn"/>
  <opensearch:totalResults>11</opensearch:totalResults>
  <opensearch:startIndex>0</opensearch:startIndex>
  <opensearch:itemsPerPage>100</opensearch:itemsPerPage>
  <album href="spotify:album:6ZGoBOEAufrFWYDKcgTs2">
    <name>Det Bästa Med Robyn</name>
    <artist href="spotify:artist:6UE7nl9mha6s8z0wFQFIZZ">
      <name>Robyn</name>
    </artist>
    <id type="upc">828768772928</id>
    <image>http://open.spotify.com/image/
e59dafe49a190e052db58c8ca3f7fd050e154017</image>
    <popularity>0.79952</popularity>
    <availability>
      <territories>CN DK EE FI HK ID TN KB LT LV NO PH SE
```

<https://ws.spotify.com/lookup/1/>

```
$ curl 'http://ws.spotify.com/lookup/1/?  
uri=spotify:artist:  
4tZwfgrHOc3mvqYlEYSvVi&oauth_token=vF9dft4qmT'
```

```
<?xml version="1.0" encoding="utf-8"?>  
<artist xmlns="http://www.spotify.com/ns/music/1">  
  <name>Daft Punk</name>  
  ...  
</artist>
```

Artists: Add &extras=album or albumdetail

Albums: Add &extras=track or trackdetail

Metadata
API

Want to use classic API? See
<http://developer.spotify.com/en/metadata-api/overview/>

https://ws.spotify.com/2.0/user/<canonical_username>/playlist/<playlist_id>

```
$ curl https://ws.spotify.com/2.0/user/jon/playlist/  
6YROYeZS57gI5eOZd2pFTg?oauth_token=vF9dft4qmT
```

```
{  
  "collaborative": false,  
  "creator": "jon",  
  "time": 1282919965,  
  "version": 639,  
  "title": "Favorites",  
  "uri": "spotify:user:jon:playlist:  
6YROYeZS57gI5eOZd2pFTg",  
  "items": [  
    {  
      "album": {  
        "name": "Discovery",  
        "uri": "spotify:album:2n0Rn2Aes5aoNVsU6iWThc",  
        "image": "http://open.spotify.com/image/939a6f415b6bea9081b25c963248b1c6c646f7bb"  
      },  
      "artists": [  
        {  
          "name": "Daft Punk",  
          "uri": "spotify:artist:36QJpDe5unUjzXkD3Dp4D",  
          "image": "http://open.spotify.com/image/36QJpDe5unUjzXkD3Dp4D"  
        }  
      ]  
    }  
  ]  
}
```

Want to use experimental API? See
[https://github.com/rewiredstate/sxsw-hackday-2011/
tree/master/apis/spotify](https://github.com/rewiredstate/sxsw-hackday-2011/tree/master/apis/spotify)
and talk to us for API keys and stuff

We would love feedback on the experimental API!
<https://github.com/spotify/playlist-api>

libspotify



```
static sp_session_callbacks session_callbacks = {  
    .logged_in = &logged_in,  
    .notify_main_thread = &notify_main_thread,  
    .music_delivery = &music_delivery,  
    .metadata_updated = &metadata_updated,  
    .play_token_lost = &play_token_lost,  
    .log_message = &message_arrived,  
    .end_of_track = &end_of_track,  
};
```

```
static sp_session_config spconfig = {  
    .api_version = SPOTIFY_API_VERSION,  
    .cache_location = "tmp",  
    .settings_location = "tmp",  
    .application_key = g_appkey,  
    .application_key_size = 0, // Set in main()  
    .user_agent = "spotify-jukebox-example",  
    .callbacks = &session_callbacks,  
    NULL,  
};
```

start stuff up

```
sp_session_create(&spconfig, &sp);
```

gives you an **sp_session**

Asynchronous API.

Playlist contents, track metadata, login status:
everything loaded lazily from disk & backend

libspotify

```
sp_session_login(sess, username, password);
```

log in;

returns immediately

```
static void logged_in(sp_session *sess, sp_error error)
{
    if (SP_ERROR_OK != error) {
        fprintf(stderr, "Login failed: %s\n",
                sp_error_message(error));
        exit(2);
}
```

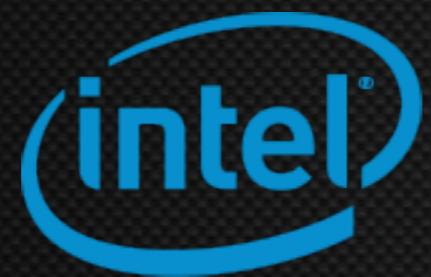
callback later,
check if it succeeded!

libspotify

0.0.7
Released today!

libspotify

YESTERDAY



bugs

TODAY

ARM®



less bugs!

better error handling

better memory management

more social APIs

libspotify

Want to use libspotify? See
<http://developer.spotify.com/en/libspotify/overview/>

Download, and request an API key; put key in appkey.c

Requires PREMIUM!
Ask us for one free month.
(if you have an actual libspotify project)

libspotify from other languages:

- Objective-C wrapper in examples/Randomify
- Ruby bindings: <https://github.com/sarnesjo/greenstripes>
- Python bindings: <https://github.com/winjer/pyspotify>

	Metadata API	libspotify
Play music	✗	✓
View playlists	In experimental	✓
Edit playlists	✗	✓
Use unauthenticated	✓	✗
Search, browse	✓	✓
Language	Any (HTTP)	C

developer.spotify.com

