OnlineFM API

Release date: 04.03.2011

Version: 0.3.1

Contacts of development team:

Georgiy Slobodyrev < gslobodyrev@gadu-gadu.pl>

Andrey Gauk < agauk@gadu-gadu.pl>

Tel.: +38(044) 463-78-55

Kiev, Ukraine

Test token - 4d7106ed1d85c

You should change this token to your personal. Please contact us to get correct token.

Raw data URLs:

- http://online.fm/api/channels.json?token=<TOKEN> - get list of all public channels with IDs.
- http://online.fm/api/fms.json?token=<TOKEN> - get all public FM stations.
- http://online.fm/api/genres_channels.json?token=<TOKEN> - get all genres with channels.
- http://online.fm/api/genres_fms.json?token=<TOKEN> - get all genres with FM stations (TODO).
- http://online.fm/api/playlist/single.json?token=<TOKEN>&id=<CHANNEL_ID> - get past, current and future songs for channel with CHANNEL_ID.
- http://online.fm/api/playlist/currentsong.json?token=<TOKEN>&id=<CHANNEL_ID> - get current song on channel with CHANNEL_ID.
- http://online.fm/api/playlist/list.json?token=<TOKEN>- get current and N future songs on all public channels, N = 5.

Data structure of JSON files:

All objects have fields 'code', 'error' and 'data'.

1. '**code**' contains status code of operation like http status codes:
    - ♦ Value 200 means that all parameters is correct and field 'data' contains data.
    - ♦ If code == 400, parameter 'id' is missing.
    - ♦ If code == 403, parameter 'token' is wrong or missing.
    - ♦ Value 404 means that channel with <CHANNEL_ID> is missing.

2. '**error**' contains error message if exists or null.
3. '**data**' - json with response data or null.

Structure of OnlineFM API source files

css/		CSS file for all templates used in API examples
examples/		Examples of code to use like tests
	channel.php	Example of channel page view
	list.php	Example of channels list page
js/		JS files for play list of each channel item
	channel.js	JS to work with playlists
	jquery.min.js	JQuery minified
lib/		PHP classes of API
	i18n/	Translation files EN, RU, UK
	templates/	Templates of views for channels list item, playlist, playing now,
	Channel.php	Object of this class represents channel item
	ChannelsApi.php	Main class to work with Channels as objects and render views
	GenresApi.php	Class to work with genres as filters
	MemcacheWrapper.php	Memcache wrapper (connect to servers, set and get functions)
	OnlinefmApi.php	**Main API class with public functions**
	config.php	Base configuration of API
	console.php	Script to run via cron each 2-3 minutes

API base configuration

in lib/config.php:

```
return $config = array(
    'token' => '4d7106ed1d85c',
    'urls' => array(
        'channels' => 'http://online.fm/api/channels.json?token=%s',
        'genres_channels' => 'http://online.fm/api/genres_channels.json?token=%s',
        'playlist_list' => 'http://online.fm/api/playlist/list.json?token=%s',
        'playlist_single' => 'http://online.fm/api/playlist/single.json?token=%s&id=%d',
        'playlist_currentsong' => 'http://online.fm/api/playlist/currentsong.json?token=%s&id=%d',
    ),
    'memcache' => array(
        'use' => true,
        'servers' => array( array('host' => 'localhost', 'port' => 11211), ),
        'lifetime' => 86400,
        'key_prefix' => 'ofmapi031',
        'keys' => array(
            'channels' => 'channels',
            'genres' => 'genres',
            'playlist' => 'playlist',
        ),
    ),
    'scripts' => array(
        'include_jquery' => true,
        'js_prefix' => '/js/', // with trailing slash!
        'css_prefix' => '/css/', // with trailing slash!
    ),
    'routes' => array(
        'channel' => '/examples/channel.php?id=%d', //with %d - is used for channel's id
        'genre' => '/examples/list.php?genre=%s', //with %s - is used for genre's slug
    ),
    'dirs' => array(
        'templates' => 'templates/', //relative path from libs/ directory; with trailing slash
    ),
);
```

Caching mechanism

You can choose to use memcache or not.

```
'use' => true,
```

You can change memcache servers settings:

```
'servers' => array( array('host' => 'localhost', 'port' => 11211), ),
```

Period of caching by default is set to 24 hours in seconds

```
'lifetime' => 86400,
```

We'll store on your memcache servers following information:

```
'keys' => array(
    'channels' => 'channels',
    'genres' => 'genres',
    'playlist' => 'playlist',
),
```

channels - flag means that we'll store information retrieved from this URL - http://online.fm/api/channels.json?token=4d5d015fb8eab **- 13.5 KB**

For example there are 2 of 40 channel items:

```
"data":{
        "1026":{
                "id":1026,
                "name":"Pop Hit",
                "url":"http:\/\/online.fm\/ru\/radio\/pop-hit",
                "logo":"http:\/\/st2.ofmimg.com\/images\/channels\/sm\/pop-hit.png",
                "streams_http":["http:\/\/91.214.237.245\/1026",
                                "http:\/\/91.214.237.246\/1026"],
                "streams_rtmp":["rtmp:\/\/91.214.237.247:80\/shoutcast\/1026",
                                "rtmp:\/\/91.214.237.248:80\/shoutcast\/1026"]
        },
        "1022":{
                "id":1022,
                "name":"\u0422\u0435\u0445\u043d\u043e",
                "url":"http:\/\/online.fm\/ru\/radio\/Techno",
                "logo":"http:\/\/st2.ofmimg.com\/images\/channels\/sm\/Techno.png",
                "streams_http":["http:\/\/91.214.237.245\/1022",
                                "http:\/\/91.214.237.246\/1022"],
                "streams_rtmp":["rtmp:\/\/91.214.237.247:80\/shoutcast\/1022",
                                "rtmp:\/\/91.214.237.248:80\/shoutcast\/1022"]
        },
        ......
]}
```

genres flag means that we'll store information retrieved from this URL - http://online.fm/api/genres_channels.json?token=4d5d015fb8eab **- 1.7 KB**

Example of 2 genres that will be used as filter. One genre contain of id, names (ru, en, uk), channels IDs

```
"data":[{
        "id":"7",
        "slug":"disco",
        "name_en":"Disco",
        "name_ru":"\u0414\u0438\u0441\u043a\u043e",
        "name_uk":"\u0414\u0438\u0441\u043a\u043e",
        "channels":[1002,1036]
        },
        {
        "id":"11",
        "slug":"electro",
        "name_en":"Electro",
        "name_ru":"\u042d\u043b\u0435\u043a\u0442\u0440\u043e",
        "name_uk":"\u0415\u043b\u0435\u043a\u0442\u0440\u043e",
        "channels":[1022,1018,1012,1006,1021,1035]
        },
        .......
]}
```

OnlinefmApi methods

See lib/OnlinefmApi.php for details

```
    public getChannels() { ... } - Return array data of channels

    public setChannels($ids) { ... } - Set channel or channels to show

    public getGenres() { ... } - Return array data of genres

    public renderList() { ... } - Return HTML with channels. Use on list page

    public renderChannelPlayingNow() { ... } - Return HTML with channel's "playing now" block with album cov

    public renderChannelPlaylist() { ... } - Return HTML with channel's playlist

    public renderGenres() { ... } - Return HTML code genres

    public renderCSS() { ... } - Return HTML code with CSS meta tags

    public renderListJS() { ... } - Return HTML code with JS meta tags for page with channels

    public renderChannelJS() { ... } - Return HTML code with JS meta tags for channel page
```

Examples

I) How to make a page with list of Online.FM radio channels?

Open /examples/list.php file. We'll show it step by step.

1) Include OnlinefmApi.php

```
include_once(realpath(dirname(__FILE__) . DIRECTORY_SEPARATOR .'..'
                . DIRECTORY_SEPARATOR . 'lib' . DIRECTORY_SEPARATOR . 'OnlinefmApi.php'));
```

2) Create an object of OnlinefmApi. This will allow you to render a list of radio channels sorting items (filter radio channels by genres), render all CSS, JS, set channel ID to show.

All default options are in lib/config.php. You can redefine $options and pass it to OnlinefmApi constructor.

```
$options = array(
    'genre_uri' => '/examples/list.php?genre=%s',
    'channel_uri' => '/examples/channel.php?id=%d',
);
$api = new OnlinefmApi($options);
$genres = $api->getGenres();
```

3) You should define channels you want to show:

```
$api->setChannels(array(1012, 1026)); //You'll see 2 channels
```

Or you can define channels after sorting:

```
if (isset($_GET['genre']) && isset($genres[$_GET['genre']])) {
    $api->setChannels($genres[$_GET['genre']]->channels);
}
```

!!! If you will not set channels you get all channels !!!

On this step we've already defined all objects.

4) We need to create an HTML template of the page and add CSS, JS meta tags. You can do this by executing following code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <?php echo $api->renderCSS(); ?>
    <?php echo $api->renderListJS(); ?>
</head>
<body>
```

So, we have rendered CSS and JS files

```
    <?php echo $api->renderCSS(); ?>
    <?php echo $api->renderListJS(); ?>
```

5) Next is to SHOW our filters by genres:

```
    <ul id="genres">

        <?php echo $api->renderGenres(); ?>

        <?php if ($showAll): ?>
        <li class="genre">
            <h3><a title="All" href="/examples/list.php">All</a></h3>
        </li>
        <?php endif; ?>
    </ul>
```

The next code will return templated HTML code of genres:

```
        <?php echo $api->renderGenres(); ?>
```

6) The last item is to SHOW Online.FM channels templated in HTML:

```
    <ul id="channels">
        <?php echo $api->renderList(); ?>
        <li class="channel-item more-block more-channel-item">
            <a title="All radio channels" href="/examples/list.php">
                <span>All radio channels</span></a>
        </li>
    </ul>
```

Executing

```
        <?php echo $api->renderList(); ?>
```

will allow you to see channels

II) How to make a page with radio channel?

Open /examples/channel.php file.

1) Include OnlinefmApi.php

```
include_once(realpath(dirname(__FILE__) . DIRECTORY_SEPARATOR .'..'
                . DIRECTORY_SEPARATOR . 'lib' . DIRECTORY_SEPARATOR . 'OnlinefmApi.php'));
```

2) Create an object of OnlinefmApi with selected Channel ID. Id of channel came from $_GET['id']

```
$api = new OnlinefmApi;
$api->setChannels($_GET['id']);
$channels = $api->getChannels();
$channel = $channels[0];
```

3) Render META tags, CSS, JS

```
<head>
    <title><?php if ($channel): echo $channel->name; endif; ?></title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <?php echo $api->renderCSS(); ?>
    <?php echo $api->renderChannelJS(); ?>
</head>
```

4) You can show logo and name of Channel

```
<div class="logo">
    <?php if ($channel): ?>
    <img src="<?php echo $channel->logo; ?>" />
    <p><?php echo $channel->name; ?></p>
    <?php endif; ?>
</div>
```

5) Show "playing now" block

```
<div class="playing-now">
<?php echo $api->renderChannelPlayingNow(); ?>
</div>
```

6) Show description of channel

```
<div class="text">
<?php echo $channel->text; ?>
</div>
```

7) Show Playlist of channel

```
<div class="playlist">
<?php echo $api->renderChannelPlaylist(); ?>
</div>
```