



Dance-Tunes user API

Author(s): ir. L.M. Brizzi
Version: 1.0
Date: 08-02-2010
Status: Draft
Copyright: Dance-Tunes.com

Introduction

This document describes the API for Dance-Tunes topic-by-topic. The API serves as an interface between the database and the client front-end. The language used is XML over HTTP. In the near future we will expand the possibilities of this API. You can think of cart/buy functionality, rss feeds, etc.

Request

Each request begins with the following five elements followed by one or more elements specific to the particular request.

```
<cmd> request </cmd>
<session>session number</session >
....
< > .. </ >..... elements specific to the particular request
....
```

The returned result always has the following format:

```
< reply>
.. </ >..... elements specific to the particular request
<security>
  <session>session number</session>
</security>
</reply>
```

Session management

The first request that must be send is the **authenticate** request. The obtained key must be part of this request. In return a session number is send back to the user, which must be used in all successive calls.

When a request is received, a check is made that the session sent with the request, is valid according to the database.

A time stamp is updated in the session for each request received. If a session has had no activity for more than 6 hours, the session is deleted using an automatic process.

The various requests

Below is a list of all available requests. A detailed description of each request is provided in the subsequent chapters.

Request	Description	Must have a session?
Authentication	Register, simple log-in, log in using cookie	No
Search	Search tunes, mixes and albums	Yes
Search_artists	Search artists	Yes
Search_labels	Search labels	Yes
Get_tune_info	All information about a tune or mix.	Yes
Get_recommendations	Recommendations for a tune.	Yes
Get_album_info	All information about an album.	Yes
Get_chart	Returns a chart: either a general one or one for a particular genre.	Yes
Get_genre_info	All information about genres.	Yes
Get_new_releases	Retrieve new releases.	Yes
Get_just_added	Retrieve tunes that have just been added.	Yes
Get_newest_albums	Retrieve recent albums	Yes
Get_tune_info_list	Returns a list of tunes.	Yes
Get_artist_tune_list	Retrieve the tune list for an artist.	Yes
Get_label_tune_list	Retrieve the tune list for a label.	Yes
Get_preview	Retrieve the preview file.	Yes *
Get_picture	Retrieve a picture.	Yes *

* These requests don't have a XML message as a response, but a file.

URL

The basic URL to call the API is as follows:

`http://main.dance-tunes.com/idt/pk_user_api.request?p_message=XML-message`

where xml-message is the request xml message.

Errors

Below is a list of the various error messages that may be returned by the API.

Incorrect command

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
  <error>Not a valid command.</error>
</reply>
```

Wrong session information or incomplete data

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
  <error>Not a valid session or request.</error>
</reply>
```

No session information

```
<?xml version="1.0" encoding="UTF-8"?>  
<reply>  
  <error>No session.</error>  
</reply>
```

Incorrect parameters for the request

```
<?xml version="1.0" encoding="UTF-8"?>  
<reply>  
  <error>Invalid parameters....</error>  
</reply>
```

Requests

In the following paragraphs all the requests are described in detail. A short description is given and an example of a request and a response XML is described.

Authentication

This is the first request that should be issued to obtain a session number for all the successive calls to the API. The <key> element must be filled with a key, which can be obtained on the Dance-Tunes.com website. This key is strictly personal. Besides the key also the username and password of the user must be provided.

In

```
<cmd>authentication</cmd>
<key>ASDEH5F2WW</key>
<username>username</username>
<password>password</password>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
  <result>ok</result>
  <security>
    <session>H0RPJ3HJBVEEOF93XZKU</session>
  </security>
</reply>
```

Search

Searches tunes, mixes and albums for <string>, which must occur in the album title, tune title, version, artist or label. Also from and till can be used for pagination. <order> indicates if the results should be sorted by: **artist**, **date**, **genre** or **title**. <sort> indicates whether the results should be sorted in ascending (asc) or descending (desc) order. <ind_tune>, <ind_mix>, <ind_album> indicate what information should be searched. The value of the indicators is Y or N. The search can be limited by using the <field> tag. Possible values are 'artist', 'title', 'version' or 'label'. If the <field> tag is empty the search will use all mentioned fields. The result is a sorted list of tunes, mixes and albums. This are formatted in the same way as the results returned by get_tune_info and get_album_info. Searches are logged. In order to avoid performance issues, the length of the string to search for must be at least two characters.

In

```
<cmd>search</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<string>Tiesto</string>
<from>1</from>
<till>10</till>
<order>artist</order>
<sort>asc</sort>
<ind_tune>y</ind_tune>
<ind_mix>y</ind_mix>
<ind_album>y</ind_album>
<field>artist</field>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<search_info>
  <total_count>125</total_count>
</search_info>
<searchresult>
  <tune>
    <rownumber>1</rownumber>
    See: get_tune_info
  </tune>
  <tune>
    <rownumber>2</rownumber>
    See: get_tune_info
  </tune>
  ...
  <album>
    <rownumber>10</rownumber>
    See: get_album_info
  </album>
  <album>
    <rownumber>11</rownumber>
    See: get_album_info
  </album>
  ...
</search_result>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Search_artists

Search for artist names. Returned is a list of artists. <sort> indicates whether the results should be sorted in ascending (asc) or descending (desc) order. Also from and till can be used for pagination. In order to avoid performance issues, the length of the string to search for must be at least two characters.

In

```
<cmd>search_artists</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<string>tiesto</string>
<from>1</from>
<till>10</till>
<sort>asc</sort>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<search_info>
  <total_count>125</total_count>
</search_info>
<searchresult>
```

```

<artist>
  <rownumber>1</rownumber>
  <art_id>22</art_id>
  <name>DJ Jean</name>
</artist>
...
</searchresult>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>

```

Search_labels

Search for label names. Returned is a list of labels. <sort> indicates whether the results should be sorted in ascending (asc) or descending (desc) order. Also from and till can be used for pagination. In order to avoid performance issues, the length of the string to search for must be at least two characters.

In

```

<cmd>search_labels</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<string>tiesto</string>
<from>1</from>
<till>10</till>
<sort>asc</sort>

```

Out

```

<?xml version="1.0" encoding="UTF-8"?>
<reply>
<search_info>
  <total_count>125</total_count>
</search_info>
<searchresult>
  <label>
    <rownumber>1</rownumber>
    <lab_id>22</lab_id>
    <name>Black hole</name>
  </label>
...
</searchresult>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>

```

Get_tune_info

Returns tune information for a specific tune or mix. In order to retrieve the recommendations (“users also purchased”), use: get_recommendations. If the requested tune is a mix (<ind_mix>Y</ind_mix>), information about the various tracks are also returned.

In

```
<cmd>get_tune_info</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<tun_id>7</tun_id>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<tune>
  <tun_id>7</tun_id>
  <ind_mix>N</ind_mix>
  <release_date>03-11-2004</release_date>
  <date_added>03-11-2004</date_added>
  <title>Thunder anthem</title>
  <version>Weird mix</version>
  <art_id>45</art_id>
  <art_name>Dj Weirido</art_name>
  <lab_id>45</lab_id>
  <lab_name>DutchBad Records</lab_name>
  <sge_name>acid</sge_name>
  <sge_id>5</sge_id>
  <alb_name>Album div</alb_name>
  <alb_id>11</alb_id>
  <preview_duration_sec>240</preview_duration_sec>
  <duration>3:52</duration>
  <size_mb>5.30432224273681640625</size_mb>
  <bitrate>192</bitrate>
```

Optional: Mix info

```
</tune>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Where mix info has the following format:

```
<mix_tracks>
  <track>
    <track_nr>1</ track_nr>
    <title>Yo</title>
    <version>1</version>
    <art_name>Joop</art_name>
  </track>
  ....
</mix_tracks>
```

Get_recommendations

Returns the top 3 recommendations (users who purchased this, also purchased...) for a specific tune.

In

```
<cmd>get_recommendations</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
```

```
<tun_id>7</tun_id>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<recommendations>
  <recommendation>
    <tun_id>7</tun_id>
    <title>Yvette</title>
    <ind_mix>N</ind_mix>
    <version>Vette remiXX</version>
    <art_name>Dj Buzz Fuzz</art_name>
  </recommendation>
  <recommendation>
    <tun_id>159</tun_id>
    <title>Bmx</title>
    <ind_mix>N</ind_mix>
    <version />
    <art_name>2 Times Eleven</art_name>
  </recommendation>
  <recommendation>
    <tun_id>16</tun_id>
    <title>XTC Love</title>
    <ind_mix>N</ind_mix>
    <version>re-mix</version>
    <art_name>Bertocucci Feranzano</art_name>
  </recommendation>
</recommendations>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_album_info

Retrieves the information about an album and the associated tunes. The first part consists of specific information about the album itself. After this, all tracks on the album are returned, in the same way as they are returned for `get_tune_info`.

In

```
<cmd>get_album_info</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<alb_id>7</alb_id>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<album>
<album_info>
  <alb_id>1</alb_id>
  <release_date>14-10-2004</release_date>
  <date_added>01-02-2005</date_added>
```

```

<art_id>99</art_id>
<art_name>Benjamin Bates</art_name>
<lab_id>9</lab_id>
<lab_name>DutchBad Records</lab_name>
<gen_name>Tech-house</gen_name>
<title>Album div</title>
</album_info>
<album_tracks>
  <tune>
    <track_nr>1 </track_nr>
    See: get_tune_info
  </tune>
  <tune>
    <track_nr>2 </track_nr>
    See: get_tune_info
  </tune>
  ...
</album_tracks>
</album>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>

```

Get_chart

This request can be used to get a chart. The <sge_id> element is used to determine which genre the element refers to. If the element is 0, a general chart is generated. The <count> element can be used to specify the number of tunes that are returned. Also from and till can be used for pagination.

In

```

<cmd>get_chart</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<sge_id>sge_id</sge_id>
<count>40</count>
<from></from>
<till></till>

```

Out

```

<?xml version="1.0" encoding="UTF-8"?>
<reply>
<chart>
  <total_count>16</total_count>
  <tune>
    <position>1</position>
    <last_position>2</last_position>
    See: get_tune_info
  </tune>
  <tune>
  ....
  </tune>
</chart>

```

```
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_genre_info

This request can be used to retrieve a list of all genres in the system. The <sge_id> can be used in other calls e.g. get_chart.

In

```
<cmd>get_genre_info</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<genres>
  <genre>
    <sge_id>1</sge_id>
    <sge_naam>house</sge_naam>
  </genre>
  <genre>
    <sge_id>2</sge_id>
    <sge_naam>trance</sge_naam>
  </genre>
</genres>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_new_releases

This request can be used to retrieve a list of new releases. The <sge_id> element is used to determine which genre the new releases refer to. The number of tunes are specified by the <count> element. Also from and till can be used for pagination.

In

```
<cmd>get_new_releases</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<sge_id>111</sge_id>
<count>12</count>
<from></from>
<till></till>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<new_releases>
  <total_count>12</total_count>
  <tune>
```

```
    <positions>1</position>
    See: get_tune_info
</tune>
<tune>
...
</new_releases>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_just_added

This request can be used to retrieve a list of just added releases. The <sge_id> element is used to determine which genre the new releases refer to. The number of tunes are specified by the <count> element. Also from and till can be used for pagination.

In

```
<cmd>get_just_added</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<sge_id>111</sge_id>
<count>12</count>
<from></from>
<till></till>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<just_added>
  <total_count>12</total_count>
  <tune>
    <position>1</position>
    See: get_tune_info
  </tune>
  <tune>
...
</just_added>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_newest_albums

Returns albums per genre ordered by release/just added date. The number of albums are specified by the <count> element. Also from and till can be used for pagination.

In

```
<cmd>get_newest_albums</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<gen_id>4</gen_id>
<count></count>
```

```
<from>1</from>
<till>22</till>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<albums>
  <album>
    <position>1</position>
    <album_info>
      See get_album_info, without track info !
    </album_info>
  </album>
  ...
</albums>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_tune_info_list

This request can be used to retrieve information about multiple files. It is actually the repeated calling of get_tune_info.

In

```
<cmd>get_tune_info_list</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<tun_id>34</tun_id>
<tun_id>744</tun_id>
<tun_id>3887</tun_id>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<tune_list>
  <tune>
    <rownumber>1</rownumber>
    See: get_tune_info
  </tune>
  ...
</tune_list>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_artist_tune_list

This request retrieves a tune list for the artist specified by the element <art_id>. The number to return is specified by the <count> element. Ordering is descending release date. Also from and till can be used for pagination.

In

```
<cmd>get_artiest_tune_list</cmd>
<session>HORPJ3HJBVEEOF93XZKU</session>
<count>23</count>
<from>1</from>
<till>3</till>
<art_id>art_id</art_id>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<total_count>10</total_count>
<tune_list>
  <tune>
    <rownumber>1</rownumber>
    See: get_tune_info
  </tune>
  ...
</tune_list>
<security>
  <session>HORPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_label_tune_list

This request retrieves a tune list for the label specified by the element <lab_id>. <order> indicates if the results should be sorted by: **artist**, **date** or **title**. <sort> indicates whether the results should be sorted in ascending (asc) or descending (desc) order. It's possible to set the maximum number of returned rows by the <count> element. Also from and till can be used for pagination.

In

```
<cmd>get_label_info_list</cmd>
<session>HORPJ3HJBVEEOF93XZKU</session>
<count>aantal</count>
<lab_id>12333</lab_id>
<from>2</from>
<till>22</till>
<order>artist</order>
<sort>asc</sort>
```

Out

```
<?xml version="1.0" encoding="UTF-8"?>
<reply>
<label_info>
  <totaal_aantal>125</totaal_aantal>
</label_info>
<tune_list>
  <tune>
    <rownumber>1</rownumber>
    See: get_tune_info
  </tune>
```

```
<tune>
<rownumber>2</rownumber>
See: get_tune_info
</tune>
....
</tune_list>
<security>
  <session>H0RPJ3HJBVEEOF93XZKU</session>
</security>
</reply>
```

Get_preview

This request retrieves the preview for a tune.

In

```
<cmd>get_preview</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<tun_id>3324</tun_id>
```

Out

The file (not XML).

Get_picture

This request retrieves the picture for a tune or album. The type can be AL for an album or TU for a tune. The URL is also a little different for this request:

In

```
<cmd>get_picture</cmd>
<session>H0RPJ3HJBVEEOF93XZKU</session>
<type>TU</type>
<id>2345</id>
```

Out

The file (not XML).