

# **QA7 and LVA7 for Docker Site Services**

# Table of Contents

Introduction.....	3
Understanding The Site.....	4
Goals of the site.....	4
Goals of the Server side.....	4
Goals of the User Interface.....	4
Basic elements of the user interface.....	4
A few notes on reset.php.....	5
The menu.....	5
Site Internals.....	6
Site Configuration.....	6
Necessary configuration.....	6
Delete Configuration.....	6
Timeout configuration.....	6
Additional configuration.....	7
Site Structure.....	7
Nemesysco Docker Service API.....	9
Get status.....	9
Set License.....	9
Analyze an audio file.....	10
Get Version.....	10
Get supported formats & codecs.....	11
User Control with userID.....	12

## **Introduction**

This document follows the document “QA7 and LVA7 for Docker - Installation Guide”. It supposes that you already have the service running and working.

If you haven't please install the QA7 or LVA7 service first.

# Understanding The Site

## Goals of the site

The site as number of goals:

1. to serve as an interface to the docker services
2. to serve as basic user interface for:
  - entering a new license and getting the current license informations
  - serve as basic user interface for post-processing analysis
3. to give APIs for the docker functions

## Goals of the Server side

The server side is written in PHP 7 and the main goal is to be as portable as possible, this includes:

- Using only basic PHP, without extensions
- Not using any database

There is still the needs of using a unique identification for each request to the docker and each answer received. The unique identification is generated by incremented a number stored in:  
dir/counter

## Goals of the User Interface

## Basic elements of the user interface

The user interface has been build to be as basic as possible, so it can be easily adapted to any needs. This includes:

1. Very basic HTML
2. Small pages
3. Optionally no JavaScript

There are 6 (or 7) pages that are accessible to the user:

1. index.php
2. status.php
3. license.php
4. analyze.php
5. version.php
6. formats.php
7. reset.php (normally this page will not be shown, it is the only page that contains javascript even if the configuration variable \$noJS is true)

These pages can be grouped as follow:

- **General informations: There is no reason to keep them after the site is finalized**

- index.php : list these pages
- version.php : print the version of the site and of the SDK
- formats.php: print the list of codecs and sound formats supported by the ffmpeg tool inside the docker
- **License tools: These are useful for the administrator**
  - status.php : give the current license database
  - license.php : allow to enter a new license
- **Post-Processing Analysis Tool: Allow a user to analyze an audio file**
  - analyze.php : allow to upload a file to the docker and get the resulting analyze
- **Delete the license: by default this is not used, or only by the administrator!**
  - reset.php : delete the current license, stop the server, and restart the server to generate a new SysID

## **A few notes on reset.php**

As was previously noted, reset.php is normally not shown. There are number of reasons for this approach:

1. There is a better approach. It is to stop and restart the current instance, for instance by running the script LVA7\_reset.sh or QA7\_reset.sh
2. The approach used by reset.php take time (often around 1 minute, sometimes more)

The page reset.php was created before LVA7\_reset.sh and QA\_reset.sh was created, and they are a cleaner solution to the same problem.

## **The menu**

The menu bar for all these pages is defined in a the page \_menu.php

# **Site Internals**

## **Site Configuration**

The configuration file is present in \_config.php

The configuration variables can be separated into different categories:

- Necessary configuration: \$APP, \$rootPATH, \$rootURL
- File delete configuration: \$DEL\_FILES, \$DEL\_RECORD\_FILES, \$STORE\_JSON
- Timeout configuration: \$PING, \$PING\_TIMEOUT, \$DEL\_FORGOTTEN\_FILES
- Additional configuration: \$SELECTED\_API, \$ALLOW\_RESET, \$READ\_URL, \$UNSUPPORTED\_OPTIONS

## **Necessary configuration**

These variables determine which application you are using and where is the site.

They are automatically updated and set during the installation

- \$APP : define which application you installed (“LVA7” or “QA7”)
- \$rootPATH: define where on your hard drive are the QA7/LVA7 site installed
- \$rootURL: define how to access the QA7/LVA7 site through your web browser (the URL of that site)

## **Delete Configuration**

These are files that determine which generated files must be stored and which must be deleted.

By default

- \$DEL\_FILES: delete or not the files used in the share/ sub-directory (use for the communication with the docker instance)
  - \$DEL\_FILES=0; //don't delete those files (can be useful for debugging)
  - \$DEL\_FILES=1; //delete those files (default value & normal use)
- \$DEL\_RECORD\_FILES: delete uploaded audio files stored in record/ sub-directory
  - \$DEL\_RECORD\_FILES=0; //don't delete audio file (can be useful for debugging)
  - \$DEL\_RECORD\_FILES=1; //delete those files (default value & normal use)
- \$STORE\_JSON: store the docker instance answer as json in the json/ sub-directory
  - \$STORE\_JSON='NONE'; //doesn't store anything in json/ (default value)
  - \$STORE\_JSON='ANALYZE'; //store only the result of analyzed files
  - \$STORE\_JSON='ALL'; //store all the LVA7/QA7 Docker

## **Timeout configuration**

A series of variable used to check the system is working and set timeout:

- \$PING: Set the time (in seconds) to wait before sending a ping request to the docker instance

- This is used to check the docker instance is running
- \$PING=10; // wait 10 seconds (default value)
- \$PING\_TIMEOUT: it is used to assure than temporary files are deleted after a time out
  - \$PING\_TIMEOUT=0; // No timeout
  - \$PING\_TIMEOUT=40; // timeout after waiting for 40 seconds (Default)
- \$DEL\_FORGOTTEN\_FILES: timeout for files that should be deleted after a times out
  - \$DEL\_FORGOTTEN\_FILES=\$PING\_TIMEOUT\*2;

Note that to function well the timeout requires the following delay order:

\$PING < \$PING\_TIMEOUT < \$DEL\_FORGOTTEN\_FILES

## **Additional configuration**

- \$SELECTED\_API: this sites has 3 different API
  - \$SELECTED\_API="nmsAPI"; // This is the usual and default API (\_nmsAPI.php
  - \$SELECTED\_API="HTML"; // enable to check \_api.php and has same exactly the same functionality than nmsAPI
  - \$SELECTED\_API="win\_webAPI"; // it uses APIs compatible to the windows version.
  - \$SELECTED\_API=""; give the option to select on each page which API to use
  - nmsAPI is the normal API for the site.
- \$ALLOW\_RESET: Show the option to reset the license (as a new page in the site)
  - \$ALLOW\_RESET=true; // allow to reset the license
  - \$ALLOW\_RESET=false; // disallow to reset the license (default)

The normal method is to delete the current instance and build a new one.
- \$READ\_URL: determe how to read the answer of a page
  - \$READ\_URL='FILE\_GET\_CONTENTS'; // uses the php function file\_get\_contents
  - \$READ\_URL='CURL'; // uses curl (default)
  - file\_get\_contents doesn't work with SSLv3. This is why 'curl' is the default
- \$UNSUPPORTED\_OPTIONS: allow to return advanced parameters.
  - UNSUPPORTED\_OPTIONS=0; // don't return the advanced parameters (default)
  - UNSUPPORTED\_OPTIONS=1; // returns the advanced parameter

## **Site Structure**

- Site configuration:
  - \_constants.php : contains a list of variables which should not be modified
    - In older version of the site, \_constant.php didn't exists and its contents was stored in \_config.php
  - \_config.php : contains a list of variables which can be modified to suit the user needs
- User Interface:
  - \_menu.php : set the menu for the site
  - index.php : show the index page and a short description of each page
  - status.php : current license status. This is the page where you get SysID which is needed to set a license
  - license.php : set a new license
  - analyze.php : upload an audio file and analyze it

- version.php : return the version of the site and the version of the SDK in the docker instance
- formats.php : return the list of codecs and audio formats supported by the ffmpeg present in the docker Instance
- reset.php : delete the current license
- Main functionality:
  - \_func.php : diverse function used by the system
  - \_nmsAPI.php : Send a request to the docker instance, and returns the answer
  - \_answer.php : Check for the docker instance's answer, and also for timeout
  - \_userCheck.php: determine if a user is authorized to make the requested action – It can be modified to suit your need
- API Interface
  - \_api.php : this is the API interface to all the offline services
- interface for win\_webAPI :
  - GetLic/ : returns the current license status
  - SetLic/ : set a new license
  - Analyze/ : analyze an uploaded file
  - Version/ : returns the current site and SDK version
  - Formats/ : return the list of codecs and audio formats supported by the ffmpeg present in the docker Instance
  - Reset/ : delete the current license
- Important directories:
  - dir/ : store the file counter which count the different requests. It is used to make sure that each request as a unique id
  - records/ : it is the directory where audio file to analyze needs to be uploaded. This directory is shared with the docker instance
  - share/ : This is the directory that is used to transmit requests to the docker instance, and the instance answers. This directory is shared with the docker instance
  - json/ : This is where the docker instances can (optionally) be stored. This depends on the value of \$STORE\_JSON in \_config.php
- Additional tools
  - \_tools.php : independent tools for:
    - converting the json resulting from an analysis (as text) to csv
    - Compute additional statistics (for the moment for LVA7 with advance data)
    - Use the statistics generate to generate important tags on the audio file
  - test\_stat.php : it is a test and example of use of \_tools.php



# **Nemesysco Docker Service API**

\_api.php allows to access all the LVA7/QA7 services from anywhere in your site, and even from an external program. See the python3 example: nmsAnalyze.py

## **Get status**

### Default command

The command to request the status (work with the default parameters):

`_api.php?action=STATUS`

### The possible options

- `userID` : set the userID for the `_userCheck.php` (allow to check that the user is allowed to use that service)

### Default request

The default request:

`_api.php?action=STATUS`

Is equivalent to:

`_api.php?action=STATUS&userID=`

### Example with options

`_api.php?action=STATUS&userID=3424`

## **Set License**

### Default command

The command to request the status (work with the default parameters):

`_api.php?action=LICENSE&key=1a2b3c&data=license_code`

Parameters:

- `license_code` is the license code you receive from Nemesysco. This license needs to be URL encoded
- `key= 1a2b3c` : is necessary and hard coded

### The possible options

- `userID` : set the userID for the `_userCheck.php` (allow to check that the user is allowed to use that service)

### Default request

The default request:

`_api.php?action=LICENSE&key=1a2b3c&data=license_code`

Is equivalent to:

`_api.php?action=LICENSE&key=1a2b3c&data=license_code&userID=`

#### Example with options

`_api.php?action=LICENSE&key=1a2b3c&data=license_code&userID=432`

## **Analyze an audio file**

#### Default command

The command to request the status (work with the default parameters):

`_api.php?action=ANALYZE&URL=filename.mp3`

Parameters:

- filename.mp3 :
  - It doesn't have to be mp3, it can be any format supported by ffmpeg, except pcm
  - The file needs to be in records/ so the docker instance can access it
  - The file name needs to be URL encoded

#### The possible options

- userID : set the userID for the \_userCheck.php (allow to check that the user is allowed to use that service)
- BG : set the noise sensibility (default 1000)
- segMaxLen: Set the maximum size of a segment (1-3, default 2)
- analyzeType : (default: 1)
- toMono : set the file to mono (default toMono=0; to convert to mono: toMono=1)
- fileName: file name (URL encrypted)
- normalize : normalize the file (default normalize=0; to normalize: normalize=1)
- logData : (not used)
- topic : describe the topic – for LVA7 (URL encrypted)
- question : describe the question – for LVA7 (URL encrypted)

#### Default request

The default request:

`_api.php?action=ANALYZE&URL=filename.wav`

Is equivalent to:

`_api.php?  
action=ANALYZE&URL=filename.wav&userID=&BG=1000&segMaxLen=2&analyzeType=  
1&toMono=0&fileName=&normalize=0&logData=&topic=&question=`

## **Get Version**

#### Default command

This command return the version of the site and of the SDK present in the docker:

`_api.php?action=VERSION`

#### The possible options

- `userID` : set the userID for the `_userCheck.php` (allow to check that the user is allowed to use that service)

#### Default request

The default request:

`_api.php?action=VERSION`

Is equivalent to:

`_api.php?action=VERSION&userID=`

#### Example with options

`_api.php?action=VERSION&userID=3424`

## **Get supported formats & codecs**

#### Default command

The command returns the list of formats and codecs supported by the docker instance:

`_api.php?action= FORMATS`

#### The possible options

- `userID` : set the userID for the `_userCheck.php` (allow to check that the user is allowed to use that service)

#### Default request

The default request:

`_api.php?action= FORMATS`

Is equivalent to:

`_api.php?action=FORMATS&userID=`

#### Example with options

`_api.php?action=FORMATS&userID=3424`

## User Control with userID

All the command can receive the argument userID. Before running a command, the run the function userCheck.php from the file \_userCheck.php

```
/**
 * Function: userCheck
 * Action: check the user is authorized to do the selected operation
 * Input:
 *     $userID : user identifier
 *     $action : one of the API actions (ANALYZE, STATUS, LICENSE, VERSION, FORMATS)
 * Return:
 *     true: if accepted
 *     false: if rejected
 * Note: Update this function to check your users
 */
function userCheck($userID, $action)
{
    global $ALLOW_RESET;

    $action=strtolower($action);
    if ($userID==-1)
        return false;

    switch($action)
    {
        case 'status':
            return true;
        case 'license':
            return true;
        case 'analyze':
            return true;
        case 'version':
            return true;
        case 'formats':
            return true;
        case 'newlic':
            if ($ALLOW_RESET==false)
            {
                return false;
            }
            return true;
        default:
            return false; //unknown action
    }
    return true;
}
```

When this function returns true, the action is accepted and the request will be made (if it is valid)

If the function returns false, the requested action is refused and an error is return, like:

```
{"userID":"123","action":"formats","error":"User 123 is not allowed to do formats"}
```

The following request types are checked:

- status: get license status
- license: set a new license
- analyze: analyze an audio file
- version: returns the version of the site and SDK
- formats: returns the list of supported audio formats and codecs
- newlic: delete the current license

At this moment this function is a dummy that will reject any userID equal to -1, and equal all the other.

To make it really functional, it needs to be modified:

1. A database should list each user and their authorization
2. And check against the database that the user is authorized to make the request