

VOICEBASE, INC

# VoiceBase API

---

Developers Guide version 1.1

VoiceBase, Inc.

7/16/2014

## Contents

Overview .....	3
Case sensitivity .....	4
Methods .....	4
Keyword Spotting .....	5
Upload Media (POST) .....	7
Upload Media (GET) .....	14
Get File Status .....	15
List Files by Status .....	16
Get Keywords .....	18
Get SEO Keywords .....	20
Delete Media File .....	22
Get Media File Metadata .....	24
Get Media File Analytics .....	26
Search Content .....	31
Search File .....	33
Keyword Spotting with Groups .....	35
Create Group .....	35
List Groups .....	36
Get Group .....	37
Delete Group .....	38
Update Group .....	38
Folders .....	39
Create Folder .....	40
Delete Folder .....	42
Move Folder .....	44
Copy File (Link) to Folder .....	46
Delete File (link) from a Folder .....	48
List Folder .....	50
Share File .....	52
Get File Shares List .....	55
Create Account .....	56

Add Comment .....	58
Get Comments .....	59
Update Transcript .....	60
Upgrade Transcript .....	62
Update Media File.....	63
Get Transcript .....	65
Get AutoNotes Html .....	67
Get AutoNotes URL.....	68
Set Spot words.....	69
Get Spot words .....	70
Add keyword.....	71
Delete Keyword .....	72
Move Keyword.....	73
GetToken .....	74
Utterance Detectors (Beta, contact VoiceBase Support).....	75
SIP, Real-time and Keyword Spotting (Beta, contact VoiceBase Support).....	75
Partner Integration URLs and Branding.....	76
Partner Integration URLs .....	77
Authentication .....	77
Web SDK (for embedded player and search widgets) .....	77
VoiceBase JWPlayer Plugin .....	77
Media Formats Supported .....	83
Transcript Format .....	84
Sample Programs .....	84

## Overview

The VoiceBase API provides a mechanism for uploading media files, monitoring the status of those files, and retrieving the keywords of files that have completed processing.

You can also use the API to search for keywords in your files, the files shared with you, or public files within the Voicebase system.

The API uses either HTTP GET's with parameters in the URL or HTTP POST's with parameters in the POST body. The results are provided in the HTTP response in JSON format.

Each request requires an API Key and password for authentication. HTTP get requests are of this form:

<https://api.voicebase.com/services?version=xxx&id=xxx&password=xxx&action=xxx&<method-specific-parms>>

(Post requests have similar parameters but in the body of the request)

All Requests are required to have a version, id, password and action.

Parameter name	Description	Notes
version	API version	1.1
apikey	API Key	The API key is provided by VoiceBase and is associated with a user account
password	API Password	API Password. This is typically the users normal account password
action	Action/Method	See section below for all method names
< Method -Specific-params>	Depends on the method field	See each method for details

The response will be returned as JSON in the body of the HTTP response.

All response will contain

- Status Code for this request (success, failure)
- Status Message (optional, in failure cases this response field contains details)
- Request specific information

In order to use VoiceBase API you must first create an account and then apply to VoiceBase to obtain an API Key.

### *Security*

All requests will require an API Key and password so that the system can authenticate the API user.

Some requests require special privilege. For this version of the API, only the special case of file upload method requires special privilege.

Only media files available to the user shall be available through the API. (public files, the users private files or files that have been shared with the user by others).

The API support uses HTTPS using SSL to encrypt the data over the wire. We strongly recommend that you always use SSL by supplying the https:// URL in all VoiceBase API calls (Note, that http is deprecated and will be not be supported in a future release).

### *Case sensitivity*

Parameter names in this doc use camel style formatting (ie theFirstParamter. theSecondParameter) but they are NOT case sensitive.

The url for the api is case sensitive and is all lower case

<https://api.voicebase.com/services>

## **Methods**

### *Upload Media Files*

1. Upload Media – upload a media file for processing (POST version)
2. Upload Media Get – upload a media file for processing (GET version)
3. Get File Status – get the status of a file upload
4. List – retrieve a list of files in the specified state (processing, error, complete)
5. Upgrade Transcript – Order a Human Transcript for a file that is already uploaded and has a machine transcript
6. Update Transcript – This method can be used to replace the existing transcript with a different one. Perhaps an edited version or a human transcript that you already have. The new transcript will be word aligned.

### *Retrieving Content*

1. Get File Metadata – get meta-data for the specified file.
2. Get File Analytics – get analytics for the file (extracted keywords).
3. Get AutoNotes HTML – get HTML with keyword links to player page organized by topic

4. Get Transcript – get transcript in srt, pdf or rtf format
5. Get Autonotes Url – get the url to autonotes player page and a list of urls for each keyword (goes to autonotes page with the keyword search)
6. Get Key Words – retrieves a list of keywords (transcript) with time positions (DEPRICATED, use Get Transcript with format=json)
7. Get SEO Key Words – retrieves a list of plain text keywords (transcript) (DEPRICATED, use Get Transcript with format=txt)

### *Search*

1. Search – search the VoiceBase system for media files.
2. Search File – search a media file for keywords.
3. Search By Owner – For partner integration. Where partner has member accounts. Can search files belonging to a member account.
4. Partner Search – For partner integration. Can search entire partner corpus (all accounts that are part of the partner account set).

### *File Management*

1. Delete File – delete a file
2. Share File – provides access to a file for one or more users
3. Get File Shares List – gets all the
4. Update Media File -- can be used to update the metadata for a file (title, description, etc.)

### *Folders*

1. Create Folder – Creates a new folder in the api owners account
2. Delete Folder – Deletes and existing folder from the api owners account
3. Move Folder – Moved an existing Folder to a new location
4. Copy File to Folder – Copies a file to a folder (files in folders are links/shortcuts)
5. Delete File from Folder – Deletes a file from a folder (remove the link/shortcut)
6. List Folder – List all the contents of a folder (files and subfolders)

### *Account Management/Other*

1. Create Account – creates a new VoiceBase user account

### *Comments*

1. Add Comment – add a comment to the timeline for a specified file
2. Get Comments – get the list of comments for a recording

### *Keyword Spotting*

1. Create Group – create a keyword spotting group with a name and a list of keywords
2. Delete Group – delete the keyword group by name
3. List Groups – list all existing keyword groups

4. Update Group – modify the keyword group (change the keyword list)
5. Get Group – get the definition of a keyword group. Return the list of keywords in the group
6. Set Spot Words – set a list of keywords to be used for keyword spotting. Setting on the account level. (BETA, Contact VoiceBase Support)
7. Get Spot Words – get the list of keywords for this account. . (BETA, Contact VoiceBase Support)

### Upload Media (POST)

Use the upload method to upload a media file to the VoiceBase system for keyword extraction and indexing. You can monitor the status of the file with the getFileStatus method and retrieve a list of keywords when the processing is done with the getSEOKeywords method. Use an HTTP post that includes the audio file as an attachment.

Parameter name	Required	Description	Notes
version	Yes	API version	This document describes version: 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	Password	API Password. This is typically the users normal account password
action	Yes	Value: "uploadMedia"	action=uploadMedia indicates this is an upload request.
file	Yes	File Attachment	This is the media file attached to the multipart post. File Name must be 80 characters or less.
title	No	Name of Media file	
desc	No	Summary of media file content	
lang	No	Code of media language Default is en-US	en-US" US English (or"en") "de" Deutsch "es-MEX" Español (Latin america) "es" Español (España)



			"fr" Français "it" Italiano "nl" Dutch "en-UK" UK English "en-SEA" Southeast Asian English "en-IND" Indian English "en-AU" Australian English
transcriptType	Yes	Value: "machine" or "human"	transcriptType=machine requests a machine transcription. transcriptType=human additionally requests a human transcription, with the type of human transcription taken from the default values for the Voicebase account. Contact Voicebase support for more information.
notesToTranscriber	No	Plain text, notes to the transcriber.	Free form text. Optional. Can be used to send notes to the transcriber if transcriptType=human
transcript	No	Attachment	An attachment containing the user supplied human transcript in plain text.
transcriptURL	No	String (URL pointing to attachment)	A URL pointing to a user supplied human transcript in plain text.
public	No	True or False	Default is false
recordedDate	No	Date-Time of media was recorded	Date format is YYYY-mm-dd HH:mm:ss. Stored as a DATETIME field type. Examples:

			2014/05/08 2014/5/07 12:34
imageUrl	No	URL of image for this recording	Shown on VoiceBase details page
sourceURL	No	URL of the source media. Indicates that VoiceBase should stream the media from this location. If not provided, VoiceBase will save the media and stream from VoiceBase servers.	If this field is provided, VoiceBase servers will delete the audio when it is done indexing it, and it will stream the audio from this location. Otherwise VoiceBase keeps the media on its servers for streaming.
uploadCollection	No	Optional Field that can be used by clients to help bookkeeping. The list by status method can be filtered by upload collection	Can be used when polling for status. For example load a days worth of content with a unique uploadCollection and then you can poll for which files in the uploadCollection are done (and don't need to poll for each file individually)
externalID	No	Optional field that be used by clients to help with bookkeeping. An external ID passed in by the client. Some methods will allow for querying by this ID as well as the native VoiceBase media Id.	String that must be unique
collection	No	Collection or group in which this recording belongs.	For example this could be a radio show. (Not used at this time. For future use such as filters).

autoCreate	No	Optional field used for partner integration along with ownerId. If this field is set to true, and there is not an existing account for ownerId, then the system will create an account for the owner and then assign ownership of the file to that user. The user will receive an email to activate the account specified by VoiceBase account "ownerid". .	Using this feature requires a VoiceBase privileged partner account. Please contact VoiceBase support if you need this capability.
ownerID	No	The file will be added to the account of this user rather than the API users account.	Using this feature requires a VoiceBase privileged partner account. Please contact VoiceBase support if you need this capability.
uploadedCallBack	No	Callback URL to be called when upload complete event occur on voicebase server.	Any HTTP or HTTPS URL
indexedCallBack	No	Callback URL to be called when index complete event occur on VoiceBase server	Any HTTP or HTTPS URL
machineReadyCallBack	No	Callback URL to be called when machine transcript ready event occur on VoiceBase server.	Any HTTP or HTTPS URL
humanReadyCallBack	No	Callback URL to be called when the machine transcript ready event occurs on the voicebase server.	Any HTTP or HTTPS URL
errorCallBack	No	Callback URL to be called when an error event occur on VoiceBase server for this file	Any HTTP or HTTPS url

shareEmails	No	A comma separated list of email addresses. This file will be shared with this list of users.	The email addresses must correspond to Voicebase accounts.
shareMessage	No	Text that can be added to the share email.	If not present, the default email will be sent.
shareSendNotification	No	If true, send a notification to share list. If false, do not send.	Defaults to true
searchHitURL	No	Override the search hit url with this url so not returning the voicebase player url, instead the url of your choice	This is use for building a web app using the web sdk and player plugin. If you want to use your own player pages.
callID	No	Indicates a real-time processing mode. Can set a sequence of small uploads each with same callID. The system will assemble them at the end (when final segment is received). Also can get callbacks	(Beta only, contact VoiceBase support if interested)
segmentNumber	No	Used in near real time mode. Can send a file segment by segment	(Beta only, contact VoiceBase support if interested)
finalSegment	No	Used in near real time mode. This indicates the final segment	(Beta only, contact VoiceBase support if interested)
rtCallbackURL	No	Used in Real time mode. This is the callback upon which all transcript and keyword notifications are sent.	(Beta only, contact VoiceBase support if interested)
transcriptNotifyFlag	No	If true, send transcript notifications for each chunk	(Beta only, contact VoiceBase support if interested)
keywordNotifyFlag	No	If true, send keyword notifications for each chunk.	The system will use the spotwords set at the account level.  (Beta only, contact

			VoiceBase support if interested)
keywordGroup	One or more keyword group names.	Characters with comma separator.	Parameter Is optional.  Specify the names of one or more group that were created through API method 'createGroup'.  Also possible to specify a set of names of groups separated by ','.
Utterances	One or more utterance detectors	Comma separated list of predefined utterance detectors	Optional
folderName	No	Folder name in which to place the recording. The delimiter is "/" (for example "topFolder/subFolder")	Optional.

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE. It also contains status details in the status message field and the unique media ID of this file media file. The media ID can be used later to check the status of the request and to retrieve the results.

Note that the status code (requestStatus) and status message returned by this method indicate the status of the upload request and does not indicate the final status of the request. It will take some time before the file will be available (depending on file size, format and overall system load). You can check the status of the keyword extraction and indexing with the getFileStatus method described later.

An alternative to polling with the getStatus method is to request a callback on an http or https URL of your choice. You can request a call back one or more of these four file processing events:

- i. to get a file upload complete callback, specify the uploadedCallback parameter
- ii. to get a machine transcript available , specify the machinereadyCallBack parameter
- iii. to get a file searchable , specify the indexedCallBack parameter

iv. to get a human transcript available , specify the `humanreadyCallBack` parameter

The callback will include the state, mediaId and externalId (if one was specified). For example

<https://mycallback.server.com/callback?state=machineReady&mediaId=1234&externalId=abcd>

If you specify “machine”, as your transcript type, you can check for machine ready status or ask for machine ready event. If you request “human”, you can use human ready event or status. Even if you select human, machine transcript will be created first and you can check for machine ready status or ask for machine ready event and then later get the human transcript ready event.

Notes regarding the user-supplied transcript option

- It is illegal to have both transcript and transcriptUrl set in the same call. You can set only one of the two.
- If either transcript or transcriptUrl are provided, the system will use the provided transcript in place of the machine transcript.
- The system will align the text to the audio.
- The system will treat this as a human transcript (same as if it was ordered through VoiceBase human transcript service)
- The system will ignore the transcript Type field if either of these fields are set

Response Example:

```
{ "requestStatus": "SUCCESS",  
  "statusMessage": "The request was processed successfully",  
  "mediaId": "4e1f24f441247" }
```

### *PHP Example using curl*

```
<?php  
$version = "1.1";  
$apikey = "your-API-Key";  
$password = "password";  
$url = "https://api.voicebase.com/services";  
$filename = $argv[1];  
$action = "uploadMedia";  
$transcriptType = "machine";  
$params = array(  
    "version"    => $version,  
    "apikey"     => $apikey,  
    "password"   => $password,
```

```

"action"      => $action,
"file"        => "@$filename",
"transcriptType" => $transcriptType,
"title"       => date("l jS \of F Y h:i:s A"),
"desc"        => "file description",
"recordedDate" => date("Y-m-d H:i:s"),
"collection"  => "mystuff",
"public"      => true,
"sourceUrl"   => "http://yoururl.com/audio.mp3",
"lang"        => "en",
"imageUrl"    => "http://debug.quart-soft.com/zuma/test.jpg"
);
set_time_limit(0);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);

curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_LOW_SPEED_LIMIT, 0);
curl_setopt($ch, CURLOPT_LOW_SPEED_TIME, 60);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>

```

## Upload Media (GET)

If your media file is accessible over the Internet you can use the GET form of the upload Media method. All parameters are the same except that instead of a multipart attachment containing the media, there is a mediaURL field containing the URL of the media and the action should be equal to UploadMediaGet instead of uploadMedia.

Parameter name	Required	Description	Notes
action	Yes	Value: uploadMediaGet	action=uploadMediaGet indicates this is an upload request and the location of the media file is specified in the mediaURL field

mediaURL	Yes	Value: Full URL of the media	Media must be accessible over internet. The file name must be 80 characters or less.
----------	-----	------------------------------	--

### Get File Status

Use the get file status method to monitor the status of a file that you have already uploaded. After the file is uploaded, it will take some time for the keywords to be extracted and indexed. You can check the status of the file with this method.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	Value: "GetFileStatus"	action=GetFileStatus indicates this is a getFileStatus request.
mediaID	If no external ID	The media ID is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and search method
externalID	If no media ID	The file ID	Unique ID assigned by the client upload time.



The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE. It also contains the fileStatus. A file status can either be PROCESSING, ERROR, MACHINECOMPLETE or HUMANCOMPLETE

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getFileStatus&mediaid=4e0dca1cb1bb8>

```
{"requestStatus":"SUCCESS",
 "statusMessage":"The request was processed successfully",
 "fileStatus":"PROCESSING",
 "response":"Waiting for Conversion"}
```

#### *PHP Example using curl*

```
<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
$url = "https://api.voicebase.com/services";
$action = "getFileStatus";
$media_id = $argv[1];
$url = "https://api.voicebase.com/services";
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$media_id;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>
```

#### **List Files by Status**

Use the List method to retrieve a list of files that are in the specified state (status). This can be useful for monitoring the progress of the files that you uploaded. When the processing is completed, you can retrieve the keywords with the getSEOKeywords method.

Parameter name	Required	Description	Notes
----------------	----------	-------------	-------

version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is typically the users normal account password
action	Yes	Value: List	action=list indicates this is a List Files request.
status	Yes	The status used in the query. (PROCESSING, MACHINECOMPLETE, HUMANCOMPLETE, ERROR)	This method will do a query for all files owned by the user with the state specified
uploadCollection	No	Limits the list to files belonging to this collection	Useful for uploading and monitoring sets of files.

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE. It also contains a list of mediaIds for the files in the specified status.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=list&status=processing>

Example:

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "Request Completed Successfully",
  "mediaIds":["12345","98765","777777"]
}
```

### PHP Example using curl

```
<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
$action = "list";
$status = "complete";
$url = "https://api.voicebase.com/services";
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&status=".$status;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
$jsonDoc =json_decode($response, true);
?>
```

### Get Keywords

Use the getKeywords method to retrieve the keywords of a media file.

Parameter name	Required	Description	Notes
version	Yes	API version	current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is typically the users normal account password

action	Yes	"GetKeywords"	Indicates this is a get Transcript request.
format	No	Format of results	Default is JSON (only JSON supported at this time)
mediaID	If no external ID	The media file ID	Unique ID that was returned from the UploadFile, List and search methods
externalID	If no media ID	The file ID	Unique ID assigned by the client upload time.
confidence	No	The confidence threshold (value between 0.0 and 1.0)	Default value is 0.75

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. It also contains the fileStatus and a list of keywords. Request Status can be SUCCESS or FAILURE. File status can either be in PROCESSING, MACHINECOMPLETE, HUMANCOMPLETE or ERROR.

The keyword list JSON data will contain position, time and confidence for each keyword.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getKeywords&mediaid=4e0dca1cb1bb8>

Example Results:

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "fileStatus": "COMPLETE",
  "transcript": [
    {
      "p": 1,
      "s": 3457,
      "e": 4118,
      "c": 0.95,
      "w": "to"
    },
    {
      "p": 2,
      "s": 4118,
      "e": 4900,
      "c": 0.84,
      "w": "be"
    },
    {
      "p": 3,
      "s": 4900,
      "e": 5141,
      "c": 0.7,
      "w": "or"
    },
    {
      "p": 4,
      "s": 5141,
      "e": 5542,
      "c": 0.59,
      "w": "not"
    },
    {
      "p": 5,
      "s": 5542,
      "e": 6644,
      "c": 1.01,
      "w": "to"
    }
  ]
}
```

```
        {"p":6,"s":6644,"e":7085,"c":0.14,"w":"be"}}
    }
}
```

### PHP Example using curl

```
<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
$url = "https://api.voicebase.com/services";
$action = "getKeywords";
$media_id = $argv[1];
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$media_id;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
#print out just the keywords
$jsonDoc = json_decode($response, true);
foreach ($jsonDoc["transcript"] as $v) {
    echo $v["w"] . " ";
}
?>
```

### Get SEO Keywords

Use the `getSEOKeywords` method to retrieve the list of keywords for a media file without the time and position information.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account

password	Yes	API Password	API Password. This is typically the users normal account password
action	Yes	Value: GetSEOKeywords	action=GetSEOKeywords indicates this is a get Transcript request.
mediaID	If no external ID	The media file ID	Unique ID that was returned from the UploadFil, list and search method
externalID	If no media ID	The file ID	Unique ID assigned by the client upload time.

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. It also contains the fileStatus and a list of keywords. Request Status can be SUCCESS or FAILURE. File status can either be in PROCESSING, COMPLETE.

The keyword field is a sequence of keywords separated by spaces.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getSEOKeywords&mediaid=4e0dca1cb1bb8>

Example Results:

The response will also contain, requests status, status message, filestatus, transcript type and the transcript itself in JSON format

Example:

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "fileStatus": "MACHINECOMPLETE",
  "keywords": "to be or not to be",
  "transcriptType": "machine"
}
```

*PHP Example using curl*

```

$url = "https://VoiceBase.com/services";
$action = "getSEOKeywords";
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$mediaid;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);

#print the json response
echo $response;
echo "\n";<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
$url = "https://api.voicebase.com/services";
$action = "getSEOKeywords";
$media_id = $argv[1];
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$media_id;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>

```

### Delete Media File

The delete method deletes a file from VoiceBase system and removes it from the indexes.

Parameter name	Required	Description	Notes
----------------	----------	-------------	-------

version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password
action	Yes	Value: DeleteFile	action=DeleteFile indicates this is an delete file request.
mediaID	Yes	The file ID	Use the unique file ID that was returned from the UploadFile method or list method

Example: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=deleteFile&mediaid=4e0dca1cb1bb8>

Example Response:

```
{
    "requestStatus": "SUCCESS",
    "statusMessage": "Request Completed Successfully",
}
```

#### *PHP Example using Curl*

```
<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
$url = "https://api.voicebase.com/services";
$action = "deleteFile";
$media_id = $argv[1];
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
```



```

$url .= "&mediaid=".$media_id;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>

```

### Get Media File Metadata

This method retrieves metadata about the media file from VoiceBase system.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
Apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password
action	Yes	Value: GetFileMetaData	action=GetFileMetaDa ta indicates this is an get file meta data request.
mediaID	Yes if no external ID	The media file ID	Unique ID that was returned from the UploadFile, list and search method
ExternalID	Yes if no media ID	The file ID (optional, either mediaid or externalID should be used not both)	Unique ID assigned by the client at upload time.

Example: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getFileMetaData&mediaid=4e0dca1cb1bb8>

The response will contain metadata for this file. Fields include:

- name -- name of the file
- description -- description of the file
- image\_url -- an image that will be used on the player page
- upload\_collection -- the collection name that can be used in the upload/polling processing
- transcription\_status -- the status of the transcript, valid values are:
- lengthMs -- length of the file in milliseconds
- length -- length of the file as human readable string
- source\_url -- if was uploaded with remote streaming this will have a value
- source\_file -- if the remote streaming was rtmp, this will be the filename for that stream
- streamUrl -- indicates streaming url of the file (if vb streamed this is a http url)
- rtmpUrl -- contains the rtmp stream url (for vb streamed files)
- rtmpFile -- contains the rtmp stream file (for vb streamed files)
- hasVideo -- indicates that the file has video if true
- recordedDate -- in mm/dd/YYYY HH:mm format. If it was not set at upload time then the upload date time is returned
- searchHitUrl -- the url that is returned for each search hit. By default points to VoiceBase url. You can override it to point to your own url.

Example Response:

```
{
  "requestStatus":"SUCCESS",
  "statusMessage":"The request was processed successfully",
  "response":{"title":"My Audio File",
    "description":"my description...",
    "length":" 5 min 57 sec",
    "source_url":"",
    "image_url":"",
    "upload_collection":"",
    "notify_url":"",
    "recordedDate":"10/21/2013 12:23",
    "transcription_status":"Transcribed"}
}
```

### *PHP Example using Curl*

```
<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
```

```

$url = "https://api.voicebase.com/services";
$action = "getFileMetaData";
$media_id = $argv[1];
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$media_id;
$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>

```

### Get Media File Analytics

This method retrieves analytic for the media file from VoiceBase system.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password
action	Yes	Value: GetFileAnalytics	action=GetFileAnalytics indicates this is a get file analytics request.
mediaID	If no external ID	The media file ID	Unique ID that was returned from the UploadFile, list and search method

externalID	If no media ID	The file ID	Unique ID assigned by the client at upload time.
rssGUID	No		If uploaded with podcast feed, this is an alternative unique identifier from the feed
podcastURL	No		If uploaded with podcast feed, this is an alternative unique identifier from the feed
IncludeStartTimes	No	If true, the system will attempt to return the start times	
returnCategories	No	If true, returns the topic structures	Default is false
returnGroups	No	Boolean (true/false). If true returns group hits if any were specified at upload time.	Parameter is optional, default is false
useStemming	No	Boolean (true/false). If true, stemming is used for the group based keyword spotting.	Parameter is optional, default is false

Example: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getFileAnalytics&mediaid=4e0dca1cb1bb8>

The response will contain a keyword array. Each element contains the word itself, count of the occurrences in the file and the rank/importance of the term.

The output is in 4 sections

1. keywords: contains the main keywords in a flat structure (list)
2. categories – keywords organized by topic
3. group – keyword spotting results if groups specified at upload
4. utterances – utterance detection for all utterance detectors specified at upload time

#### Example Response

##### Keywords section

```
{
  "requestStatus":"SUCCESS",
  "statusMessage":"The request was processed successfully",
  "keywords":{
    "0":{"word":"destruction",
      "count":1,
      "t":[31.08] },
    "1":{"word":"lubrizol",
      "count":12,
      "t":[17.59,18.88]},  },
  "transcriptType":"machine"
}
```

##### Topical results (returnCategories=true)

```
"categories":[
  {"name":"Member states of United Nations",
    "score":6.502,
    "type":"category",
    "subcategories": [{"name":"G20 nations", "score":4.63},
      {"name":"G8 nations", "score":2.007}
    ],
    "similarCategories": []
  },
  "keywords": [ {"name":"Venezuela", "internalName":"venezuela", "score":0.624},
    {"name":"Vietnam", "internalName":"vietnam", "score":0.624},
    {"name":"Saudi Arabia", "internalName":"saudi_arabia", "score":0.723},
    {"name":"Mexico", "internalName":"mexico", "score":0.624},
    {"name":"Russia", "internalName":"russia", "score":0.624},
    {"name":"United States", "internalName":"usa", "score":0.759},
    {"name":"China", "internalName":"china", "score":0.652},
    {"name":"Brazil", "internalName":"brazil", "score":0.624},
    {"name":"Qatar", "internalName":"qatar", "score":0.624},
    {"name":"Japan", "internalName":"japan", "score":0.624}
  ],
  {"name":"G20 nations",
    "score":4.63,
```

```

"type":"subcategory",
"similarCategories": [],
"keywords": [
  [{"name":"Saudi Arabia", "internalName":"saudi_arabia", "score":0.723},
  {"name":"Mexico", "internalName":"mexico", "score":0.624},
  {"name":"Russia", "internalName":"russia", "score":0.624},
  {"name":"United States", "internalName":"usa", "score":0.759},
  {"name":"China", "internalName":"china", "score":0.652},
  {"name":"Brazil", "internalName":"brazil", "score":0.624},
  {"name":"Japan", "internalName":"japan", "score":0.624}
]
},
{"name":"Communist states",
"score":2.035,
"type":"category",
"subcategories": [],
"similarCategories": [{"name": "Single-party states", "score":2.035}
]
"keywords": [
  [{"name":"Soviet Union", "internalName":"ussr", "score":0.759},
  {"name":"China", "internalName":"china", "score":0.652},
  {"name":"Vietnam", "internalName":"vietnam", "score":0.624}
]
},
{"name":"G8 nations",
"score":2.007,
"type":"subcategory",
"similarCategories": []
"keywords": [
  [{"name":"Russia", "internalName":"russia", "score":0.624},
  {"name":"United States", "internalName":"usa", "score":0.759},
  {"name":"Japan", "internalName":"japan", "score":0.624}
]
}
],
"keywordsToCategories":[
{"name": "Japan",
"score": 0.624,
"categories": [{"name":"Member states of United Nations", "score":6.502},
  {"name":"G20 nations", "score":4.63},
  {"name":"G8 nations", "score":2.007}
]
},
{"name": "Venezuela"
"score": 0.624
"categories": [{"name":"Member states of United Nations", "score":6.502},
  {"name":"Federal countries", "score":3.255}
]
}
],

```

Group resultsL if returnGroups is true:

```
"groups":[{"name":"KGroup1",
  "type":"group",
  "keywords":[{"name":"supplies","t":["1004.49","1012.91"]},
    {"name":"return","t":["459.43","604.07"]},
    {"name":"school","t":["137.28","281.72","348.06","453.63"]}],
  {"name":"KGroup2",
  "type":"group",
  "keywords":[{"name":"gold pieces","t":["378.86","387.94","405.51"]}]},

  {"name":"KGroup3",
  "type":"group",
  "keywords":[{"name":"field","t":["370.6","371.44","373.75","494.32","532.44","799.07"]},
    {"name":"return","t":["459.43","604.07"]},
    {"name":"closer","t":["635.85","908.93","909.61"]},
    {"name":"things","t":["356.15","660.61","999.7"]},
    {"name":"resolutions","t":["417.13"]}]]}
```

### Utterance detector Results

Results given that 2 utterance detectors (named CHData and type2) were specified at upload time.

```
"utterances":[{"type":"CHdata",
  "name":"Credit Card Holder Data",
  "segments":[{"e":600.51,"u":"The text in
the utterance","s":560.98} ] },
  {"type":"type2",
  "name":"A name for type 2",
  "segments":[{"e":10.12,"u":"The text in the
utterance2","s":123.45} ] }
```

### *PHP Example using Curl*

```
<?php
$version = "1.1";
$apikey = "your-API-Key";
$password = "password";
$url = "https://api.voicebase.com/services";
$action = "getFileAnalytics";
$media_id = $argv[1];
$url .= "?version=".$version;
$url .= "&apikey=".$apikey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$media_id;
```

```

$ch = curl_init();
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>

```

## Search Content

The search method will query the indices for the keywords included in the query string. It will return a list of media files that match your search terms.

You can iterate through large results sets using the “from” and the “to” fields. Typically you will want to retrieve the first N results. You can do this by setting from= 1 and to=N. If a user action want to display the next N results, then you can the method again and set from=N+1 and to=2N

Ranking can be controlled with the rank field.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password
action	Yes	Value: Search	action=Search indicates this is a search request.
from	No	First media iD in the list that is returned	Default to first item
to	No	Last media iD in the list that is returned	Default to 10 <sup>th</sup> item



public	No	Boolean search public or private	Default is true
rank	No	The Ranking criteria (default, uploaddate)	Default ranking is TBS
terms	Yes	Search terms	Array of search terms
unquoted		If the term is a phrase, you can set this flag to true and the system will return only hits that matches the entire phrase matched.	If this flag is not true, the system will return hits for matches of the entire phrase (if it exists) and hits for the individual words in the phrase.

The HTTP response will contain

- RequestStatus
- Statusmessage
- Number of hits (media files)
- List of media Files containing
  - mediald
  - length
  - title
  - detail link (url to the VoiceBase web page for this file)
  - list of terms with matches for this file
    - term
    - list of times for each hit for this term
    - list of text snippets (markers) for hit for this term. Term has html bold tag

Example url: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=search&public=true&from=1&to=25&terms=social%20network>

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "hits": [
    {
      "mediald": "4ebe7f74d8c72",
      "title": "NPR: 11-11-2011 Fresh Air",
      "length": "2729",
      "detaillink": "https://api.voicebase.com/voice_file/public_detail/4064",
      "hits": [
        {
          "term": "work",
          "times": [76.91, 797.13, 1125.35, 1211.1, 1548.68, 1705.7, 2041.19],

```

```

        "markers":[" rolling <b>work</b>, ... ]},
        {"term":"job",
         "times":[515.19,971.99,1437.79,1658.01],
         "markers":[" a terrific <b>job</b> you","t do a good <b>job</b> and",...]}
    ]},
    {"mediaId":"4ebd2a768a3d5",
     "title":"NPR: 11-10-2011 Fresh Air",
     "length":"2735",
     "detaillink":"https://api.voicebase.com/voice_file/public_detail/4063",
     "hits":{"term":"work",
            "times":[261.14,708.27,2584.74],
            "markers":[" legal <b>work</b> contraception"," her <b>work</b> and",...]}
    }},
    "numberOfHits":125}

```

### Search File

The search method will allow you to query a single file for search markers. Hits are returned in time order.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password
action	Yes	Value: SearchFile	action=SearchFile indicates this is a search Media File request.
mediaID	If no external ID	The media file ID	Unique ID that was returned from the UploadFile, list and search method

externalID	If no media ID	The file ID	Unique ID assigned by the client at upload time.
terms	Yes	Search terms	Array of search terms
folderName	No	Folder name in which to search for the terms. File name delimiter is / (for example "topFolder/subFolder")	Search is done for all recordings in the folder specified as well as all subfolders.

The HTTP response will contain

- RequestStatus
- Statusmessage
- Number of hits for this file
- List of terms. Each term containing
  - Search term
  - List of hits for this term Each hit containing
    - Position
    - Time
    - Phrase

Example url: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=searchFile&terms=work%job>

Note in this example the search terms are "work job"

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "hits": {
    "title": "#442: Thugs",
    "length": "3516",
    "numberOfHits": 12,
    "hits": [
      {
        "term": "work",
        "hits": [
          {
            "pos": 1,
            "time": 132.37,
            "phrase": "..... the work that",
            "pos": 9,

```

```

        "time":2187.35,
        "phrase":" i continued on to work the"},
        {"pos":12,
        "time":3236.69,
        "phrase":" sluggish at work .. five"}]],
    {"term":"job",
    "hits":[{"pos":2,
    "time":1193.05,
    "phrase":" good job of making"},
    {"pos":3,
    "time":1604.7,
    "phrase":" tooka job at the" },
    {"pos":11,
    "time":2495.21,
    "phrase":" at her job what"}]}
  ]},
}

```

### Keyword Spotting with Groups

You can automatically do keyword spotting at upload time by using keyword groups. A keyword group is a named set of keywords. You can specify one or more keyword groups at upload time for your files. The VoiceBase system will search the file for each keyword in each group and return all the hits in the get file analytics method.

In addition to the methods described in this section there are also parameters you should use in the Upload methods and in the Get File Analytics methods.

The keyword group parameter is an optional parameter to all upload commands. You can use it to specify one or more groups that have been created with the create Group method. You can specify a set of names of groups separated by ','.

Two new parameters have been added to the Get File Analytics method. The return groups flag tells the VoiceBase system to include all group hits in the response. The Use Stemming tells the VoiceBase system if use should use stemming for keyword spotting or only return exact matches.

### Create Group

This method is using for creating group of keywords that assigned to owner of API key.

Parameter name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	createGroup	Parameter Is required.

name	Name of group	Characters	Parameter Is required. Max value: 256
keywords	Set of keywords	Characters	Parameter Is required. String of keywords separated by ",".

**Example** of the Create Group request:

```
http://beta.voicebase.com/services?version=1.1&apikey=51295918-1BAC-B7D2-C664-553E49DAB2FE&password=123456&action=creategroup&name=KGroup1&keywords=keyword1,keyword2,keyword3,keyword4,keyword5,keyword6,keyword7
```

Create Group will return the following data in the JSON payload

- requestStatus – SUCCESS or FAILURE
- statusMessage- Description of status
- groupName – name of the created group
- groupId – unique id of the group

**Example** Create Group response

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully","groupName":"KGroup1","groupId":3}
```

## List Groups

This method is using for getting list of groups of keywords that assigned to owner of API key.

Parameter name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	listGroups	Parameter Is required.

**Example** of List Groups request:

```
http://beta.voicebase.com/services?version=1.1&apikey=51295918-1BAC-B7D2-C664-553E49DAB2FE&password=123456&action=listgroups
```

Create Group will return the following data in the JSON payload

- requestStatus – SUCCESS or FAILURE
- statusMessage- Description of status
- groupName – name of the created group
- groups --- a list of group names owned by the this account

#### Example List Groups response

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully","groups":["KGroup1"]}
```

### Get Group

This method is using for getting of group of keywords that assigned to owner of API key by group name.

Parameter name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	getGroup	Parameter Is required.
name	Name of group	Characters	Parameter Is required. Max value: 256

#### Example of Get Group request:

```
http://beta.voicebase.com/services?version=1.1&apikey=51295918-1BAC-B7D2-C664-553E49DAB2FE&password=123456&action=getgroup&name=KGroup1
```

Get Group will return the following data in the JSON payload

- requestStatus – SUCCESS or FAILURE
- statusMessage- Description of status
- name – the group name
- keywords – a list of keywords in the group

#### Example Get Group response

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully","group":{"name":"KGroup1","keywords":["school","supplies","return","te"]}}
```

```
achers,promise,struggle"}}
```

## Delete Group

This method is using for delete of group of keywords that assigned to owner of API key by group name.

Parameter name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX- XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	deleteGroup	Parameter Is required.
name	Name of group	Characters	Parameter Is required. Max value: 256

**Example** of Delete Group request:

```
http://beta.voicebase.com/services?version=1.1&apikey=51295918-1BAC-B7D2-C664-553E49DAB2FE&password=123456&action=deletigroup&name=KGroup1
```

Delete Group will return the following data in the JSON payload

- requestStatus – SUCCESS or FAILURE
- statusMessage- Description of status

**Example** Delete Group response

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

## Update Group

This method is using for update group of keywords that assigned to owner of API key by group name.

Element	Value	Type/Format	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX- XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for	Characters	Parameter Is required.

	customer account		
action	Name of method	updateGroup	Parameter Is required.
name	Name of group	Characters	Parameter Is required. Max value: 256
keywords	Set of keywords	Characters	Parameter Is required. String of keywords separated by ",".

**Example** of updateGroup request:

```
http://beta.voicebase.com/services?version=1.1&apikey=51295918-1BAC-B7D2-C664-553E49DAB2FE&password=123456&action=updategroup&name=KGroup1&keywords=keyword1,keyword2,keyword3,keyword4,keyword5,keyword6,keyword7,keyword8
```

Update Group will return the following data in the JSON payload

- requestStatus – SUCCESS or FAILURE
- statusMessage- Description of status

**Example** Update Group response

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

## Folders

There are 6 folder: Create Folder, Delete Folder, Move Folder, Copy File to Folder and Delete File from Folder, List Folder. In addition you can specify a folder at upload time (see section describing upload methods) and you can filter searches by folder, (see search method).

Folders are specified in all methods as full path names with the / character as folder delimiter.

/firstfolder/subfolder

Top-level folder is by default the API accounts top-level folder. You can only use these methods with files either owned by or that have been shared with you. This means that for API accounts that are tenant organization accounts, you cannot manage or otherwise use a tenant member accounts folder. This also means that you cannot specify both an ownerId and a folder at upload time



## Create Folder

Use this method for creating new folders in an account. You must specify the full path name using / character for folder delimiters. The fields for this method are:

Parameter name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX- XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	createFolder	Parameter Is required.
folderName	Full path name of a folder in the API key account	Characters	Parameter Is required

### Notes:

- folder name with will no trailing “/” and subfolders will indicate the new folder is in an accounts top level folder
- All parent folders will be created, if they do not already exist
- API returns error code if the ”leaf” folder already exists

```
<?php
$version = "1.1";
$api_key = "A8429AAF-39CD-FFBC-4CAD-235E59FD52FE";
$password = "123456";
$action = "createFolder";
$folderName = "folder1/folder1.1/folder1.1.1";

$url = "https://api.voicebase.com/services";
$params = array(
    "version"    => $version,
    "apikey"     => $api_key,
    "password"   => $password,
    "action"     => $action,
    "folderName" => $folderName
);
```

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);
```

***Examples of response:***

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

```
{"requestStatus":"FAILURE","statusMessage":"You can not create a folder of the fourth level"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Request does not include required parameter: 'folderName'"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Folder with such name has already existed"}
```

```
{"requestStatus":"FAILURE","statusMessage":"You can no longer add folders at this level"}
```

## Delete Folder

The delete folder method can be used to remove a folder. Provide the full path of the folder to delete. There are two optional flags: force and recursive. If force is true the folder will be delete file content and the delete the folder. if recursive is true, the folder and all subfolders as well as the files will be deleted. It should be noted that deleting a file from a folder does not delete the content. File elements in folders are actually links (or shortcuts) to the actual file. Use the deleteFile method to delete the actual content. The fields for this method are:

Parameter Name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	deleteFolder	Parameter Is required.
folderName	Full path name of a folder in the API key account	Characters	Parameter Is required
forceFlag	If true, delete all files and then delete the folder specified. If false, the folder must be empty for this action to complete	boolean true   false	Optional parameter. Default is false
recursiveFlag	if true, delete all files and subfolders recursively. If false, any subfolder in the specified will cause this delete requests to fail	boolean true   false	Optional parameter  Default is false

### Notes:

- If both force and recursive flags are false, the folder must be empty for the action to be completed (they are false by default – when not specified)

- if force is true and recursive is false, the system will delete all file content and the delete the folder. But if any subfolders are in the specified folder are present, the action will not complete
- If you want to delete all files in the folder and subfolders recursively, you must set both recursive and force to true.

```
<?php
$version = "1.1";
$api_key = "A8429AAF-39CD-FFBC-4CAD-235E59FD52FE";
$password = "123456";
$action = "deleteFolder";
$folderName = "folder1/folder1.1/folder1.1.1";
$forceFlag = true;
$recursiveFlag = true;

$url = "https://api.voicebase.com/services";
$params = array(
    "version"      => $version,
    "apikey"       => $api_key,
    "password"     => $password,
    "action"       => $action,
    "folderName"   => $folderName,
    "forceFlag"    => $forceFlag,
    "recursiveFlag" => $recursiveFlag
);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);
```

### **Examples of response:**

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Invalid path name of a folder: folder  
'{FOLDER}' does not exist"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Request does not include required  
parameter: 'folderName'"}
```

```
{"requestStatus":"FAILURE","statusMessage":"The action can not be completed, because  
specified folder contains {COUNT} subfolder(s). To continue set up parameter  
'recursiveFlag' to true"}
```

```
{"requestStatus":"FAILURE","statusMessage":"The folder must be empty to complete this  
action. To continue set up parameter 'forceFlag' to true"}
```

```
{"requestStatus":"FAILURE","statusMessage":"The action can not be completed, because  
subfolder '{SUBFOLDER}' is not empty. To continue set up parameter 'forceFlag' to true or  
clear specified subfolder"}
```

## Move Folder

The move folder method can be used to move a folder to a new location or to rename a folder. Provide the full path of source (folder to be moved) and the full path of the destination (the new location and name). The fields for this method are:

Parameter Name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	moveFolder	Parameter Is required.
source	Full path name of the folder to be moved (an existing folder in the API key account)	Characters	Parameter Is required.
destination	Full path name of the location and name of the new folder	Characters	Parameter Is required.

### Notes:

- To move the folder and retain the same name, just make sure the name is same in destination parameter. So for example moving a folder called newStuff from top level account folder to a subfolder called archive (also in the top level account folder) use following params
  - o source=newStuff
  - o destination=archive/newStuff
- To simply rename the folder newStuff to betterNameForNewStuff and keep it in same location (to level account folder) use these parameters
  - o source=newStuff
  - o destination=betterNameForNewStuff

```

<?php
$version = "1.1";
$api_key = "A8429AAF-39CD-FFBC-4CAD-235E59FD52FE";
$password = "123456";
$action = "moveFolder";
$source = "test2/test1.3";
$destination = "test2/test1.33";

$params = array(
    "version"      => $version,
    "apikey"       => $api_key,
    "password"     => $password,
    "action"       => $action,
    "source"       => $source,
    "destination"  => $destination
);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);

```

### ***Examples of response:***

```

{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Invalid path name of a folder, which was
specified in the parameter 'source': folder '{FOLDER}' does not exist"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Request does not include required
parameter: 'source'"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Request does not include required
parameter: 'destination'"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Invalid destination path. You can not create
a folder of the fourth level"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Destination path contains folder that was
specified in source parameter"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Invalid path name of a folder, which was
specified in the parameter 'destination': folder '{FOLDER}' does not exist"}

```

```

{"requestStatus":"FAILURE","statusMessage":"The folder specified in destination is already
existed"}

```

```

{"requestStatus":"FAILURE","statusMessage":"The source folder contains subfolders. You
can not create a folder of the fourth level"}

```

### Copy File (Link) to Folder

Use this method to place a copy of a file into a folder. In order to take this action, you must have the mediald or the externalID and the full path of the destination folder. Your API key account must also have access to the file (either owner or the file has been shared with you). The fields for this method are:

Parameter Name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	copyFileToFolder	Parameter Is required.
mediald	Unique ID for the file. Returned by voicebase at upload time.	Characters	Parameter Is required (unless externalId is specified)
externalId	Unique ID for the file, provided by the user at upload time.	Characters	Parameter Is required (unless mediald is specified)
destination	Full path name of the location and name of the new folder	Characters	Parameter Is required.

#### Notes:

- Files that are in a folder are really a shortcut or a link to the actual file. So moving to a folder is moving the link.
- All folder paths are defined from the API keys top level folder.

#### Examples PHP code:

```
:
<?php
$version = "1.1";
$api_key = "A8429AAF-39CD-FFBC-4CAD-235E59FD52FE";
$password = "123456";
```

```

$action = "copyFileToFolder";
$destination = "folder1/folder1.1/folder1.1.1";
$mediaId = "4e09fd40cd9fc";
$externalId = "";

$url = "https://api.voicebase.com/services";
$params = array(
    "version"      => $version,
    "apikey"       => $api_key,
    "password"     => $password,
    "action"       => $action,
    "destination" => $destination,
    "mediaId"     => $mediaId,
    "externalId"  => $externalId
);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);

```

### ***Examples of response:***

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Use only one parameter: Media ID or External ID"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Specify Media ID or External ID"}
```

```
{"requestStatus":"FAILURE","statusMessage":"File should be transcribed to use this action"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Request does not include required parameter: 'destination'"}
```

```
{"requestStatus":"FAILURE","statusMessage":"No rights to view file"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Invalid path name of a folder, which was specified in the parameter 'destination': folder '{FOLDER}' does not exist"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Specified record is already presented in this folder"}
```



### Delete File (link) from a Folder

Use this method to delete a file from a folder. To use this method you need the mediald or the externalID of the file that will be deleted and the folder that it will be deleted from. The fields for this method are:

Parameter Name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	copyFileToFolder	Parameter Is required.
mediald	Unique ID for the file. Returned by voicebase at upload time.	Characters	Parameter Is required (unless externalId is specified)
externalId	Unique ID for the file, provided by the user at upload time.	Characters	Parameter Is required (unless mediald is specified)
destination	Full path name of the location and name of the new folder	Characters	Parameter Is required.

#### Notes:

- Files that are in a folder are really a shortcut or a link to the actual file. So moving to a folder is moving the link.
- All folder paths are defined from the API keys account top level folder.

#### Example of request:

```
<?php
$version = "1.1";
$api_key = "A8429AAF-39CD-FFBC-4CAD-235E59FD52FE";
$password = "123456";
$action = "deleteFileFromFolder";
$destination = "folder1/folder1.1/folder1.1.1";
$mediald = "4e09fd40cd9fc";
```

```

$externalId = "";

$url = "https://api.voicebase.com/services";
$params = array(
    "version"      => $version,
    "apikey"       => $api_key,
    "password"     => $password,
    "action"       => $action,
    "destination"  => $destination,
    "mediaId"     => $mediaId,
    "externalId"   => $externalId
);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);

```

### ***Examples of response:***

```

{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Use only one parameter: Media ID or External ID"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Specify Media ID or External ID"}

```

```

{"requestStatus":"FAILURE","statusMessage":"File should be transcribed to use this action"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Request does not include required parameter: 'destination'"}

```

```

{"requestStatus":"FAILURE","statusMessage":"No rights to view file"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Invalid path name of a folder, which was specified in the parameter 'destination': folder '{FOLDER}' does not exist"}

```

```

{"requestStatus":"FAILURE","statusMessage":"Specified record is not presented in this folder"}

```

## List Folder

Use this method to list the content of a folder. The fields for this method are:

Parameter Name	Required	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	listFolder	Parameter Is required.
folder	Full path name of the location of the folder	Characters	Parameter Is required.

### Notes:

Output of the list folder method contains the folderContents json element which is a json array of the folder contents. Each piece of content is an element in the list and either represents a file or another folder. Each element contains metadata in the form of name value pairs (hash map or dictionary). So the output is json array of dictionaries. For example here is example output with 2 elements a file and a subfolder.

```
[ {"name": "fileName1",
  "type": "file",
  "mediaId": "xxxxxx",
  "externalID": "yyyyy",
  "recordedDate": "10/23/2014 18:24:26"},
  {"name": "folderName1",
  "type": "folder",
  "ownerId": "x@a.b.com",
  "shared": true,
  "fpn": "/x/y/x"}, ...
]
```

1. All elements have the property called type that is a string and can either be value "file" or value "folder"
2. All files have 4 additional string properties: name, mediaId, externalID, recordedDate and owner

3. All folders have 4 additional properties: name, ownerId, shared and full path name(fpn). All of these properties are strings except shared which is a boolean and indicates if the folder is a shared folder (shared in the tenant organization).

### ***Example of request:***

```
<?php
$version = "1.1";
$api_key = "A8429AAF-39CD-FFBC-4CAD-235E59FD52FE";
$password = "123456";
$action = "listFolder";
$folderName = "test2";

$params = array(
    "version"    => $version,
    "apikey"     => $api_key,
    "password"   => $password,
    "action"     => $action,
    "folderName" => $folderName
);
$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 0);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);
curl_close($ch);
```

### ***Examples of response:***

```
{
  "requestStatus":"SUCCESS",
  "statusMessage":"The request was processed successfully",
  "folderContents":[
    {
      "type":"file",
      "name":"Iphone4 recording Original8kh",
      "mediaId":"test1",
      "externalId":"123456",
      "recordedDate":"06V01V2011 11:46"
    },
    {
      "type":"file",
      "name":"13101_podcast111",
      "mediaId":"4e09fd40ce99c",
      "externalId":"3007",
      "recordedDate":"03V09V2011 14:23"
    },
    {
      "type":"folder",
      "name":"test1.33",
      "fpn":"test2V/test1.33"
    }
  ],
}
```

```
{
  "type":"folder",
  "name":"test1.3.1",
  "fpn":"test2\test1.33\test1.3.1"
}
]
```

```
{"requestStatus":"FAILURE","statusMessage":"Invalid path name of a folder, which was specified in the parameter 'folderName': folder '{FOLDER}' does not exist"}
```

```
{"requestStatus":"FAILURE","statusMessage":"Request does not include required parameter: 'folderName'"}
}
```

## Share File

Use the share method to share file with other users or make the file public. You can use this method to provide access to a file to a set of named users.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by voicebase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	“share”	Indicates this is a share request.
mediaID	Yes	The media Id is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and search method
externalId	If no media ID	The file Id	Unique ID assigned by the client at upload

			time.
email	Yes, if public is false.	User's email (voicebase account) with whom the file will be shared. can be an array of emails.	The email (voicebase account) to share file.
canShare	No	Flag indicating user's rights on file (TRUE or FALSE)	Flag indicating that user will be able to share file with other users.
canEdit	No	Flag indicating user's rights on file (TRUE or FALSE)	Flag indicating that user will be able to edit file information.
sendMessage	No	Flag indicating send notification of share to specified user or not (TRUE or FALSE).	If option is set, specified user will get email notification about share.
message	No	Content of the message, if send message is true.	If previous option ("sendMessage") is set, the system will added your message to user's email message.
public	No	TRUE / FALSE	Set option to TRUE to make file public.

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL:

<https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=share&mediaid=4e09fd40dcc28&email=user-email&canShare=false&canEdit=true&sendMessage=true&message=test+message>

```
{"requestStatus":"SUCCESS",  
"statusMessage":"The request was processed successfully",  
"response": true}
```

#### *PHP Example using curl*

```
<?php  
$version = "1.1";  
$apikey = "your-API-Key";  
$password = "password";  
$url = "https://api.voicebase.com/services";  
$action = "share";  
$media_id = 11111;  
$email = 'user@voicebase.com';  
$can_share = true;  
$can_edit = true;  
$send_message = true;  
$message = 'test message';
```

```
$url = "https://api.voicebase.com/services";  
$url .= "?version=".$version;  
$url .= "&apikey=".$apikey;  
$url .= "&password=".$password;  
$url .= "&action=".$action;  
$url .= "&mediaid=".$media_id;  
$url .= "&email=".urlencode($email);  
$url .= "&canShare=".$can_share;  
$url .= "&canEdit=".$can_edit;  
$url .= "&sendMessage=".$send_message;  
$url .= "&message=".urlencode($message);
```

```
$ch = curl_init();  
curl_setopt($ch, CURLOPT_VERBOSE, 1);  
curl_setopt($ch, CURLOPT_TIMEOUT, 300);  
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);  
curl_setopt($ch, CURLOPT_URL, $url);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
$response = curl_exec($ch);  
curl_close($ch);  
echo $response;  
?>
```

## Get File Shares List

This method is using for getting of list users who have been given share access to the file.

Element	Value	Type/Format	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX- XXXX-XXXXXXXXXXXX	Parameter Is required.
PASSWORD	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	getFileSharesList	Parameter Is required.
mediald	Unique file ID	13 characters	Parameter is optional. You can use only one parameter 'mediald' or 'externalId'.
externalId	Unique file ID	Characters	Parameter is optional. You can use only one parameter 'mediald' or 'externalId'.

**Example** of getFileSharesList request:

```
http://redesign.beta.voicebase.com/services?version=1.1&apikey=51295918-1BAC-B7D2-C664-553E49DAB2FE&password=123456&action=getfileshareslist&mediald=4e09fd40da515
```

The following table lists the output returned from the VoiceBase for an API call

- requestStatus
- statusMessage
- sharesList – list of all users who have share access to this file

**Example** Get File Shares List response

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully","response":[{"userId":"422","userName":"Marinok Fadeeva","shareRights":"0","adminRights":"1"}, {"userId":"146","userName":"Mstislav Begunov","shareRights":"1","adminRights":"0"}, {"userId":"435","userName":"Eugene Voytenko","shareRights":"1","adminRights":"1"}]}
```



## Create Account

Use the create Account method to create an account for a named user with an email address as the account name. The user will receive an email from voicebase with instructions for activating the account.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by voicebase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	"createAccount"	Indicates this is a create account request.
account	Yes	The account ID to be created.	All voicebase account ids are email addresses.
accountPassword	Yes	The default password for this account	The users default password
firstName	Yes	First name user	Users first name
lastName	Yes	Last name of user	Users Last name

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL:

```
https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=createAccount&&account=wtell@someemail.com&password=secret2&firstName=William&lastName=Tell
```

```
{"requestStatus":"SUCCESS",  
"statusMessage":"The request was processed successfully",  
"response": true}
```

#### *PHP Example using curl*

```
<?php  
$version = "1.1";  
$apikey = "your-API-Key";  
$password = "password";  
$url = "https://api.voicebase.com/services";  
$action = "createAccount";  
$account = 'user@voicebase.com';  
$password = "secret2";  
$firstName = "William";  
$lastName = "Tell";  
  
$url = "https://api.voicebase.com/services";  
$url .= "?version=" . $version;  
$url .= "&apikey=" . $apikey;  
$url .= "&password=" . $password;  
$url .= "&action=" . $action;  
$url .= "&account=" . urlencode($account);  
$url .= "&password=" . $password;  
$url .= "&firstName=" . $firstName;  
$url .= "&lastName=" . $lastName;  
  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_VERBOSE, 1);  
curl_setopt($ch, CURLOPT_TIMEOUT, 300);  
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);  
curl_setopt($ch, CURLOPT_URL, $url);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
$response = curl_exec($ch);  
curl_close($ch);  
echo $response;  
?>
```

## Add Comment

Use the Add Comment method to add a comment/annotation to the files timeline. The comment will be displayed in the player.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	“AddComment”	Indicates this is a AddComment request.
mediaID	If no external ID	The media Id is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and search method
timestamp	yes	Time in seconds from beginning of the.	A comment has a time position.
Comment	Yes	The comment itself	Text of the comment
commentId	No	Unique id of a previous comment. Should be used if you want to reply to a comment	Optional. if not included, this is a new comment (not a reply to a comment).

The HTTP response will contain a requestStatus Code and a statusMessage. The requestStatus code indicates if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=addComment&mediaid=4e0dca1cb1bb8&comment=Check%20it%20out!>

```
{"requestStatus":"SUCCESS",  
"statusMessage":"The request was processed successfully"}
```

### Get Comments

This method is using for getting of comments list for the media file. It is an list of threads. Each thread is positioned at a time within the recording.

Parameter Name	Value	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	addComment	Parameter Is required.
mediaId	Unique file ID	13 characters	Parameter is optional. You can use only one parameter 'mediaId' or 'externalId'.
externalId	Unique file ID	Characters	Parameter is optional. You can use only one parameter 'mediaId' or 'externalId'.

**Example** of Get Comments request:

```
http://redesign.beta.voicebase.com/services?version=1.1&apikey=8682BF70-265B-C159-658B-6E11EA7486FB&password=123456&action=getComments&mediaId=52eb7655045f8
```

Results of Get Comments method

- requestStatus – status of request SUCCESS/FAILURE
- statusMessage – description of result status
- response -- array

- threads – list of threads
  - id – id for this thread
  - timestamp – time position in recording for this thread
  - comments -- list of comments on this thread
    - id – id for this comment
    - content – the comment itself
    - level --
    - position --
    - username – name of user who added this comment
    - addDate – date/time in ISO 8601 format the comment was added

#### Example Get Comments response

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "response": {
    "threads": [
      {
        "id": "640",
        "timeStamp": "0",
        "comments": [
          {
            "id": "1966",
            "content": "Test comment 1 for thread 1",
            "level": "1",
            "position": "1",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:52:57Z"
          },
          {
            "id": "1967",
            "content": "Test reply comment 1 to comment 1 in thread 1",
            "level": "2",
            "position": "2",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:53:19Z"
          },
          {
            "id": "1971",
            "content": "Test reply comment 2 to comment 1 in thread 1",
            "level": "2",
            "position": "3",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:54:44Z"
          },
          {
            "id": "1972",
            "content": "Test reply comment 1 to replied comment 2 in thread 1",
            "level": "3",
            "position": "4",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:55:16Z"
          },
          {
            "id": "1968",
            "content": "Test comment 2 for thread 1",
            "level": "1",
            "position": "5",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:53:40Z"
          }
        ]
      },
      {
        "id": "641",
        "timeStamp": "303",
        "comments": [
          {
            "id": "1969",
            "content": "Test comment 1 for thread 2",
            "level": "1",
            "position": "1",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:54:05Z"
          },
          {
            "id": "1970",
            "content": "Test comment 2 for thread 2",
            "level": "1",
            "position": "2",
            "userName": "Eugene Voytenko",
            "addDate": "2014-02-06T15:54:22Z"
          }
        ]
      }
    ]
  }
}
```

#### Update Transcript

This method is used for replacing the existing automated machine transcript of a recording with a user-supplied transcript. The user-supplied transcript must be included in as an attachment to the post of this method.

Note that you can also include a user-supplied transcript at upload time (see upload method).

Parameter name	Value	Description	Notes
version	1.1	Characters	Parameter Is required.
apikey	The valid API Key Assigned to customer account	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Parameter Is required.
password	The valid Password for customer account	Characters	Parameter Is required.
action	Name of method	updateTranscript	Parameter Is required.
mediald	Unique ID for the file. Returned by voicebase at upload time.	Characters	Parameter Is required (unless externalId is specified)
externalId	Unique ID for the file, provided by the user at upload time.	Characters	Parameter Is required (unless mediald is specified)
transcript	Plain text attachment	Characters	An attachment containing the user supplied human transcript in plain text.

### *Transcript Format*

- Plain text in UTF8
- Transcriber notes are in square brackets. No other use of square brackets is allowed. Examples: [INAUDIBLE], [DOOR SLAM]
- Speaker turns should always be on a new line with up to 6 words followed by a colon :
- Optional time sync tags in square brackets interspersed inside the transcript [HH:MM:SS] format. This is the time since the beginning of the recording. Time sync tags are recommended in cases where there are long silences within a recording or bad recording quality or noisy recordings in order to improve word alignment performance. The time will be applied to the start of the next word following the tag in the transcript.
- Include white space for formatting.

Example:

```
[BLANK_AUDIO]
[00:01:45]
DEEP THOUGHT: Can I help you?
FOOK: Deep Thought. Do you have...
DEEP THOUGHT: An answer for you? Yes. I have.
FOOK: There really is an answer?
DEEP THOUGHT: Yes. There really is one.
FOOK: [almost strangled with emotion] Oh!
LUNKWILL: Can you tell us what it is?
DEEP THOUGHT: Yes. Though I don't think you're going to like it.
FOOK: Doesn't matter! We must know it!
```

DEEP THOUGHT: You're really not going to like it.

FOOK: Tell us!

DEEP THOUGHT: Alright. The answer to the ultimate question ...

LUNKWILL: Yes ...

DEEP THOUGHT: ... of Life, the Universe, and Everything ...

FOOK: Yes!

DEEP THOUGHT: ... is ...

CROWD: Yes ...!

[00:02:01]

DEEP THOUGHT: [longest pause yet] Forty two. I checked it thoroughly. It would have been simpler, of course, to have known what the actual question was.

### Upgrade Transcript

Use the Upgrade File method if you want a human transcript for a file that's already processed as a machine transcript. Human transcript charges apply.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	"UpgradeTranscript"	Indicates this is a Upgrade File (to human transcript) request.
mediaID	If no external ID	The media Id is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and search method
externalID	If no media ID	The file Id	Unique ID assigned by

			the client upload time.
notesToTranscriber	No	Plain text, notes to the transcriber.	Free form text.

The HTTP response will contain a requestStatus Code and a statusMessage. The requestStatus code indicates if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=upgradeTranscript&mediaid=4e0dca1cb1bb8>

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully"
}
```

### Update Media File

Use the Update Media File method to change any metadata for the file.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	"UpdateMediaFile"	Indicates this is a UpdateMediaFile request.
mediaID	If no external ID	The media Id is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and



			search method
externalID	If no media ID	The file Id	Unique ID assigned by the client upload time.
title	No	Title of file	Can change the value in system
desc	No	Description of file	Can change the value in system
sourceUrl	No	Source url (for remote streaming)	Can change the value in system
collection	No	Collection name	Can change the value in system
recordedDate	No	Recorded date	Can change the value in system Date format is YYYY-mm-dd HH:mm:ss. Stored as a DATETIME field type. Examples: 2014/05/08 2014/5/07 12:34
imageUrl	No	Image url shown on detail page	Can change the value in system
uploadCollection	no	Upload collection (can be used for polling a set of files)	Can change the value in system
searchHitUrl	No	Override the search hit url with this url so not returning the voceibase player url, instead the url of your choice	This is use for building a web app using the web sdk and player plugin. If you want to use your own player

			pages.
callbackBacks		Array of (uploadedCallback, indexedCallback machineReadyCallback humanReadyCallback)	Can change the value in system

The HTTP response will contain a requestStatus Code and a statusMessage. The requestStatus code indicates if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=updateMediaFile&mediaid=4e0dca1cb1bb8>

```
{"requestStatus":"SUCCESS",
"statusMessage":"The request was processed successfully"}
```

### Get Transcript

Use the Get Transcript method to retrieve the files transcript in pdf, rtf or srt format.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	“GetTranscript”	Indicates this is a GetTranscript request.
mediaID	If no external ID	The media Id is a unique	Unique ID that was

		identifier for the media file.	returned from the uploadFile, list and search method
externalID	If no media ID	The file Id	Unique ID assigned by the client upload time.
format	Yes	Format of the transcript to be returned. Valid formats are: Pdf, rtf or srt	Json can be retrieved with the getKeywords method

The HTTP response will contain a requestStatus Code and a statusMessage. The requestStatus code indicates if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getTranscript&mediaId=4e0dca1cb1bb8&format=txt>

Success responses:

TEXT

```
{ "requestStatus":"SUCCESS",
  "statusMessage":"The request was processed successfully",
  "transcript":"california regulators have approved a record number of large skill solar energy projects is ... will be up for the next ninety days in california or part time are",
  "transcriptType":"machine"
}
```

JSON

```
{
  "requestStatus":"SUCCESS",
  "statusMessage":"The request was processed successfully",
  "transcript":[{"p":1,"s":10,"e":712,"c":0.67,"w":"california"}, {"p":2,"s":712,"e":1323,"c":0.17,"w":"regulators"}],
  "transcriptType":"machine"
}
```

SRT

```
{
  "requestStatus" => "SUCCESS",
  "statusMessage" => "The request was processed successfully",
  "transcript" =>
1
00:00:00,01 --> 00:00:06,86
california regulators have approved a record number of large skill solar energy projects is here
```

2

00:00:06,86 --> 00:00:13,61

many are still a long ways away from levels in the ground in part because of environmental and now

```
"transcriptType" => machine
}
```

PDF

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "transcript": "http://api.voicebase.com/voice_file/downloads/2209/pdf",
  "transcriptType": "machine"
}
```

### Get AutoNotes Html

Use the Get Autonotes HTML methods to retrieve html text for the auonotes page (including topic groupings and keyword links). This could be used in emails or for cloud storage.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	"GetAutoNotesHtml"	Indicates this is a GetAutoNotesHtml request.
mediaID	If no external ID	The media Id is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and search method

The HTTP response will contain a requestStatus Code and a statusMessage. The requestStatus code indicates if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getFileStatus&mediaid=4e0dca1cb1bb8>

```
{"requestStatus": "SUCCESS",  
  "statusMessage": "The request was processed successfully",  
  "autonotes": "<html/>"}
```

### Get AutoNotes URL

Use the Get Autonotes URL methods to retrieve a to the autonotes player page. (Including topic groupings and keyword links). This could be used in your web app to direct users to the branded player page. The urls for each keyword is also provided so you could setup links to particular keywords too.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1
apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	"GetAutoNotesHtml"	Indicates this is a GetAutoNotesHtml request.
mediaID	If no external ID	The media Id is a unique identifier for the media file.	Unique ID that was returned from the uploadFile, list and search method

externalId	If no media ID	The file Id	Unique ID assigned by the client at upload time.
rssGuid			If uploaded with podcast feed, this is an alternative unique identifier from the feed
podCastUrl			If uploaded with podcast feed, this is an alternative unique identifier from the feed

The HTTP response will contain a requestStatus Code and a statusMessage. The requestStatus code indicates if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: <https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=getAutonotesUrl&mediaid=4e0dca1cb1bb8>

Response:

```
{
  "requestStatus": "SUCCESS",
  "statusMessage": "The request was processed successfully",
  "keywordUrls": [
    {
      "keyword": "recording",
      "url": "http://theurltothiskeyword"
    },
    {
      "keyword": "grapes",
      "url": "http://theurltothiskeyword"
    }
  ],
  "fileUrl": "http://theurltothefile"
}
```

### Set Spot words

Use the Set Spot words method to specify the list of keywords to use in keyword spotting mode. The Spot words are the words searched for in the file.

Parameter name	Required	Description	Notes
version	Yes	API version	Current version is 1.1

apikey	Yes	API key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	“SetSpotwords”	
spotWords		List of spotwords to be used in keyword spotting	

### Get Spot words

Use the Get Spot words method to specify the list of keywords to use in keyword spotting mode. The Spot words are the words searched for in the file.

Parameter name	Required	Description	Notes
version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	Value: AddKeyword	action=AddKeyword

### Add keyword

You can use the Add Keyword method to add a new keyword to a file. You can enter one or more keywords. If at least one word doesn't exist in keywords list, the response will be successful.

Parameter name	Required	Description	Notes
version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	Value: DeleteKeyword	action=DeleteKeyword
keyword	Yes	One or more keywords	

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL:

<https://api.voicebase.com/services?version=1.1&apikey=your-api-key&password=secret&action=addKeyword&mediaID=1234567&kkeyword=fire+and+ice>

Example Responses:

```
{"requestStatus":"FAILURE","statusMessage":"You haven't access to add keyword."}
{"requestStatus":"FAILURE","statusMessage":"Specified keyword already exists."}
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```



### Delete Keyword

You can use the delete Keyword method to delete an existing keyword from a file. You specify the topic from which to remove the keyword. If no topic is entered, then the specified keyword is removed from the “all topic” list for the file.

Parameter name	Required	Description	Notes
version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	Value: DeleteKeyword	action=DeleteKeyword
Keyword	Yes	One or more keywords	
categoryId	No	Remove the keyword from the topic specified here. (the term category and topic is used interchangeably)	IF no categoryId, then keyword will be removed from the “ALL Topic”

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL: TBS

```
{"requestStatus":"FAILURE","statusMessage":"You haven't access to edit keywords."}
```

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

### Move Keyword

You can use the move Keyword method reposition an existing keyword in a file. You specify the topic and the keyword and the mode. Mode can be up, down or first. Up moves up one position, down moves the keyword down one position and first moved the keyword to the top of the list. If no topic is entered, then the specified keyword is removed from the “all topic” list for the file

Parameter name	Required	Description	Notes
version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	Value: SetSpotwords	action=SetSpotwords
Keyword	Yes	One or more keywords	
categoryID	No	Remove the keyword from the topic specified here. (the term category and topic is used interchangeably)	IF no categoryId, then keyword will be removed from the “ALL Topic”
mode	Yes	Set to: Up, down or first	Move up or down one position or move to top of list.

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL:

TBS

ExampleResponses:

```
{"requestStatus":"FAILURE","statusMessage":"Parameter 'mode' is invalid. It should be either 'up' or 'down' or 'first'."}
```

```
{"requestStatus":"FAILURE","statusMessage":"You haven't access to edit keywords."}
```

```
{"requestStatus":"SUCCESS","statusMessage":"The request was processed successfully"}
```

### GetToken

The getToken method is used to get a token that can be used in method calls in place of the password. This is useful when using the API in Javascript code inside the browser. You can control which file the token can be used for and you can specify a timeout for the token.

Parameter name	Required	Description	Notes
version	Yes	API Version	Current version is 1.1
apikey	Yes	API Key	The API key is provided by VoiceBase and is associated with a user account
password	Yes	API Password	API Password. This is the users normal account password
action	Yes	"GetToken"	
timeout	Yes	One or more keywords	

The HTTP response will contain a requestStatus Code indicating if this request was completed successfully. Request Status can be SUCCESS or FAILURE.

Example URL:

## Utterance Detectors (Beta, contact VoiceBase Support)

VoiceBase has a predefined set of utterance detectors that you can specify at upload time using the utterances parameter of the upload method. The results of the utterance detector will be returned as part of the getFileAnalytics method. Results of an utterance detector is a list of utterances. An utterance has a start time and an end time.

At present there is one utterance detector app called CHData. It identifies the location where Credit Card Data was spoken. This can be used to scrub the audio.

## SIP, Real-time and Keyword Spotting (Beta, contact VoiceBase Support)

### *Realtime*

Realtime processing can be achieved with the Post version of the upload method using the optional parameters

1. Callid
2. Realtime callback Url
3. finalFlag
4. transcriptNotify
5. Keyword Notify

And if you are doing keyword spotting, you can use the setSpotwords and getSpotwords methods set the keywords to be searched for – and to get a list of what is currently set.

You should split your audio up and send it to voicebase using the upload method in chunks. Make sure all chunks have the same callid and that the final chunk has the finalFlag set to true.

If you want to be notified via a callback with the transcript for each chunk, set the flag (and be sure to include the callbackUrl). If you want the keyword notifications set that flag (and also need the url). Don't forget to specify the spotwords you are looking for.

```
curl -X POST -F version=1.1 -F apikey=<your-key> -F password=<your-password> -F
action=uploadmedia -F file=@your-file.mp3 -F transcriptType=machine -F public=false -F
title=MyTestFile -F callid 12345-unique -rtCallbackUrl http://theurl/handler -
transcriptNotify=true https://api.voicebase.com/services
```

### *SIP/RTP Recording*

Contact Voicebase to provision a sip url that you can use for your recordings. The sipURL will be of the form [sip:yourname@voicebase.com](mailto:sip:yourname@voicebase.com)

You can then send invite requests to the sip url. We will then process the audio as it arrives. Essentially we will use the real-time api for you with 30 second chunk sizes. If you use the sip URL input, you can set the notify flags and urls using the TBS method.

### Callbacks

Callback data will be returned in a post with the following parameters

Parameter name	Required	Description	Notes
mediaID	Yes	Unique mediaID	String
chunkID	Yes	ID of this chunk	Typically an integer sequence
startTime	Yes	startTime of this chunk	
finalFlag	Yes	Boolean flag indicating if this the last chunk	
transcript	No	JSON format of the transcript for this chunk	
keywords	No	JSON list of keywords hits. An array of keyword/hitlist pairs.	Example: [{"keyword":"friday","hits":[{"p":12,"s":12890,"e":13890,"c":0.828,"w":"Friday"}]},{"keyword":"Sunday","hits":[{"p":8,"s":8270,"e":9310,"c":0.958,"w":"Sunday"}]},{"p":10,"s":10570,"e":12290,"c":0.754,"w":"Sunday"}]}

## Partner Integration URLs and Branding

Once you have uploaded content into VoiceBase, you can then make it available for your customers using the partner integration urls and branding feature. The basic idea is that you form VoiceBase search and explore urls on your web site which will direct your users to the VoiceBase site so they can access content that has already been loaded and indexed.

You can use your own branding by specifying the html to be used at the top and bottom of your audio search results and details pages. This will provide a seamless experience for your users when switching between sites. The html is typically the same html you are already using on your existing web page.

One typically usage is to add a simple audio search box in the header of your web page. This can be part of an existing search control or it could be a separate audio search control in the header. The search

control would direct the user to a VoiceBase search url that includes the search terms that the user specified. In response, VoiceBase servers will display a search results page with your header and footer. Users can select one of the files in the result set and play the media, comment on it, see the keywords and do other searches on their audio and video. They can also easily navigate back to your web site since the header links are your links.

You can also add an explore url that will bring users directly to a particular recording. The explore urls require the mediaId that was returned from the upload method (or you can use an externalId if one was specified at upload time).

There are basically 3 steps to using partner integration and branding.

1. Contact VoiceBase and request partner integration to be enabled on your account.
2. Once partner integration is enabled, login to your VoiceBase account and go to the partner integration page under the account settings. You can use this page to design the look and feel of your search results page so that it matches your existing web site. Typically you will add the header navigation on the top and footer on the bottom. You will find 4 text boxes where you can specify this html. They are:
  - a. Html header (can be used for css and js includes)
  - b. Top content
  - c. Bottom content
3. Add an audio search control to you web site

## Partner Integration URLs

The search url takes the following form ( this example search for "social network")  
<https://www.voicebase.com/partner/search/SEARCH+REQUEST/pid=yourPartnerId>

Or this form

<https://www.voicebase.com/partner/search?q=global+warming&pid=a-partner-id>

The explore url takes the following form. This example will take you to a file with externalId=9871bc using the partner branding specified and logging into the account specified by username/password.

<https://www.voicebase.com/details/partnerId=yourPartnerId&userid=username@VoiceBase.com&password=password&externalId=987abc>

<https://www.voicebase.com/details/partnerId=yourPartnerId&userid=username@voicebase.com&password=password&externalId=unique-xid>

## Authentication

Contact VoiceBase support.

## Web SDK (for embedded player and search widgets)

### VoiceBase JWPlayer Plugin

The jwplayer plugin extends jwplayer to include

- audio search
- keyword and topic hyperlinks

- Synchronized transcripts

You can add these player extensions by taking these 3 steps

- Indexing the media file with voicebase API. the API will return a unique identifier for the file.
- Using JWPlayer on your web page in the typical way (See long tail video web site for details on JWPlayer).
- Include the VoiceBase JWPlayer plugin configuring it to reference the indexed file using the unique identifier you obtained in step 1.

The VB plugin is written in JavaScript and does ajax style api calls to the voicebase server to retrieve time indexed keywords and topics, to do adhoc search with time synced results and to show time synced transcript.

There are 2 modes of operation.

- Pure Javascript mode.
- Server side SDK, to generate the transcript and keyword HTML on the server. This will be better for SEO.

## Getting Started

In order to use the plugin first

- Get Jwplayer ( <http://www.longtailvideo.com/jw-player/>)
- get a voicebase account and API key (Contact us at [support@voicebase.com](mailto:support@voicebase.com))
- Javascript requirements
  - o jquery >v1.9
  - o bootstrap-tooltip.js v2.3.2

Then for each file you want to play with search, keyword links and synchronized transcripts:

1. Upload the file using the voicebase API upload method. The response will include a unique mediaId. Make a note of this string, you will need it later. The upload method uses a simple http base api that can accept the media file as an attachment in a post, or a url in a get command. The following is a get example (using a link to the media file) and a post example using curl command that uploads the file as an attachment.

```
https://api.voicebase.com/services?version=1.1&apikey=<your-key>&password=yourpassword&action=uploadMediaGet&mediaURL=<url-to-file>&transcriptType=machine&public=false&title=MyTestFile
```

```
curl -X POST -F version=1.1 -F apikey=<your-key> -F password=<your-password> -F action=uploadmedia -F file=@your-file.mp3 -F transcriptType=machine -F public=false -F title=MyTestFile https://api.voicebase.com/services
```

2. Build an HTML page with jwplayer and the vb plugin. It is assumed that you are familiar with JWPlayer. if you are not, go to jwplayer site and check out the documentation.

```

<html><head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <!-- Include jquery. This is required by the plugin -->
  <script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>

  <!-- include the jwplayer js in the normal way -->
  <script type="text/javascript" src=jwplayer/jwplayer.js></script>
  <script>jwplayer.key="ADD YOUR JWPLAYER KEY HERE"</script>

  <!-- tooltip js is required by player plugin -->
  <script src="http://demo.vocibasejwplayer.dev3.sibers.com/plugin3/tooltip.js"></script>
  <link rel="stylesheet" type="text/css" href="http://demo.vocibasejwplayer.dev3.sibers.
com/plugin3/tooltip.css">

  <!-- player plugin style sheet -->
  <link rel="stylesheet" type="text/css" href="http://demo.vocibasejwplayer.dev3.sibers.
com/plugin3/vb_styles.css" />

</head>
<body>
  <!-- Note that this is typical JWPlayer usage with typical file parameter to specify media, height and width -->
  <div id='my'></div>
  <script type='text/javascript'>
    jwplayer('my').setup({
      file: 'ADD YOUR MEDIA FILE LOCATION HERE',
      width: '600',
      height: '200',
      <!-- specify the player plugin here, with some basic parameters -->
      plugins: {
        "http://demo.vocibasejwplayer.dev3.sibers.com/plugin3/vbplayer.js": {
          mediaID: 'ADD_YOUR_MEDIAID_HERE', // The mediaID of file you uploaded earlier
          apiKey: 'ADD_YOUR_KEY_HERE', // your key
          password: 'ADD_YOUR_PW_HERE', // your password (NOTE: Security tokens to be added
soon for security)
          keywordsHeight: 150, // height of keyword section (width will be same as player)
          transcriptHeight: 150, // height of transcript section (width will be same as player)
          apiURL: 'http://www.vocibase.com/services',
          humanOnly: false
        }
      }
    });
  </script>
</body>
</html>
</body></html>

```

Figure 1 -- Javascript example of jwplayer plugin



```

<?php
/* Include VoiceBase Class library */
include_once 'vbClass.php';
/*****
    Initialize the vbclass with credentials and mediald (and api url optional)
    vbClass('apikeyapikey', "password", "mediald", "apiUrl")
    *****/
$vb = new vbClass('YOUR-KEY-HERE', "YOUR-PW-HERE", "YOUR-MEDIA-ID");?>

<html>
<head>
<title>JW Plugin</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<!-- Include jquery. This is required by the plugin -->
<script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>

<!-- include the jwplayer js in the normal way -->
<script type="text/javascript" src=jwplayer/jwplayer.js></script>
<script>jwplayer.key="ADD YOUR JWPLAYER KEY HERE"</script>

<!-- tooltip js is required by player plugin -->
<script src="http://demo.voicsebasejwplayer.dev3.sibers.com/plugin3/tooltip.js"></script>
<link rel="stylesheet" type="text/css" href="http://demo.voicsebasejwplayer.dev3.sibers.com/plugin3/tooltip.css">

<!-- player plugin style sheet -->
<link rel="stylesheet" type="text/css"
href="http://demo.voicsebasejwplayer.dev3.sibers.com/plugin3/vb_styles.css" />

</head>
<body>
<div class="content">
<div id="my"></div>
<script type="text/javascript">
jwplayer('my').setup({

    file: 'http://ec2-50-19-181-137.compute-1.amazonaws.com/20110422_atc_07.mp3',
    width: '640',
    height: '480',
    plugins: {
        "http://demo.voicsebasejwplayer.dev3.sibers.com/plugin3/vbplayer.js": {
            medialD: 'ADD_YOUR_MEDIAID_HERE', // The mediald of file you uploaded earlier
            apikeyapikey: 'ADD_YOUR_KEY_HERE', // your key
            password: 'ADD_YOUR_PW_HERE', // your password (NOTE: Security tokens to be added soon for
security)
            serverMode: true //this is required in server mode
        }
    }
});
</script>
<!-- place Keywords here style=2, width=600 and height=200 -->
<?php echo $vb->getKeywords(2, 600, 100); ?>
<!-- place Transcript here width=600, height=280 and 5 second transcript highlighting-->
<?php echo $vb->getTranscript(600, 280, 5); ?>
</div>
</body>
</html>

```

Figure 2 -- PHP example of JWpalyer plugin

### *JWPlayer plugin Parameters*

There are a set of parameters that you can use to customize the player for you needs.

Parameter name	Required	Description	Notes
mediaID	yes	The unique mediaID of the file. This identifier is returned at upload time. Also returned as file identifier in search command.	
apikey	Yes	Your API Key	
password	Yes	Your secret password	
serverMode	No	Boolean. Set to true if using PHP app (not using pure javascript mode)	default false
apiURL	No	default <a href="https://api.voicebase.com/services">https://api.voicebase.com/services</a>	
contextMenu	No	allow context menu copy/share	default true
topicHover	No	allow highlighting topics and showing markers on the timeline	default false
transcriptHighlight	No	Length of synchronized text in transcript highlight time (in seconds) indicating the audio being played. Minimum 5. Default is 5. 0 indicates no highlighting.	
keywordsDivID	No	id of the alternative block for showing the keywords.	
keywordsHeight	No	Height of keywords block (in pixels) default 170. If 0, the keywords are hidden.	
keywordsWidth	No	the width of keywords block (in pixels) default is to make same size as player on the whole block's	Minimum is 220
transcriptDivID	No	ID of the div for showing the transcript.	
transcriptHeight	No	the height of transcript block (in pixels) default 148. If 0 is set, the transcript is not shown	
transcriptWidth	No	The width of transcript div (in pixels) Minimum size if 220	
humanOnly	No	shows any transcript if false. If true, shows the transcript only if it's a human generated transcript	default - false

keywordStyle	No	The style of the keywords. Two options 1 and 2.	
--------------	----	---	--

### *Parameters for PHP SDK version*

The following parameters are supported by the JWPlayer in PHP mode

1. (string) mediaID (req) - mediafile id
2. (string) apikey (req) - user's key
3. (string) password (req) - user's password
4. (boolean) serverMode set in true (this is required to be set to true in server mode true)
5. apiURL
6. contextMenu
7. topicHover

PHP Methods (part of the PHP library) used in Server mode.

#### **getKeywords** (keywordStyle, Width, Height)

- keywordStyle 2 styles are support (values are 1 and 2) default is 1
- Width width of the keyword block. If less than 220, it is automatically set to 220.
- Height height of the keyword block default 170

#### **getTranscript** (Width, Height, highlight, humanOnly)

- Width places on the whole block's width. If less than 220, it is automatically set to 220.
- Height default 170
- highlight interval of the transcript highlight time (in seconds) minimum 5 if less than 5, the default is 5, if 0 highlight and scrolling I disabled. Default is 5
- humanOnly if true always show the transcripts section. If false, only show the transcript if it is a human transcript

### *Sharing and Copy*

For share functionality, we use AddThis. You will need to download the AddThis JavaScript library and signup for an AddThis account.

```

<script type="text/javascript" src="sdk-v1.0.3/js/jquery.zclip.js"></script>
<script type="text/javascript">
  var addthis_share = {
    url_transforms: { clean: true, shorten: { twitter: "bitly" } },
    shorteners: { bitly : { login: "xxx",apikeyapikey: "yourkey" } }
  };
  var addthis_config = addthis_config || {};
  addthis_config.data_track_addressbar = false;
  addthis_config.data_track_clickback = false;
</script>
<script type="text/javascript" src="//s7.addthis.com/js/300/addthis_widget.js#pubid=your addthisid"></script>

```

Figure 3 -- sample add this code for sharing (add this to top of web page)

### Streaming

If you are having your own streaming servers, you can simply place the url of the media file stream in jwplayer file parameter (See jwplayer docs for details). Here are 2 examples for a http stream and a rtmp stream.

file: 'http://ec2-50-19-181-137.compute-1.amazonaws.com/20110422\_atc\_07.mp3',

or

file: 'rtmp://xxxx.cloudfront.net/cfx/st/\_definst\_/111/222/filename.mp4',

If you would like use to store and stream, you still need to set the file parameter. But set it to the location of the voicebase stream for the file.

The API method GetFileMetaData retruns 3 fields for this purpose 'streamUrl', 'rtmpUrl' and 'rtmpFile'

- streamUrl -- is an http stream
- rtmpUrl and rtmpFile -- are the location and file name defining the rtmpstream for the file.

It is recommended using the rtmp stream in most cases. The HTTP streams can have some delays and the RTMP streams have better performance at the time this was written.

### Using the GetToken method with the Player

Instead of passing the api password in the javascript code, you can use a token. Tokens are valid for a specific file and they have a timeout. Use the getToken method to get a token. Instead of passing in the 2 parameters "password and apikey" parameters, you should pass in a single parameter named "token".

## Media Formats Supported

Contact VoiceBase support.

## Transcript Format

Transcript format shall be in JSON format. It is an array of words. Each word has 5 fields

1. P – the position
2. S – the start time in milliseconds (measured from beginning of recording)
3. E – the end time in milliseconds (measured from beginning of recording)
4. C – Confidence level
5. W – the word itself
6. M – metadata tag. Values can be word, turn or note. If no m tag, default is word. Turn indicates a speaker turn and contains. Note indicates a transcriber note.

Example:

```
[{"p":1, "s":2132, "e":2372, "c":0.00, "w":"CALL"},  
{"p":2, "s":2382, "e":2522, "c":0.00, "w":"ME"},  
{"p":3, "s":2532, "e":2862, "c":0.00, "w":"ISHMAEL"}]
```

## Sample Programs

One typical use case is to upload a file for indexing, poll the status of the media file and when it is done get the keywords. This PHP sample program does this for one file (passed in on the command line):

```
<?php  
  
# this test program will upload a file check for completion and then fetch the  
# keywords for the file. It requires a file called test.config with 4 lines  
#  
# ---- test.config ----  
# <your apikey>  
# <your password>  
# http://api.voicebase.com/services  
# 1.1  
# -----  
#  
# php uploadandget.php filename  
#  
# example php uploadandget.php myaudio.mp3
```

```
$lines = file('test.config');  
$apikey = trim($lines[0]);  
$password = trim($lines[1]);
```

```

$baseurl = trim($lines[2]);
$version = trim($lines[3]);

# Variables used in each API call

echo "Starting...";

#the files to process
$filename = $argv[1];
#$filename = "fire_and_ice_frost_add_64kb.mp3";

# upload the file with a post
$action = "uploadMedia";
$transcriptType = "machine";           # indicates machine
$public="false";                       # not public
$recordedDate= date();
$imageUrl="http://myserver.com/images/image.jpg";
$collection = "mystuff";
$params = array(

    "version"      => $version,
    "apikey"       => $apikey,
    "password"     => $password,
    "action"       => $action,
    "file"         => "@$filename",
    "transcriptType" => $transcriptType,
    "title"        => $filename,
    "desc"         => "file description",
    "recordedDate" => date("Y-m-d H:i:s"),
    "collection"   => $collection,
    "public"       => $public,
    "sourceUrl"    => "http://api.voicebase.com",
    "lang"         => "en",
    #"modelId"     => $model,
    #"imageUrl"    => "http://mydomain.com/test.jpg",
    "externalId"   => 'external_id - ' . date("Y-m-d H:i:s"),
    "uploadCollection" => 'uploadCollection1'

);
set_time_limit(0);

$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
#curl_setopt($ch, CURLOPT_VERBOSE, 1);

curl_setopt($ch, CURLOPT_TIMEOUT, 0);

```

```

curl_setopt($ch, CURLOPT_LOW_SPEED_LIMIT, 0);
curl_setopt($ch, CURLOPT_LOW_SPEED_TIME, 60);

curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $baseurl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
$response = curl_exec($ch);

curl_close($ch);

$jsonDoc = json_decode($response, true);
$statusCode=$jsonDoc['statusCode'];
$statusMessage=$jsonDoc['statusMessage'];
$mediaId=$jsonDoc['mediaId'];
echo "Uploaded the file, status : ".$statusCode." , ".$statusMessage;
echo "\n";
echo "Media ID : ". $mediaId;
echo "\nResponse:\n";
echo $response;
echo "\n";

# Poll for status of upload (wait till complete or error)
$notReady=true;
while($notReady) {
    sleep(20);

    $url = $baseurl;
    $action = "getFileStatus";
    $url .= "?version=".$version;
    $url .= "&apikey=".$apikey;
    $url .= "&password=".$password;
    $url .= "&action=".$action;
    $url .= "&mediaid=".$mediaId;
    #echo $url;
    $ch = curl_init();
    #curl_setopt($ch, CURLOPT_VERBOSE, 1);
    curl_setopt($ch, CURLOPT_TIMEOUT, 300);
    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $response = curl_exec($ch);
    curl_close($ch);

    $jsonDoc = json_decode($response, true);
    $statusCode=$jsonDoc['statusCode'];
    $statusMessage=$jsonDoc['statusMessage'];
    echo ("\"n\" . $response .\"n\n");
}

```

```

#echo "checking for ". $mediaId. " , status : " . $statusCode . " , " . $statusMessage;
$fileStatus=$jsonDoc['fileStatus'];
#echo $state;

# PROCESSING, MACHINECOMPLETE, HUMANCOMPLETE, ERROR
if ($fileStatus == "MACHINECOMPLETE") {
    $notReady = false;
}

}

# get the keywords
$url = $baseUrl;
$action = "getKeywords";
$url .= "?version=".$version;
$url .= "&apikey=".$apiKey;
$url .= "&password=".$password;
$url .= "&action=".$action;
$url .= "&mediaid=".$mediaId;
$ch = curl_init();
#curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 300);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$response = curl_exec($ch);
curl_close($ch);
#
echo "----- Results -----\\n";
echo $response;
echo "\\n";
?>

```