ViSS Technologies

RESTfull ViSSAPI Specification and User Manual

RESTful API (Application Programming Interface) Specification and User Manual for innovative functionality on Video Indexing and Search, provided by ViSS (Video Search Systems)
www.viss-tech.com

Revision 2.50

February 14, 2013

1. Document Specification

Document Attribute	Document Attribute Value	
Title	RESTful ViSSAPI Specification and User Manual	
CSCI[1]	02-00-00-0A31D-02-50	
Category	User Manual, Technical Specification	
File	ViSS_RESTful_API_User_Manual.docx	
Pages	14	
Distribution		
Author	ViSS Technical Department	
Distribution Characteristics	Public	
Distribution List	Public	
Status		
Approvals		**Date**
Project Manager	Federico Carrasco (CTO)	Feb. 14, 2013
Project Quality Controller	Federico Carrasco (CTO)	Feb. 14, 2013

[1] Computer Software Configuration Item

2. Table of Contents

3. Executive Summary

This document describes the ViSSAPI (Application Programming Interface), by ViSS Technologies. This ViSSAPI is a RESTful API that provides the required set of controls to access the powerful functions of ViSS for video indexing and video search.

With this ViSSAPI, ViSS introduces to the video industry a new, ultra-fast, stream based, multimodal video search engine, with an innovative method for characterizing (fingerprinting) video segments. ViSSAPI enables fast creation of fingerprint database (indexing functionality) and fast video retrieval from a video repository (search functionality).

4. Contact information and support

ViSSAPI has been designed and developed by ViSS (Video Search Systems), an early stage video processing company with over two years of innovative research in the development of a revolutionary core technology, acting as a versatile multimodal video search engine. Our technology provides unique solution to the Media Content retrieval problem.

We are a Silicon Valley based company located at Sunnyvale CA, (440 N. Wolfe Rd. - Sunnyvale, CA 94085) constantly evolving our core technology to meet the demanding field of video and media content.

Our team of engineers will provided you top level support related on our API, with samples, and solutions to your technical enquires.

www.viss-tech.com
ViSS (Video Search Systems)
440 N. Wolfe Rd. Sunnyvale, CA 94085
info@viss-tech.com | +1 408 459 1504

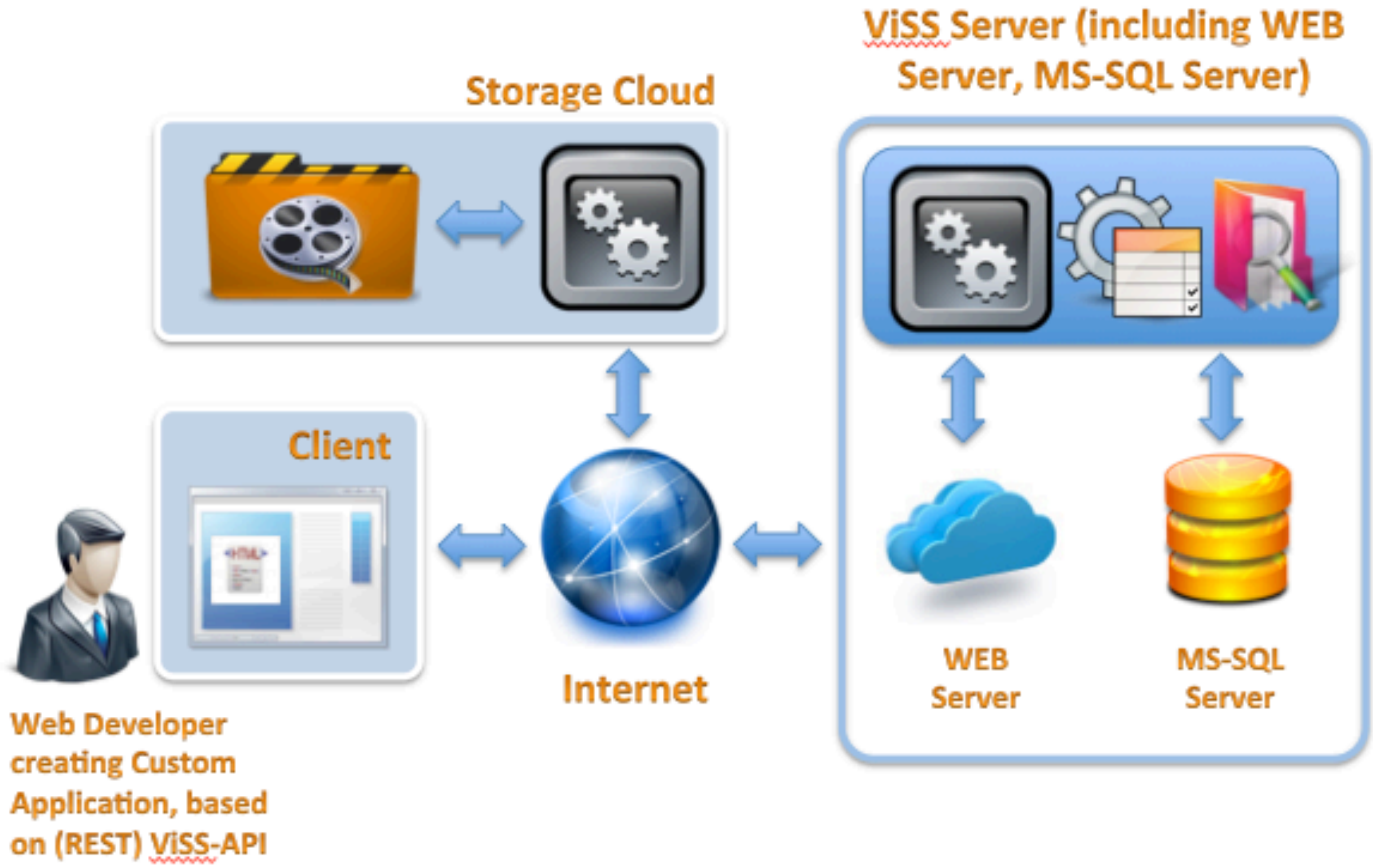
5. RESTful ViSSAPI on the Cloud

The overall schema of ViSS technology on the cloud is depicted in the next figure. The main parts of this technology reside on the ViSS server. This implements the following main functions:

- Connect with the Web Server and implement the REST protocol
- Insert the info (fingerprint) extracted from videos into the MS-SQL database (Indexing)
- Search an indexed video among the already indexed videos residing in an MS-SQL database
- List all videos in the database
- Delete a video from the database

Videos to be indexed can reside either in the ViSS server or in a cloud storage device (e.g. Amazon S3)

A web developer can create custom services, based on the agnostic RESTful API of ViSS, using the web technology and language (php, JavaScript, Python, Ruby, etc.) of his choice.


ViSS Server (including WEB Server, MS-SQL Server)
Storage Cloud
Client
<HTML>
Internet
WEB Server
MS-SQL Server
Web Developer creating Custom Application, based on (REST) ViSS-API

ViSS technology on the cloud

6. ViSSAPI Reference

6.1 General schema of URL

http://<IP address of ViSS Server>:4567/<command ID>/<parameters>/<parameters>/<parameters>

where <command ID> can be one of the following values:

Command	Command ID
Connect to Server	1
Disconnect from Server	2
List of indexed videos	6
Delete an indexed video	7
Index Setup & Index	34
Search Setup	**53**
Search (simple)	5
Search Setup & Search (simple)	535
Search (extended)	51
Get Search Results	52

Note:

For all the examples listed below, we use a sample user with user name (un): einstein and password (pw): viss. User specific credentials, user name (un) and password (pw) will be given to any registered user on our service and will uniquely identify the user. Those credentials will be used as parameters on our ViSSAPI calls.

Moreover, those credentials are valid for the access of our ftp server that the used has to upload his video files, BEFORE the usage of the ViSSAPI calls.

6.2 Connect to Server

Connects to a ViSS videos database.

URL
http://<IP address of ViSS Server>:4567/**1**/<VISS database server>/un=<user name>,pw=<password>

Example API call
http://192.102.7.47:4567/**1**/VISSAPISERVER/un=einstein,pw=viss

Response
ViSSDBConnect() status: OK !

Comments
:<IP address of ViSS Server> and <VISS database server> fields will be provided by ViSS

6.3 Disconnect from Server

Disconnects from a ViSS video database.

URL

http://<IP address of ViSS Server>:4567/**2**/<VISS database server>/un=<user name>,pw=<password>

Example API call
http://192.102.7.47:4567/**2**/VISSAPISERVER/un=einstein,pw=viss

Response
ViSSDBDissconnect() status: OK !

Comments
<IP address of ViSS Server> and <VISS database server> fields will be provided by ViSS

6.4 List of indexed videos

Returns the list of videos already indexed into the database.

URL
http://<IP address of ViSS Server>:4567/**6**/un=<user name>,pw=<password>

Example API call
http://192.102.7.47:4567/**6**/un=einstein,pw=viss

Response
JSON formatted objects, representing data for each indexed video in the database, owned to the specific user.

The string fields are: “videoID”, “videoName”, “videoFPS”, “videoDuration”.

The next example presents, as a result, two videos objects: “0” and “1”, with the string fields as described below:

```
{
"0":    {
        "videoID":"102",
        "videoName":"e1_Amalfi.mp4",
        "videoFPS":"25",
        "videoDuration":"10:28.01"
        },

"1":    {"videoID":"213",
        "videoName":"e2_Amalfi.mp4",
        "videoFPS":"25",
        "videoDuration":"10:28.01"
        }
}
```

6.5 Delete an indexed video

Deletes a video from the database.

URL
http://<IP address of ViSS Server>:4567/**7**/un=<user name>,pw=<password>/<video ID>

Example API call
http://192.102.7.47:4567/**7**/un=einstein,pw=viss/666

Response
1). On successful deletion:
ViSSDeleteVideo() status: OK !
2). On failure due to non existent video ID:
ViSSDeleteVideo() status: ERROR: ["PROBLEM during DELETE; video ID does not exist!"]
3). On failure due to non recognizable user ID:
ViSSDeleteVideo() status: ERROR: ["PROBLEM during DELETE; USER does not exist!"]

Comment
<video ID> is the video ID of the video to be deleted. A list of indexed videos is required to be acquired, prior to any deletion, so the correct video is deleted.

6.6 Index Setup and Index Completion

Indexes the video file and inserts the indexing data to the database.

URL
http://<IP address of ViSS Server>:4567/**34**/un=<user name>,pw=<password>/<video file>/<scene change storage option: 0 or 1>

Scene change storage options:

- "1", will produce .jpg image files of the scene changes in the video, as detected by ViSS engine.
- "0", no scene change .jpg image files will be created.

Example API call
http://192.102.7.47:4567/**34**/un=einstein,pw=viss/Godfather.mp4/1

Response
1). On success: ViSSIndex() status: OK !
2). While still in the process of extracting the video index:
ViSSIndexSetup() status: ERROR: ["Extracting frames info in progress: [1069 of 2675 (40%)]"]

6.7 Search Setup

Setups the search engine by defining the video or video fragment to be used as query

URL
http://<IP address of ViSS Server>:4567/**53**/un=<user name>,pw=<password>/<video ID>/from=<frame start>,to=<frame end> or from=<time start>,to=<time end>

Example API call
http://192.107.7.47:4567/**53**/un=einstein,pw=viss/666/from=16,to=1024
or
http://192.107.7.47:4567/**53**/un=einstein,pw=viss/666/from=000:05.17,to=012:10.24

Response
1). On success: ViSSCreateQuery() status: OK !
2). Video to be used as query is non-defined or void of usable content:
ViSSCreateQuery() status: ERROR: ["No records inserted in the QUERRY; most probably video ID does not exist!"]

Comments
<video ID> for the video that should be searched for matches in the video database.

The “from” and “to” parameters can be either frame ranges or time ranges.

Frame ranges start from 0 up to 2147483648.

Time ranges from 000:00.00 up to 199:59.99 where the first three digits are minutes, the second two are seconds (from 00 up to 59) and the third two are hundredths of a second (from 00 up to 99).

Don’t leave blank spaces in the “from”, “to” fields.

6.8 Search (simple)

Searches the video, as defined by the previous Search Setup step, into the database.

URL
http://<IP address of ViSS Server>:4567/**5**/un=<user name>,pw=<password>/

Example API call
http://192.102.7.47:4567/**5**/un=einstein,pw=viss

Response
ViSSSearch() status: OK !

6.9 Search Setup and Search (simple)

Is the sequential combination of Setup and simple Search, as defined in the previous paragraphs.

URL
http://<IP address of ViSS Server>:4567/**535**/un=<user name>,pw=<password>/<video ID>/from=<frame start>,to=<frame end> or from=<time start>,to=<time end>

Example API call
http://192.107.7.47:4567/**535**/un=einstein,pw=viss/666/from=16,to=1024
or
http://192.107.7.47:4567/**535**/un=einstein,pw=viss/666/from=000:05.17,to=012:10.24

Response
1). On success: ViSSSearch() status: OK !
2). Video to be used as query is non-defined or void of usable content:
ViSSCreateQuery() status: ERROR: ["No records inserted in the QUERRY; most probably video ID does not exist!"]

Comments
The “from” and “to” parameters can be either frame ranges or time ranges.

Frame ranges start from 0 up to 2147483648.

Time ranges from 000:00.00 up to 199:59.99 where the first three digits are minutes, the second two are seconds (from 00 up to 59) and the third two are hundredths of a second (from 00 up to 99).

Don’t leave blank spaces in the “from”, “to” fields.

6.10 Get Search Results

Returns to the user the results of the search, as produced by the previous Search step. This result is the set of videos that include as of their part, the video with that <videoID> used in the Search action.

URL
http://<IP address of ViSS Server>:4567/**52**/un=<user name>,pw=<password>>/<results display option: 0 or 1>

Results display options:

- "0", JSON results of the search.
- "1", JSON results AND .jpg array with the jpg images on the resulted scene changes.

Example API call
http://192.102.7.47:4567/**52**/un=einstein,pw=viss/1
or
http://192.102.7.47:4567/**52**/un=einstein,pw=viss/0

Response
JSON formatted objects, representing data for the resulted videos. **The videos are sorted in descending videoID order.**

The string fields are: “videoID”, “videoName”, “videoFPS”, “videoDuration”, “videoFirstFitFrame”, videoFirstFitTime, “videoLastFitFrame”, “videoLastFitTime”.

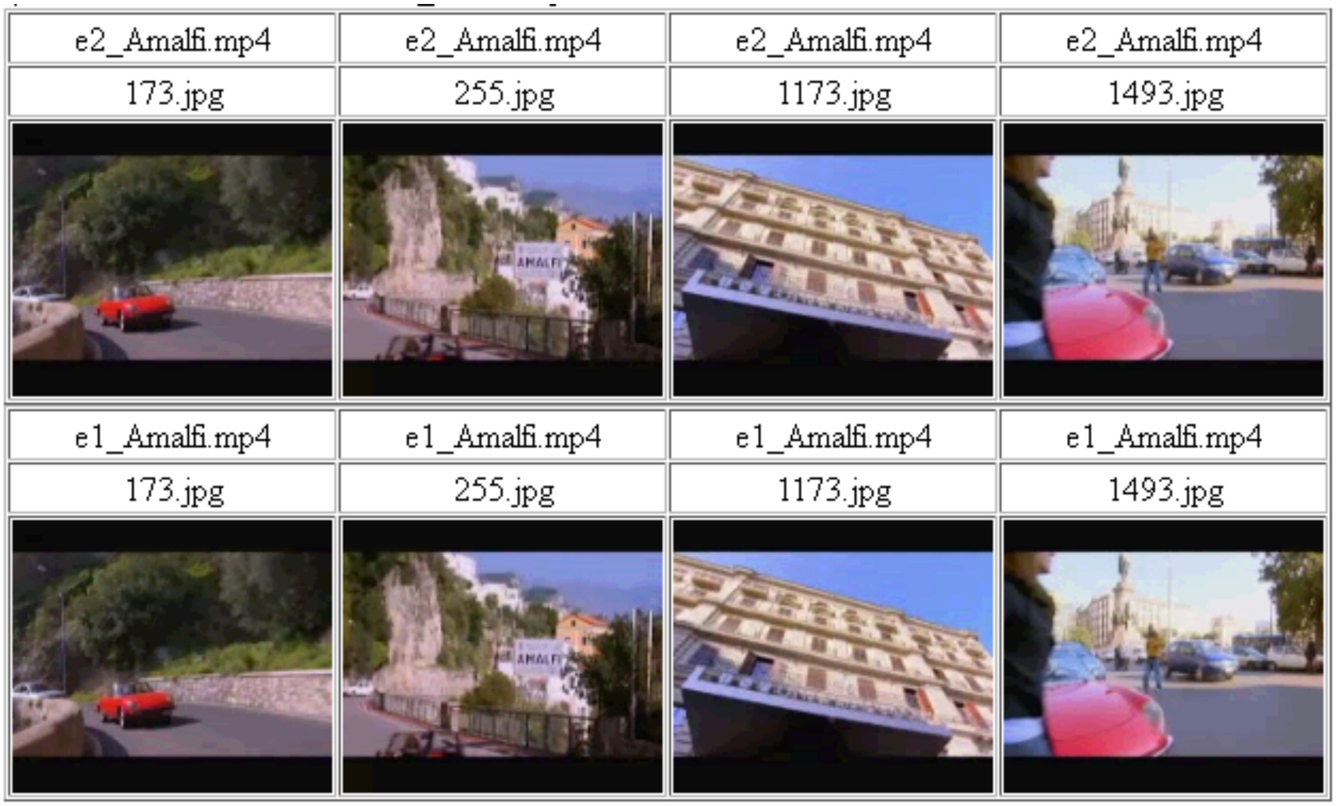
The next example presents, as a result, two videos objects: “0” and “1”, with the string fields as described below:

```
{
"0":    {
        "videoID":"1",
        "videoName":"e2_Amalfi.mp4",
        "videoFPS":"25.000",
        "videoFirstFitFrame":"173",
        "videoFirstFitTime":" 00:06.92",
        "videoLastFitFrame":"1493",
        "videoLastFitTime":"00:59.72"
        },

"1":    {
        "videoID":"0",
        "videoName":"e1_Amalfi.mp4",
        "videoFPS":"25.000",
        "videoFirstFitFrame":"173",
        "videoFirstFitTime":" 00:06.92",
        "videoLastFitFrame":"1493",
        "videoLastFitTime":" 00:59.72"
        }
}
```

In case of the “1” parameter used as results display options, the results is the JSON stream plus the array of images for the scenes included in JSON.

Next image depicts an example of the graphical results:



e2_Amalfi.mp4	e2_Amalfi.mp4	e2_Amalfi.mp4	e2_Amalfi.mp4
173.jpg	255.jpg	1173.jpg	1493.jpg
e1_Amalfi.mp4	e1_Amalfi.mp4	e1_Amalfi.mp4	e1_Amalfi.mp4
173.jpg	255.jpg	1173.jpg	1493.jpg

7. Using the RESTful ViSSAPI

This paragraph presents the basic methodology and usage of the ViSSAPI functionality.

Step 1. Connect to the database
http://192.102.7.47:4567/**1**/VISSAPISERVER/un=einstein,pw=viss

Step 2. Index (=insert a video)
http://192.102.7.47:4567/**34**/un=einstein,pw=viss/Godfather.mp4/1

Step 3. Search a video within the database
http://localhost:4567/**535**/un=einstein,pw=viss/666/from=16,to=1024

Step 4. Get Search Results
http://192.102.7.47:4567/**52**/un=einstein,pw=viss/1

Step 5. Disconnect from the database
http://192.102.7.47:4567/**2**/VISSAPISERVER/un=einstein,pw=viss

While connected to the database, all videos indexed by the user can be listed, new ones can be inserted or existing ones deleted, as specified above.

8. How to Upload Videos

ViSSAPI is a cloud service providing to the user the ability to upload the Video files to our servers. Consequently, the first step that the user must perform, before the usage of ViSS RESTful API (as described in the previous paragraphs) is to upload his Video files, in any video encoding format, to our repository server, using the user name and password provided by our support team. This user name and password combination is the one and only identifying the user and used as parameters in all ViSSAPI calls.

The same user name and password combination can be used to access our ftp server using any ftp client. The user files will be stored at user specific directories that can be accessed only by the registered user, using his user name and password unique credentials.