Speaker Verification Engine©

Application Program Interface

ValidVoice™
10832 Old Mill Rd. Ste 3
Omaha, NE 68154

March 27, 2017

V-0107-SVE-0300

Copyright Notice

Disclaimer

This document has been prepared with all care to reflect the intended operation of the described software for the stated version. ValidVoice makes no warranties as to the operation or fitness for purpose of the software and the user will determine through evaluation and testing its suitability for their application. Where operation of the software described deviates from this document ValidVoice at its sole discretion will amend either the software or the documentation or both the software and documentation to make consistent. This document provides no obligation for ValidVoice to support any features described in future versions.


alid
VOICE

Document History

Revision	Software Version	Published	Notes
001	0300	Nov-2010	Initial Implementation
002	0300	Nov-2015	Slight Modifications Made
003	0300	Mar-2016	Formatted with ValidVoice Document Template
004	0300	May-2016	Updated to reflect changes in API
005	0300	May-2016	Added audio pre-requisites in section 5
006	0300	May-2016	Updated for Version 1 of the SaaS API Interface
007	0300	Sep-2016	Updated the cancel method
008	0300	Mar-2017	Liveness Verification added


alid
VOICE

About US

Established in 2005, ValidVoice™ provides the core Speaker Verification Engine (SVE) for some of the leading deployments of this technology. ValidVoice now has operations in North America, and Asia Pacific with plans for further expansion. Long term customers have proven the reliability and robustness of the ValidVoice SVE.

ValidVoice's Speaker Verification Engine provides significantly better performance than other available engines. Independent comparisons show performance in terms of Equal Error Rate (ERR) as being an order of magnitude better than the leading players. ValidVoice packages the SVE with an Application Programming Interface (API). The packaged engine is available to corporate and application developer customers as a superior authentication engine to those already on the market.

ValidVoice continues to build on its more than 5 years of intensive development. EER is considered the best measure of usability of a verification system. It is the measure of where the potential for false accepts and false rejects is balanced. A lower EER value means superior customer satisfaction whilst maintaining the required level of security.

ValidVoice's product focus is to provide enhanced security and privacy for the web on-line world and emerging 3G technologies (transactional & content management). ValidVoice aims to provide security, privacy and convenience for the individual and also reduce costs for the business and call center operator. ValidVoice enhances privacy in offshore and virtual contact centers (on-shoring and home agents), but still allow businesses to utilize the cost benefit from operating in low cost jurisdictions. ValidVoice assists in driving automation of self service requirements by providing a secure and convenient method of identifying customers and employees. ValidVoice also continues to work closely with leading research institutions to maintain the Speaker Verification Engine's leading edge performance.

Table of Contents

Table of Figures


alid
VOICE

1. Introduction

Voice biometrics also known as Voice Authentication or Speaker Verification, verifies the identity of the speaker based on numerous vocal tract traits and style of speech. Voice biometrics is based on the fact that each individual's voice is unique. It is more effective and efficient than other biometric solutions as well as one of the latest non-intrusive technologies commercially available for identity verification. ValidVoice's Speaker Verification Engine (SVE) processes both Text Independent and Text Dependent utterances to provide flexibility across a range of security requirements. These utterances can be used to not only identify what the speaker is saying but it can also be used to identify the speaker by his/her vocal traits.

The Speaker Verification Engine (SVE) by itself when run does nothing except to wait for incoming audio. It is developed with a set of powerful API calls that allow for seamless integration into existing environments. In a live production environment there is usually some other application that interacts with the SVE, such as an Interactive Voice Response (IVR) system that forwards incoming requests to the SVE for enrollment and verification.

This document covers the use of the Representational State Transfer (REST) Application Program Interface (API) for ValidVoice's Speaker Verification Engine (SVE); it does not address issues of engine deployment or tuning. Should greater control of the ValidVoice engine be required, such as changing the locations of stored data or the use of specific encryptions contact your nearest ValidVoice representative for additional support.

This interface supports enrollment and verification processes. It is up to the application developer to ensure that all other processing logic, security and procedures are developed appropriately. Storage of enrollment vectors is the responsibility of the application. The general structure is depicted in Figure 1 below.

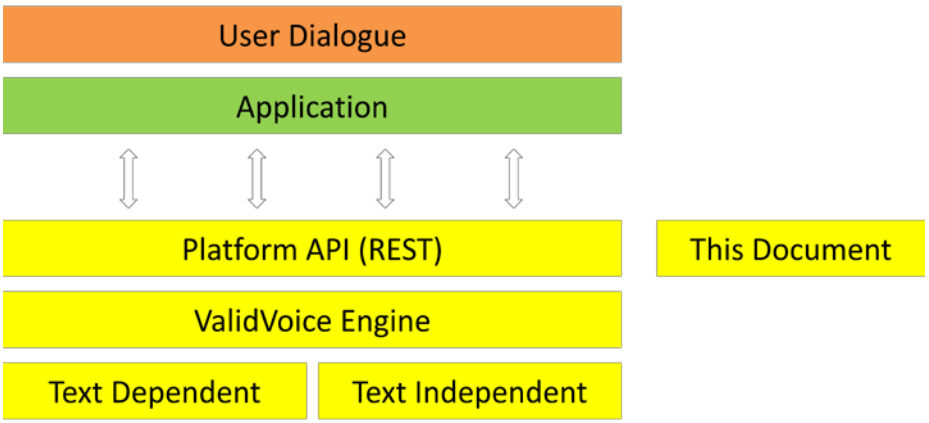

Figure 1-1: Platform API Diagram


alid
VOICE

2. Response Messages

The SVE REST API uses the JSON Language to send result responses. The API returns only two HTTP response codes: HTTP 200 OK and HTTP 500 INTERNAL SERVER ERROR. The HTTP 200 OK response message varies from API call to API call, while the HTTP 500 INTERNAL SERVER ERROR response message remains the same.

The following is an example of an HTTP 500 INTERNAL SERVER ERROR response message.

```
/* HTTP 500 - Json Error Response */
{
    /* Error Code (API error, not a catastrophic error) */
    "error":0,

    /* Description of the error code */
    "description":"Description of error"
}
```

It is worth noting that for some instances, the HTTP 200 OK response message can have an error block as well; this is considered a catastrophic error and the data inside needs to be transmitted to ValidVoice for inspection, along with the Developer-Key, Application-Key, Client-Id, and Vv-Session-Id from the message it came from.


alid
VOICE

3. Cancel Method

The Cancel Method is used to cancel an Enrollment or Verification session in the event an error occurred in the voice data phases, noted in sections 4.2 and 5.2.

DELETE - http:// v2ondemand.com:50102/1/sve/Cancel/{reason}

HTTP Request Headers

Name	Type	Values	Description
Vv-Session-Id	string	Session Id returned by start enrollment/verification session(s)	Unique identifier of the enrollment/verification.

Figure 3-1: Session ID Parameter

{reason} – is a "-" separated string of text of length char (64) supplied as the reason for cancelling the session.

URL Example: http://v2ondemand.com:50102/1/sve/Cancel/testing-session-connection

3.1 Successful Result

There is no JSON result response for a successful cancel.


alid
VOICE

4. Enrollment

The following calls are used to start process and terminate an enrollment.

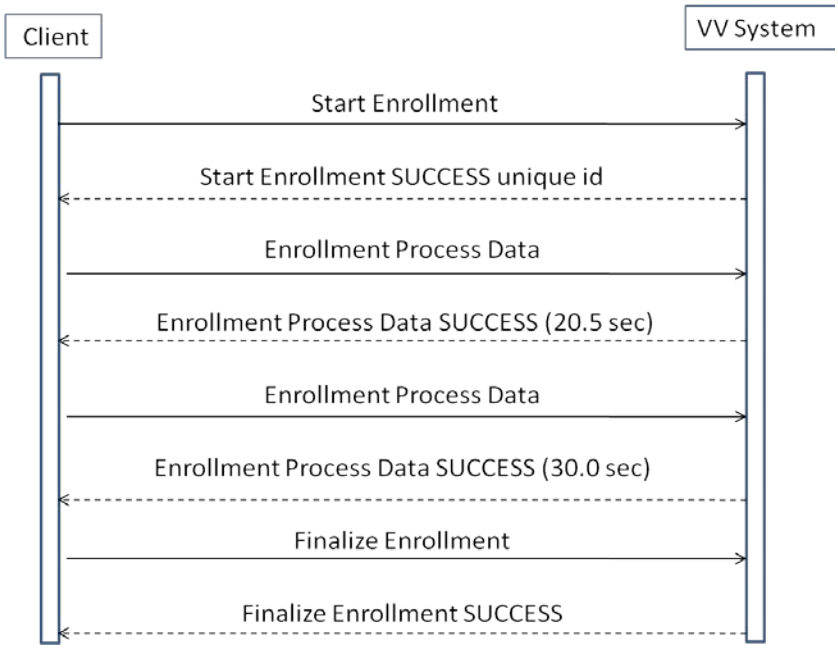

Figure 4-1: Successful Enrollment Sequence

We recommend having over 50 seconds of enrollment data for text independent profiles. In Figure 4-1 above, we have 20.5 + 30.0 = 50.05 seconds. The seconds returned is the amount of usable speech that the SVE feature extraction process found in the enrollment audio data file. It is important to note here that 50 seconds of audio data does not necessarily equal 50 seconds of enrollment speech. A general rule of thumb is to associate every 3 seconds of audio data with 1 second of enrollment speech data.

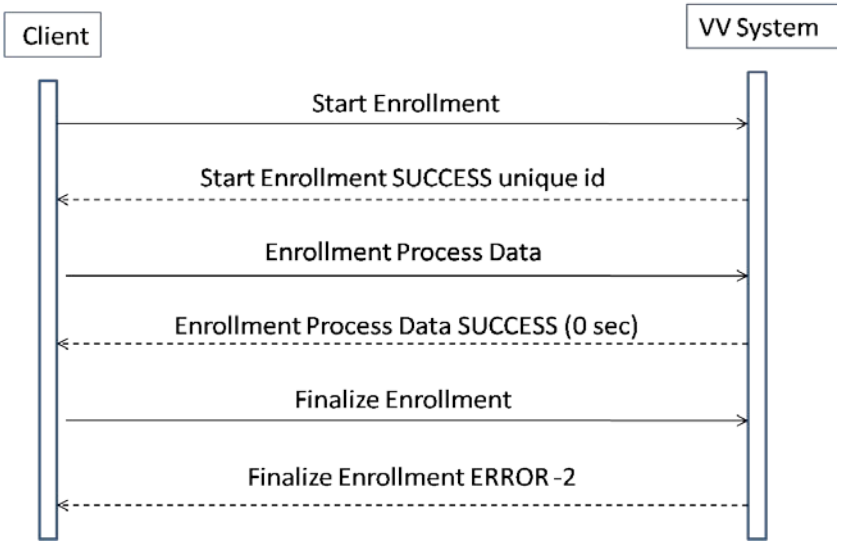

Figure 4-2: Rejected Enrollment Sequence

In this case the client should have called the Enrollment Process Data again, even with the same data. In the case of Text-Dependent processing, this would be the same enrollment phase, while additional data could be sent for Text Independent processing.

4.1 Starting an Enrollment

An enrollment session is initiated using a Developer-Key and an Application-Key to uniquely identify the developers application and requires a unique client id and the sub-pop (gender) of the client enrolling.

POST - http://v2ondemand.com:50102/1/sve/Enrollment/{client_id}/{subpop}

URL Parameters:

Name	Type	Values	Description
client_id	String		Unique-ID any alphanumeric string
subpop	Char	M=MALE F=FEMALE	Supported Values

Figure 4-3: Enrollment Start Parameters

HTTP Request Headers:

Name	Type	Description
Developer-Key	String	The Developer-Key located on the Dashboard
Application-Key	String	The Application specific Key located on the Dashboard
Interaction-Id	String	(Optional) An external session id, used to correlate internal data with.

Figure 4-4: Enrollment Start Request Headers

HTTP Response Headers:

Name	Type	Description
Vv-Session-Id	String	The session Id required to be sent with subsequent processing calls.

Figure 4-5: Enrollment Start Response Headers

4.1.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Enrollment Start Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{ ... },

    /* Profile information only present on enrollment start message */
    "profile.enroll":{

        /* Index number of the profile */
        "index":0,

        /* Kind of Profile: Independent - 1, Liveness - 2 */
        "kind":0,

        /* Type of Audio Codec: pcm_little_endian -or- alaw */
        "codec":"",

        /* Number of seconds of extracted speech required to train an enrollment. */
        "min_seconds_of_speech":0.0
    }
}
```

4.1.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Error Description Code	Description
101	**Session Exception**
104	**Developer has reached their limit**
107	**Unable to acquire a session.**
202	**Unable to start the feature extractor**
302	**Unable to start the enrollment processor**
309	**Invalid Application Key / Application Profile Not Found**
310	**Subpopulation parameter incorrect**
311	**Enrollment Limit Reached**

Figure 4-6: Enrollment Start Status Codes

4.2 Sending Voice Data to An Enrollment

Sending voice data to an enrollment can be called multiple times. For Text-Independent Processing, we recommend to gather at least 50 seconds of buffered vectors.

POST - http://v2ondemand.com:50102/1/sve/Enrollment

HTTP Request Headers:

Name	Type	Values	Description
Vv-Session-Id	string	Session Id returned by start enrollment session	unique identifier of the enrollment. Received in the start enrollment API.

Figure 4-7: Enrollment Process Session ID Parameter

Additional Notes:

- Uses File Upload Standard RFC 1867.
- "Form-based File Upload in HTML". An HTTP request submitted using the POST method with a content-type of "multipart/form-data".
- The audio data to be transmitted needs to be encoded by the same audio codec that was received with the ***profile.enroll*** JSON object from the start enrollment HTTP Data Stream. The audio data is **required** to be transmitting as a raw wav stream without any "RIFF" headers.

Stream Data:

Name	Type	Values	Description
data	binary	Max file size is 2 megabytes	Form data field name to store binary data is "data"

Figure 4-8: Enrollment Process Data Parameter

4.2.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Enrollment Process Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{
        ...
    },

    /* Present on enrollment process and end messages */
    "result.enroll":{

        /* The client_id used in the enrollment start message */
        "{client_id}":{

            /* Index number of the profile */
            "index":0,

            /* Number of seconds of speech extracted from all voice samples thus far */
            "seconds_extracted":0.0,

            /* Current processing call sampling error */
            "error":0
        }
    }
}
```

Sampling Error Code	Description
0	Success
303	Need More Voice
304	Enrollment Training Error (Only shows up on Finalizing an Enrollment)

Figure 4-9: Enrollment Sampling Codes

4.2.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Error description code	Description
101	Session Exception (See Log)
102	Invalid Session (Session Timeout?)
103	Corrupt Session Id
104	Developer Limits Reached
311	Enrollment Limits Reached
315	Invalid Post Data, No Stream Found, or No voice found in stream
317	Invalid Post Data, Stream Found, Missing data field
318	Invalid Post Data, Unknown stream type, Missing form-data
701	Error occurred while updating database

Figure 4-10: Enrollment Process Status Codes

4.3 Finalize an Enrollment

Finalize an Enrollment, train and save data, destroy the session.

DELETE - http://v2ondemand.com:50102/1/sve/Enrollment

HTTP Request Headers:

Name	Type	Values	Description
Vv-Session-Id	string	Session Id as returned in start enrollment process	unique identifier of the enrollment. Received in the start enrollment process.

Figure 4-11 Enrollment Terminate Session ID Parameter

4.3.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Enrollment End Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{
        ...
    },

    /* Present on enrollment process and end messages */
    "result.enroll":{

        /* The client_id used in the enrollment start message */
        "{client_id}":{

            /* Index number of the profile */
            "index":0,

            /* Number of seconds of speech trained from all voice samples */
            "seconds_trained":0.0,

            /* Current processing call sampling error */
            "error":0
        }
    }
}
```

4.3.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Error description code	Description
101	**Session Exception (See Log)**
102	**Invalid Session (Session Timeout?)**
103	**Corrupt Session Id**
104	**Application Limits Reached**
304	**Enrollment Training Error**
701	**Error occurred while updating database**

Figure 4-12: Enrollment Finalize Status Codes


alid
VOICE

5. Verification

The following calls are used to start, process, and terminate a verification:

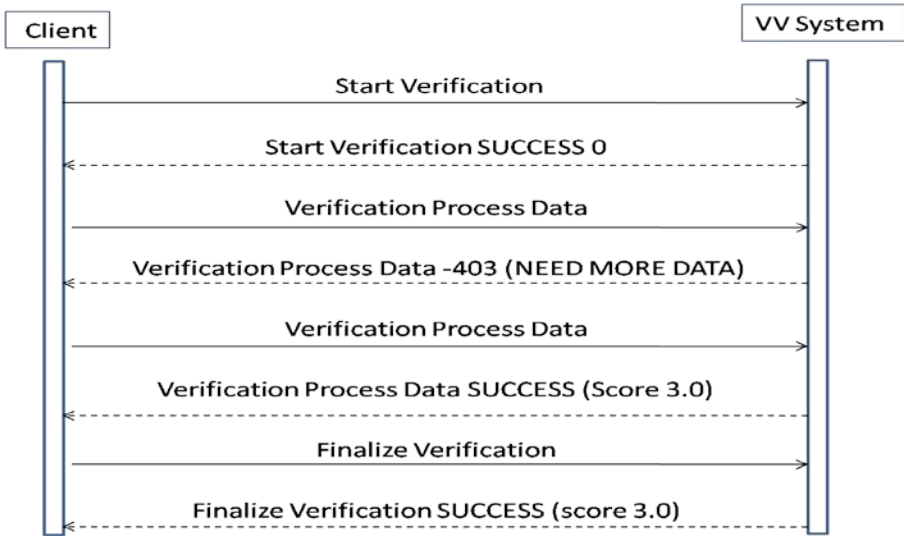

Figure 5-1: Successful Verification Process Flow

The general sequence for a verification is to start a session which may then be left open as long as required. This supports being able to do continuous verifications during the course of an interaction. A verification session can be finalized at any time.

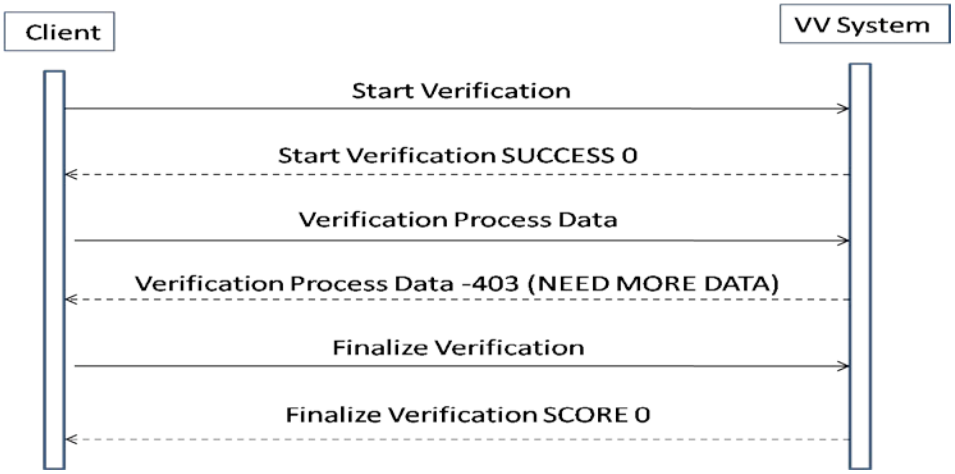

Figure 5-2: Failed Verification Example

Figure 5-2 above gives an example of what a failed verification looks like. In this example, the error returned indicates more data is required to produce a reliable result. In this case we would persist and submit additional data, or if trying to verify against a text dependent profile, we would resubmit the same data file.

5.1 Starting a Verification

Initiate a verification session using a pre-configured profile and the same unique client-id that was used in the enrollment session.

POST - http:// v2ondemand.com:50102/1/sve/Verification/{client_id}

URL Parameters:

Name	Type	Description
client_id	String	Client Id for the enrollment that this verifier is claiming – i.e. the enrollment that we wish to verify against.

Figure 5-3: Verification Start Parameters

HTTP Request Headers:

Name	Type	Description
Developer-Key	String	The Developer-Key located on the Dashboard
Application-Key	String	The Application specific Key located on the Dashboard
Interaction-Id	String	(Optional) An external session id, used to correlate internal data with.

Figure 5-4: Verification Start Request Headers

HTTP Response Headers:

Name	Type	Description
Vv-Session-Id	String	The session Id required to be sent with subsequent processing calls.

Figure 5-5: Verification Start Response Headers

5.1.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Verification Start Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{
        ...
    },

    /* Profile information only present on verification start message */
    "profile.verify":{

        /* Index number of the profile */
        "index":0,

        /* Profile Kind: Independent - 1, Liveness - 2 */
        "kind":0,

        /* Profile Type: Single - 2 */
        "type":0,

        /* Type of Audio Codec: pcm_little_endian -or- alaw */
        "codec":"",

        /* Passing Threshold */
        "pass":0

        /* Failing Threshold */
        "fail":0
    }
}
```

5.1.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Error description code	Description
101	**Session Exception**
104	**Developer has reached their limit**
107	**Unable to acquire a session**
202	**Unable to start the feature extractor**
402	**Unable to start the verification processor**
408	**Invalid Application Key / Application Profile Not Found**
409	**Invalid Parameter specified**
410	**No Model found for the specified client id**
412	**Verification limits have been reached**

Figure 5-6: Verification Start Error Codes

5.2 Sending Voice Data

Sending Voice Data can be called multiple times even with the same data.

POST - http:// v2ondemand.com:50102/1/sve/Verification

HTTP Request Headers:

Name	Type	Values	Description
Vv-Session-Id	String		The Id received in the start verification result.

Figure 5-7: Verification Process Session ID Parameter

Additional Notes:

- Uses File Upload Standard RFC 1867.
- An HTTP request submitted using the POST method with a content-type of "multipart/form-data".
- The audio data to be transmitted needs to be encoded by the same audio codec that was received with the ***profile.verify*** JSON object from the start verification HTTP Data Stream. The audio data is **required** to be transmitting as a raw wav stream without any "RIFF" headers.

Streaming Data:

Name	Type	Values	Description
data	binary	Max file size is 2 megabytes	Form data field name to store binary data is "data"

Figure 5-8: Verification Process Data Parameter

5.2.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Verification Process Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{
        ...
    },

    /* Present on verification process and end messages */
    "result.verify":{

        /* The client_id used in the verification start message */
        "{client_id}":{

            /* Index number of the profile */
            "index":0,

            /* Current processing call sampling error */
            "error":0,

            /* Number of seconds of speech extracted from all voice samples thus far */
            "seconds_extracted":0.0,

            /* Score determined from all samples thus far */
            "score":0.0,

            /* Status of score. (P)assing, (A)mbiguous, (F)ailing */
            "status":"P"
        }
    }
}
```

Sampling Error Code	Description
0	Success
403	Need More Voice

Figure 5-9: Verification Sampling Codes

5.2.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Error description code	Description
101	Session Exception (See Log)
102	Invalid Session (Session Timeout?)
103	Corrupt Session Id
104	Developer Limits Reached
412	Verification Limits Reached
416	Invalid Post Data, Unknown stream type, Missing form-data
417	Invalid Post Data, No Stream Found, or No voice found in stream
418	Invalid Post Data, Stream Found, Missing data field
701	Error occurred while updating database

Figure 5-10: Verification Process Status Codes

5.3 Sending Live Voice Data

Sending live voice data, or more commonly referred to as a liveness test, is a proven method of ensuring the person verifying is alive, present at the time the voice is captured, and is not a recording.

POST - http:// v2ondemand.com:50102/1/sve/Verify/Liveness/{detect_liveness_text}

HTTP Request Headers:

Name	Type	Values	Description
Vv-Session-Id	String		The Id received in the start verification result.

Figure 5-11: Verification Liveness Process Session ID Parameter

Additional Notes:

- Uses File Upload Standard RFC 1867.
- An HTTP request submitted using the POST method with a content-type of "multipart/form-data".
- The audio data to be transmitted needs to be encoded by the same audio codec that was received with the ***profile.verify*** JSON object from the start verification HTTP Data Stream. The audio data is **required** to be transmitting as a raw wav stream without any "RIFF" headers.

Streaming Data:

Name	Type	Values	Description
data	binary	Max file size is 2 megabytes	Form data field name to store binary data is "data"

Figure 5-12: Verification Liveness Process Data Parameter

5.3.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Verification Process Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{
        ...
    },

    /* Present on verification process and end messages */
    "result.verify":{

        /* The client_id used in the verification start message */
        "{client_id}":{

            /* Index number of the profile */
            "index":0,

            /* Current processing call sampling error */
            "error":0,

            /* Number of seconds of speech extracted from all voice samples thus far */
            "seconds_extracted":0.0,

            /* Score determined from all samples thus far */
            "score":0.0,

            /* Status of score. (P)assing, (A)mbiguous, (F)ailing */
            "status":"P"

           /* Allowed results. (Y)es, (N)o */
            "alive":"Y"
        }
    }
}
```

Sampling Error Code	Description
101	Session Exception (See Log)
403	Need More Voice

Figure 5-13: Verification Liveness Sampling Codes

5.3.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Sampling Error Code	Description
0	Success
102	Invalid Session (Session Timeout?)
103	Corrupt Session Id
104	Developer Limits Reached
412	Verification Limits Reached
416	Invalid Post Data, Unknown stream type, Missing form-data
417	Invalid Post Data, No Stream Found, or No voice found in stream
418	Invalid Post Data, Stream Found, Missing data field
420	Returns a Non-Live State
701	Error occurred while updating database

Figure 5-14: Verification Liveness Process Status Codes

5.4 Finalize a Verification

Finalize the verification session and get the final score result.

DELETE - http:// v2ondemand.com:50102/1/sve/Verification

HTTP Request Headers:

Name	Type	Values	Description
Vv-Session-Id	String		The Id received in the start verification result.

Figure 5-15: Verification Terminate Session ID Parameter

5.4.1 Successful Result

HTTP Status Code: 200 (Success)

```
/* HTTP 200 - JSON Verification End Response */
{
    /*
     * Only present if a catastrophic error occurs inside the API.
     * Data inside this block should be sent to ValidVoice for investigation.
     */
    "error":{
        ...
    },

    /* Present on verification process and end messages */
    "result.verify":{

        /* The client_id used in the verification start message */
        "{client_id}":{

            /* Index number of the profile */
            "index":0,

            /* Current processing call sampling error */
            "error":0,

            /* Number of seconds of speech extracted from all voice samples thus far */
            "seconds_extracted":0.0,

            /* Score determined from all samples thus far */
            "score":0.0,

            /* Status of score. (P)assing, (A)mbiguous, (F)ailing */
            "status":"P",

           /* Has the client been authorized. Is status equal to P. true -or- false */
            "authorized":true

        }
    }
}
```

5.4.2 Failure Result

HTTP Status Code: 500 (Internal Server Error)

Error description code	Description
101	**Session Exception (See Log)**
102	**Invalid Session (Session Timeout?)**
103	**Corrupt Session Id**
701	**Error occurred while updating database**

Figure 5-16: Verification Terminate Status Codes


alid
VOICE

6. Audio Formats

The engine currently supports the following audio formats:

- Wave A-Law, 8000 Hz, 64 kbps, mono
- Wave PCM signed 16 bit, 8000 Hz, 128 kbps, mono

The expected audio format is returned in the Enrollment and Verification initialization response JSON messages located in the field ***codec***. Please note, that failure to send in the audio format in any of the specified formats above may result in unintended results, such as for example, an unusually high amount of extracted speech. The extracted speech should always be less than the amount of audio sent to the engine.


alid
VOICE