nVoq.SPS Dictation API Quick Start Guide

Getting Started

This document will walk you through getting started with nVoq.SPS (Speech Processing Services) dictation APIs. nVoq.SPS is comprised of multiple speech-to-text (STT) services that are accessed using RestFUL WebServices calls, packaged as APIs.

To get underway with your integration you will need the following:

1. USB headset with microphone for audio submission/playback
2. nVoq.Administrator / Account Login & Password (Provided by nVoq)
3. Development Sandbox (test.nvoq.com) to submit audio and review results
4. Access to nVoq.SPS/API documentation

    - https://test.nvoq.com/apidoc/howto/index.html
    - https://test.nvoq.com/apidoc (then select Dictation from the menu bar)
    - https://support.nvoq.com/api

Which APIs should I use?

Selecting the right APIs is critical to your success with our platform.
Answering a few quick questions, can help you get underway.

What kind of voice-driven application are you looking to build?

- If you are looking to build a voice-driven solution that will convert audio to text in *real time*, then you will want to use our Real-Time/Streaming Dictation API.

    - WebSocket Dictation
      https://test.nvoq.com/apidoc/howto/ws/index.html

- If your solution will convert/transcribe audio as a *single file* after the dictation is complete, then you will want to use our Batch Dictation API.

    - HTTP Dictation
      https://test.nvoq.com/apidoc/howto/http/index.html

nVoq.SPS Dictation API Quick Start Guide

Will your application support both Dictation and Voice Commands?

- If you intend to support both dictation and voice commands with your application, you have two options to consider:
  1. Voice Commands using the dictation server and Word Markers
     - You can allow commands to be spoken along with the dictation audio (i.e., all audio is submitted to the Dictation API). With this approach the command is transcribed along with the rest of the dictation audio, thus requiring you to *keyword spot* the command in the returning text and initiate the appropriate action.
       - This is the easiest approach and allows end users to include/interlace commands within their dictations.
       - With this approach we would recommend using the Word Markers API
         - https://support.nvoq.com/api-word-markers
         - https://test.nvoq.com/apidoc/dictation#operation/GetAudioProps
       - The Word Markers API will allow you to match the transcribed text with the appropriate audio segment so you can easily filter out the commands from the dictated text.
  2. Voice Commands using the nVoq Mid-dictation Command API service
     - With this approach, we will use our dictation server to identify your pre-defined command phrases and remove them from the transcript.
     - The command will then be returned as a separate action by the WebSocket Dictation API, so that it can be processed correctly by the client.
     - This approach allows the speaker to interlace voice commands along with the dictation (i.e., off the same button press).

What development environment will you use to build your application?

- Audio capture and send will need to be handled by your application. Depending on the development technology chosen (iOS, .NET, JAVA, HTML), the approaches will be different.
- Although nVoq.SPS/APIs do not mandate how the audio capture and send occurs, we can direct you to development resources to help you in your effort. So feel free to ask nVoq Support for additional development support, if you need help getting started.

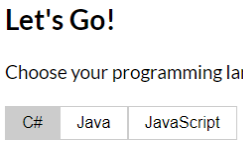
nVoq.SPS Dictation API Quick Start Guide

Will your application be mobile-based or desktop-based?

- If you intend on developing a mobile application, then audio capture and send will need to be handled on the mobile device. Depending on the device type (iOS, Android), the approaches will be different.
- Mobile applications will use the embedded microphone in the device. Many embedded microphones are not noise cancelling, so this can be a factor for dictation accuracy. We would recommend trying out the microphone with other speech applications to ensure the microphone fidelity is sufficient for your needs.
- Desktop-based applications allow for the use of external headsets for audio capture, which can reduce background noise and improve dictation accuracy.
- When designing your application, think through the end-user workflow for both point and click/touch as well as speech to optimize the complete experience with your application.

Are code samples available to help me with my development?

- Yes, code samples are available for review. They are not meant to be copied directly; instead they are considered exemplars to help you with your development.
  - Web Applications: nVoq.API How-To: Web Dictation
  - iOS: https://test.nvoq.com/apidoc/howto/ios/index.html



  - .NET/C# nVoq.API How-To: WebSocket Dictation
  - JAVA: nVoq.API How-To: WebSocket Dictation
  - JavaScript: nVoq.API How-To: WebSocket Dictation

nVoq.SPS Dictation API Quick Start Guide

What audio format is optimal for my dictation application?

- nVoq recommends 16kHz .ogg WebM (Ogg Opus) audio format for all dictation applications. This audio format is a near loss-less compressed version of .wav. Meaning it uses a reduced audio packet size when communicating with our servers - with little degradation in dictation accuracy.
- A listing of the nVoq audio formats supported can be seen below:

Encoding	Ogg (Vorbis)	WebM (Opus)	PCM-16khz	PCM-8khz
Sample Rate	8000* or 16000 Hz	16000 Hz	16000 Hz	8000 Hz
Sample Size	n/a	n/a	16 bit	16 bit
Quality Setting	0.5	default	N/A	N/A
Channels	1 (mono)	1 (mono)	1 (mono)	1 (mono)
Byte Order	N/A	N/A	little-endian	little-endian
Signed / Unsigned	signed	signed	signed	signed
HTTP Format Alias	ogg	webm	pcm-16khz	pcm-8khz
HTTP Content-Type	audio/ogg	audio/webm	audio/x-wav	audio/x-wav
Maximum Upload File Size	21,000,000 bytes	21,000,000 bytes	21,000,000 bytes	21,000,000 bytes

- If your application is capturing audio in a different format (such as 44 kHz), you can use the Sound eXchange (SoX) audio file conversion and analysis utility to convert it to 16kHz .ogg *Vorbis* before sending the audio to our servers.
    - https://sox.sourceforge.net/

What other nVoq.SPS/APIs are available?

- A complete listing of nVoq.SPS/APIs can be found at the following links:
    - https://test.nvoq.com/apidoc/howto/index.html
    - https://test.nvoq.com/apidoc (then select Dictation from the blue menu bar)
    - https://support.nvoq.com/api

nVoq.SPS Dictation API Quick Start Guide

Development Best Practices:

- Ensure you are sending the correct dictation audio format:
  - 16kHz, 16-bit, mono, PCM-WAV
  - 16 kHz, mono, .ogg Vorbis - 0.5 Quality or higher
  - 16 kHz, mono, .ogg Opus (WebM) - 0.5 Quality or higher
- Make sure your WebSocket connection to our dictation servers is established before sending audio
  - Consider buffering audio locally, then stream it to our servers once the connection is established
  - Send audio in 300ms packets or greater
- Gracefully shut down upon error, to keep your application in a good state.
  - For instance, each WebSocket will timeout automatically if no audio is received within 30 seconds. So, you need to handle/code for this possible exception in your application.
- Send audioDone to get Stable text to return faster, since the engine waits a few seconds for more context before finalizing the text.
  - Please keep in mind, that you will have to open a new WebSocket after you send audioDone and the text is received
  - If you want to speed up stable text return, but don't want to close out the current dictation, you can send a BOUNDARYREQUEST which will prompt the dictation server to send you stable text a little bit quicker – but won't close out the WebSocket.
- You can provide end users a simple visual verification that a recording/transcription is in progress, by using Hypothesis text.
  - Also include colors, graphics, gain indicator, button depress indicator, etc. to show that a recording is in process
- If you want your users to be able to edit the text while they're dictating, place Stable text in your application, not Hypothesis text.

nVoq.SPS Dictation API Quick Start Guide

- Make sure you've done your due diligence in testing and diagnosing issues on your side of the API, before reaching out to our Support team. For example:
    - Are the submitting credentials correct? (login and password)
    - Is the system/environment correct? (test.nvoq.com, healthcare.nvoq.com, Canada.nvoq.com)
    - Is the issue affecting multiple user accounts/locations, or just one?

- Include end user logging in your application, to improve troubleshooting.
    - We recommend that you update end user account properties on our servers (via the Account API) upon end user login, such as:
        - Microphone type (Last Mic Mixer Name)
        - IP address (Last Request Ip)
        - O/S name and Rev level (O S name / O S Version)
    - Specific events/times such as:
        - WebSocket Open
        - WebSocket Close
        - audioDone
- If you want to support spoken command and control during a dictation (on a single button press), use Word Markers to identify and remove the spoken command and control phrases from the transcript before rendering the text to the screen.

nVoq.SPS Dictation API Quick Start Guide

Top 10 Development Gotcha's:

1. **Poor Audio Capture**
    - Think through how your users will dictate to your application, and what device they will need/use for input.
    - For instance, if you are developing a web-based application that will be running on a laptop or desktop, we would advocate the use of USB headsets for optimal audio quality.
    - We would also advocate staying away from built in Array mics, as they have a tendency to pick-up background noise.
2. **Audio chunk size too large or too small**
    - Make sure that the audio chunk size is set to 300 milliseconds or just slightly higher.
3. **Sending incorrect audio format to our servers**
    - nVoq dictation servers will only accept:
        - 16kHz, 16-bit, mono, PCM-WAV
        - 16 kHz, mono, .ogg Vorbis - 0.5 Quality or higher
        - 16 kHz, mono, .ogg Opus (WebM) - 0.5 Quality or higher
    - Keep in mind, that if you are sourcing audio from a browser or smartphone, it will most likely be recorded at 44kHz. This means you will need to downsample the audio to 16 kHz before sending to our platform.
4. **Submitting audio with incorrect nVoq.SPS credentials** (i.e., incorrect Account Login or Environment)
    - Make sure you are using proper nVoq.SPS credentials when sending audio to the platform.
    - Your nVoq.SPS credentials will be provided to you by your nVoq development liaison.

5. **Submitting nVoq.SPS account lacks appropriate authorization**
    - Make sure that the account you are submitting audio with has a Dictation role. This can be verified in the nVoq.Administrator console.
6. **Lack of error handling** (e.g., no resubmission on failed socket open)
    - Include appropriate error handling in your application to gracefully handle situations when a resource is unavailable.
    - Also, if you find a resource is unavailable, resubmit the audio a specific amount of times before erroring out.
7. **Improper WebSocket closure**
    - Ensure that you close all open WebSockets, before opening a new one. This will reduce the chance of accidently leaving a WebSocket(s) open – which can tie up dictation servers on our platform.
8. **Improper usage of audio/text mark-up** (Supplanting Hypothesis w Stable text)
    - If you display Hypothesis text within your application, make sure that you use our Text with Mark-up feature so that you can easily identify the beginning and end of the Hypothesis text stream, when replacing it with Stable text.
9. **Buffer audio collection while the WebSocket is opening**
    - WebSockets may take a few milliseconds to setup – so buffer any audio that is spoken by the end user while the WebSocket is opening – then stream the buffered audio once the WebSocket is fully established/open.
10. **Don't forget to unmute your mic before testing your application!**

Commented [RM1]: Switched 7 & 8 order to match ISV Dev presentation