



Application Programming Interface (API) Developer Guide

iSpeech Developer Support
<http://www.ispeech.org/developers>
iSpeech Inc. Version 2.2

Last revised: 3/13/2013
Copyright © 2007 – 2013 iSpeech Incorporated
<http://www.ispeech.org/developers/>

iSpeech Inc. ("iSpeech") has made efforts to ensure the accuracy and completeness of the information in this document. However, iSpeech Inc. disclaims all representations, warranties and conditions, whether express or implied, arising by statute, operation of law, usage of trade, course of dealing or otherwise, with respect to the information contained herein. iSpeech Inc. assumes no liability to any party for any loss or damage, whether direct, indirect, incidental, consequential, special or exemplary, with respect to (a) the information; and/or (b) the evaluation, application or use of any product or service described herein.

iSpeech Inc. disclaims any and all representation that its products or services infringe upon any existing or future intellectual property rights. iSpeech Inc. owns and retains all right, title and interest in and to the iSpeech Inc. intellectual property, including without limitation, its patents, marks, copyrights and technology associated with the iSpeech Inc. services. No title or ownership of any of the foregoing is granted or otherwise transferred hereunder. iSpeech Inc. reserves the right to make changes to any information herein without further notice.

Revision History

PUBLISH DATE	UPDATES
Aug 8, 2011	Document created
Sept 13, 2011	Added ASR
Sept 21, 2011	Added AMR to ASR
Nov 9, 2011	Added voice command examples
Nov 10, 2011	Removed references to ASR Raw POST
Nov 11, 2011	Made output variable explicit for ASR and voice list examples
Nov 17, 2011	Specified HTTP POST/GET instead of REST, fixed /r/n typos
Nov 22, 2011	Added reference for Speex in ASR content-type example
Dec 12, 2011	Added endpadding and startpadding variables
Dec 12, 2011	Added TTS examples, added background highlighting to emphasize examples
Aug 24, 2012	Added position markers and visemes
Aug 24, 2012	Added Flash, Javascript/Flash, Ruby, Python, Perl SDKs
Aug 24, 2012	Added link to automated payment system
Aug 24, 2012	Added pitch parameter to TTS
Aug 30, 2012	Added standard non-freeform speech recognition models
Aug 30, 2012	Added speech recognition speex modes
Sept 10, 2012	Added bit depth
Sept 10, 2012	Added TTS file formats: alaw, ulaw, vox, and mp4
Sept 10, 2012	Added errors
Sept 10, 2012	Changed ASR language to locale
Oct 3, 2012	Add filename parameter to TTS
Nov 12, 2012	Removed eurdutchmale voice
Jan 22, 2013	Added Math Markup Language (MathML) information and examples
Jan 24, 2013	Added Speech Synthesis Markup Language (SSML) information and examples
Jan 25, 2013	Removed "coming soon" from voicemail transcription model

Last revised: 3/13/2013

Copyright © 2007 – 2013 iSpeech Incorporated
<http://www.ispeech.org/developers/>

Page 3

Revision History

PUBLISH DATE	UPDATES
Mar 4, 2013	Corrected content-length for JSON TTS example
Mar 13, 2013	Updated ASR locale list, added Web ASR freeform mode.

Table of Contents

Revision History.....	3
------------------------------	----------

Section 1 Introduction	8
-------------------------------------	----------

Minimum Requirements	
Internet Connection	
HTTP Protocol	
API Key	
Managing API Key Settings	
View/Edit Keys	
API Features	
Information	
Text to Speech	
Automated Speech Recognition	
Position Markers	
Visemes	
API Key Information Retrieval	
Developer Support	
Sales	
Support/Troubleshooting	
Software Development Kits	
Availability	
API Access Pricing	

Section 2 Text to Speech	12
---------------------------------------	-----------

Transaction Types and URL Formats	
HTTP GET/POST: URL Encoded, XML, JSON	
Request Parameters	
Voices	
Voice List Retrieval	
HTTP GET: URL Encoded, XML, JSON	
Speed	
Formats	
Bitrates	
Padding	
Pitch	
Bit Depth	
Filename	
Speech Synthesis Markup Language (SSML) tags	
Math Markup Language (MathML) tags	
Example Transactions	
Summary	
HTTP POST: URL Encoded → Binary Audio	
HTTP POST: JSON → Binary Audio	
HTTP POST: XML → Binary Audio	
HTTP GET: URL Encoded → URL Encoded (this transaction purposely contains an error)	
Math Markup Language HTTP GET/POST: URL Encoded, JSON, XML → Binary Audio	
Error Codes for Text-to-Speech and General Errors	

Table of Contents

Section 3 Automated Speech Recognition29

- Transaction Types and URL Formats
 - HTTP GET/POST: URL Encoded, XML, JSON
- Request Parameters
- Locales
 - Standard Locales
 - Custom Locales
- Speech Recognition Models
 - Standard freeform models
 - Standard non-freeform models
 - Custom
- Speex Modes
- Example Transactions for Freeform Speech
 - Format of Examples
 - HTTP POST: URL Encoded → URL Encoded
 - HTTP POST: JSON → JSON
 - HTTP POST: XML → XML
- Command Lists
 - About
 - Example Transactions for Command Lists
 - Formatting of Examples
 - HTTP POST: XML → XML
 - HTTP POST: URL Encoded → URL Encoded
 - HTTP POST: JSON → JSON
 - HTTP POST: XML → XML -- Detecting multiple audio commands from multiple command lists
- Error Codes for Speech Recognition and General Errors

Section 4 Position Markers.....41

- Transaction Types and URL Formats
 - HTTP GET/POST: URL Encoded, XML, JSON
- Request Parameters
- Introduction
- Example Transaction
 - HTTP GET: URL Encoded → XML
- Marker Information Usage Technique
 - Media Player Considerations
 - Implementation
- Error Codes for Position Markers and General Errors

Table of Contents

Section 5 Visemes.....45

- Transaction Types and URL Formats
 - HTTP GET/POST: URL Encoded, XML, JSON
- Request Parameters
- Introduction
- Example Transaction
 - HTTP GET: URL Encoded → XML
- Viseme Chart
- Viseme Usage Technique
 - Media Player Considerations
 - Implementation
- Error Codes for Visemes and General Errors

Section 1

Introduction

Welcome to the iSpeech Inc. Application Programming Interface (API) Developer Guide. This guide describes the available variables, commands, and interfaces that make up the iSpeech API.

The iSpeech API allows developers to implement Text-To-Speech (TTS) and Automated Voice Recognition (ASR) in any Internet-enabled application.

The API's are platform agnostic which means any device that can record or play audio that is connected to the Internet can use the iSpeech API.

Minimum Requirements

Below are the minimum requirements needed to use the iSpeech API. The API can be use with and without a software development kit (SDK).

Internet Connection

iSpeech services require an Internet connection.

HTTP Protocol

The iSpeech API follows the HTTP standard by using GET and POST. Some web browsers limit the length of GET requests to a few thousand characters.

Request/Responses

Requests can be in URL-encoded, JSON, or XML data formats. You can specify the output data format of responses. For TTS, binary data is usually returned if the request is successful. For speech recognition, URL-encoded text, JSON, or XML can be returned by setting the output variable.

API Key

An API key is a password that is required for access. To obtain an API key please visit: <http://www.ispeech.org/developers> and register for a developer account.

Managing API Key Settings

View/Edit Keys

Manage your API keys by using the iSpeech developer website: <http://www.ispeech.org/developers>. You can request additional features for your API keys on that website.

API Features

Information

You can retrieve the properties of your API keys. Key information includes a voice list, amount of credits, locales, and many other parameters.

Text to Speech

You can synthesize spoken audio through iSpeech TTS in a variety of voices, formats, bitrates, frequencies, and playback speeds. Math markup language (MathML) and Speech synthesis markup language (SSML) are also supported.

Automated Speech Recognition

You can convert spoken audio to text using a variety of languages and recognition models. We can create custom recognition models to improve recognition quality.

Position Markers

You can get the position in time when words are spoken in TTS audio.

Visemes

You can get the position in time of mouth positions when words are spoken in TTS audio.

API Key Information Retrieval

Replace “developerdemokeydeveloperdemokey” with your API key.

HTTP GET Request and Response
<code>http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey &action=information&output=rest</code>
<code>[...]&voice-locale-94-1=ja&voice-locale-94-2=ja-jp&voice-gender-94=female &voice-description-94=Japanese+Female+Voice&wordlimit=40&model=assistant%2cdate %2cnfl%2cnba%2cusmoney%2cmlb%2cnumbersto9%2cnumbersto99%2cnumbersto999%2ctime %2cphonenumber%2cstreets%2csportsteam%2ccitystate&credits=5672606&unlimited=enabled &alias=enabled&asr-sms=enabled&asr-voicemail=enabled&asr-dictation=enabled</code>

Developer Support

Sales

Automated purchasing system: <https://www.ispeech.org/developer/purchase/>

iSpeech sales can be contacted at the following phone number: +1-917-338-7723 from 10 AM to 6 PM Eastern Time, Monday to Friday. You can also email sales@ispeech.org.

Support / Troubleshooting

Please look for the answer to your problem in the iSpeech Developer Forum:
<http://www.ispeech.org/forums/>

Software Development Kits

iSpeech SDKs simplify the iSpeech API. You should use iSpeech SDKs if the option is available. Only mobile SDKs made by iSpeech allow you to use the iSpeech API for free.

Availability

iPhone, Android, BlackBerry, .NET, Java (Server), PHP, Flash, Javascript/Flash, Ruby, Python, Perl

API Access Pricing

PLATFORMS	PRICE
iPhone, Android, BlackBerry	Free with fair usage using iSpeech SDK for non-revenue generating apps. Apps must follow the iSpeech standard usage guidelines for branding.
.NET, Java, PHP, Flash, Ruby, Python, Perl	Between \$0.05 and \$0.0001 per word (TTS) or transaction (ASR), depending on quantity

Section 2

Text to Speech

The iSpeech Text-To-Speech API allows you to synthesize high-quality spoken audio in multiple formats. The iSpeech API doesn't use callbacks because it's fast and synchronous. You'll always receive audio data or an error message in the same HTTP transaction.

Transaction Types and URL Formats

TRANSACTION TYPE	INPUT FORMAT	URL
HTTP GET/POST	URL Encoded	http://api.ispeech.org/api/rest
HTTP GET/POST	XML	http://api.ispeech.org/api/xml
HTTP GET/POST	JSON	http://api.ispeech.org/api/json

Request Parameters

PARAMETER	DATA TYPE	EXAMPLE VALUE
Apikey	32 character hex integer	abcdef1234567890abcdef1234567890
Action	String	convert, ssml
Text	String	Hello World
Ssml (optional)	String	<?xml version="1.0" ?><speak version= [...]
Voice (optional)	String	usenglishfemale (default)
Format (optional)	String	mp3 (default)
Frequency (optional)	Integer (hertz)	16000 (default)
Bitrate (optional)	Integer (kbps)	48 (default)
Speed (optional)	Integer	-10 to 10 (default: 0)
Startpadding (optional)	Integer (seconds)	5 (default: 0)
Endpadding (optional)	Integer (seconds)	5 (default: 0)

Pitch (optional)	Integer	0 to 200 (default: 100)
Bitdepth (optional)	Integer (bits per sample)	8 or 16 (default: 16)
Filename (optional)	String	myaudio or myaudio.mp3 (default: rest.*)
Library (optional)	String	libmath

Example HTTP GET Request (Using most variables)
<code>http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey&action=convert&text=something&voice=usenglishfemale&format=mp3&frequency=44100&bitrate=128&speed=1&startpadding=1&endpadding=1&pitch=110&filename=myaudiofile</code>

Voices

Standard Voices

Name	Alias
US English Female (default)	usenglishfemale
US English Male	usenglishmale
UK English Female	ukenglishfemale
UK English Male	ukenglishmale
Australian English Female	auenglishfemale
US Spanish Female	usspanishfemale
US Spanish Male	usspanishmale
Chinese Female	chchinesefemale
Chinese Male	chchinesemale
Hong Kong Cantonese Female	hkchinesefemale
Taiwan Chinese Female	twchinesefemale
Japanese Female	jpjapanesefemale
Japanese Male	jpjapanesemale
Korean Female	krkoreanfemale

Korean Male	krkoreanmale
Canadian English Female	caenglishfemale
Hungarian Female	huhungarianfemale
Brazilian Portuguese Female	brportugueseefemale
European Portuguese Female	eurportugueseefemale
European Portuguese Male	eurportuguesemale
European Spanish Female	eurspanishfemale
European Spanish Male	eurspanishmale
European Catalan Female	eurcatalanfemale
European Czech Female	eurczechfemale
European Danish Female	eurdanishfemale
European Finnish Female	eurfinnishfemale
European French Female	eurfrenchfemale
European French Male	eurfrenchmale
European Norwegian Female	eurnorwegianfemale
European Dutch Female	eurdutchfemale
European Polish Female	eurpolishfemale
European Italian Female	euritalianfemale
European Italian Male	euritalianmale
European Turkish Female	euturkishfemale
European Turkish Male	euturkishmale
European German Female	eurgermanfemale
European German Male	eurgermanmale
Russian Female	ruussianfemale
Russian Male	ruussianmale
Swedish Female	swswedishfemale
Canadian French Female	cafrenchfemale
Canadian French Male	cafrenchmale

HTTP GET Request (Setting voice to European French Female)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&format=mp3&voice=eurfrenchfemale
```

Custom Voices

Custom Voices may be enabled for your account. They can be found in the developer portal -> api key properties -> custom voices. You can use them by setting the variable voice to the custom alias.

Name	Alias
President Obama	Obama (voice=obama)
Custom Voice	customvoice1 (voice=customvoice1)

Voice List Retrieval

A current list of voices that are enabled for an API key can be retrieved in REST, JSON, and XML format by using the following service. HTTP GET and POST are supported. A web browser or a REST client can be used to make these HTTP requests.

HTTP GET Network Transaction to get XML voice list.

HTTP GET Request and XML Response

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=information&output=xml
```

```
<?xml version='1.0'?>
<data>
  <result>success</result>
  <voice-1>krkoreanfemale</voice-1>
  <voice-locale-1-1>ko-kr</voice-locale-1-1>
  <voice-locale-1-2>ko</voice-locale-1-2>
  <voice-gender-1>female</voice-gender-1>
  <voice-description-1>Korean Female Voice</voice-description-1>
  <voice-2>usenglishfemale</voice-2>
  <voice-locale-2-1>en-us</voice-locale-2-1>
  <voice-locale-2-2>en</voice-locale-2-2>
  <voice-gender-2>female</voice-gender-2>
  <voice-description-2>United States English Female Voice</voice-
description-2>
  [... more voices ...]
</data>
```

Last revised: 3/13/2013

Copyright © 2007 – 2013 iSpeech Incorporated
<http://www.ispeech.org/developers/>

Page 15

HTTP GET Network Transaction to get JSON voice list.

HTTP GET URL Encoded Request and JSON Response
<code>http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey&action=information&output=json</code>
<code>{"voice-gender-48":"female","voice-locale-22-1":"fr-ca","voice-locale-8-1":"pt-br","voice-description-2":"Finnish Female Voice","voice-description-3":"Hong Kong Chinese Male Voice","voice-58":"eurdanishfemale","voice-description-1":"Korean Female Voice","voice-description-6":"Chinese Female Voice","voice-description-7":"United Kingdom English Female Voice","voice-description-4", [...more voices...]}</code>

HTTP GET Network Transaction to get URL Encoded voice list.

HTTP GET URL Encoded Request and URL Encoded Response
<code>http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey&action=information&output=rest</code>
<code>result=success&voice-1=krkoreanfemale2&voice-locale-1-1=ko-kr&voice-gender-1=female&voice-description-1=Korean+Female+Voice&voice-2=eurfinnishfemale&voice-locale-2-1=fi-fi&voice-gender-2=female&voice-description-2=Finnish+Female+Voice&voice-3=chchinesemale1&voice-locale-3-1=zh&voice-locale-3-2=zh-hk[...more voices...]</code>

Speed

Most voices support speed controls.

Speed	Value (integer)
Fastest	10
Faster	Speed > 0
Normal (default)	0
Slower	Speed < 0
Slowest	-10

HTTP GET Request (Setting speed to 5)

`http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&format=mp3&speed=5`

Bitrates

Note: Bitrates can only be selected for MP3s.

Valid values are 16, 24, 32, 48 (default), 56, 64, 80, 96, 112, 128, 144, 160, 192, 224, 256, or 320.

Bitrates are listed in kilobits per second.

HTTP GET Request (Setting bitrate to 16 kilobits per second)

`http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&format=mp3&bitrate=16`

Formats

Name	File extension
Audio Interchange File Format	aiff
MPEG Layer 3 (default)	mp3
Ogg Vorbis	ogg
Windows Media Audio	wma
Free Lossless Audio Codec	flac
Wave PCM	wav
Wave (alaw)	alaw
Wave (μ -law)	ulaw
Dialogic ADPCM	vox
MPEG-4	mp4

Example HTTP GET Request (Setting format to wav)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&format=wav
```

Frequencies

Possible values: 8000, 11025, 16000 (default), 22050, 24000, 32000, 44100, 48000 cycles per second (Hertz)

Example HTTP GET Request (Setting frequency to 16000 Hz)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&frequency=16000
```

Padding

Padding adds silence to a section of the audio file.

Start Padding

Adds a period of silence to the beginning of the audio file.

Example HTTP GET Request (Setting start padding to 3 seconds)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&startpadding=3
```

End Padding

Adds a period of silence to the beginning of the audio file.

Example HTTP GET Request (Setting end padding to 3 seconds)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&endpadding=3
```

Pitch

Possible values: 0 to 200 (integer), 0 is lowest pitch, 100 is default, 200 is highest pitch. Pitch is enabled only on some voices.

Example HTTP GET Request (Setting pitch to 50)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&pitch=50
```

Bit Depth

The bit depth is amount of audio detail for each audio sample.

Possible values are 8 and 16 (default) bits/sample on AIFF, FLAC, and WAVE file formats.

Example HTTP GET Request (Setting bit depth to 8)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&format=wav
&bitdepth=8
```

Filename

The filename is the name of the audio file that will download. Specifying the extension is optional. If the extension is missing, the correct extension will be added automatically. The default is rest.[extension], for example: rest.mp3.

Example HTTP GET Request (Setting filename of audio)

```
http://api.ispeech.org/api/rest?apikey=developerdemokeydeveloperdemokey
&action=convert&text=something&voice=usenglishfemale&format=wav
&filename=myaudiofile
```

Speech Synthesis Markup Language (SSML)

SSML tags are used to customize the way a text-to-speech engine creates audio. The tags can be used to add pauses, change emphasis, and change pronunciation. This option is disabled by default but can be requested by emailing sales@ispeech.org.

The parameter “action” must set to “ssml” and the parameter “ssml” must be set to a complete SSML XML statement. The parameter “text” is not used and the parameters voice and speed should be represented using the “voice” and “prosody” SSML tags instead of request parameters.

Example HTTP GET Request (Emphasis added on the word big)

```
http://api.ispeech.org/api/rest?apikey=YOUR_API_KEY_HERE&
action=ssml&ssml=%3C%3Fxml%20version%3D%221.0%22%3F%3E%3Cspeak
%20version%3D%221.0%22%20xmlns%3D%22http%3A%2F%2Fwww.w3.org
%2F2001%2F10%2Fsynthesis%22%20xmlns%3Aksi%3D%22http%3A%2F%2F
www.w3.org%2F2001%2FXMLSchema-instance%22
%20ksi%3ASchemaLocation%3D%22http%3A%2F%2Fwww.w3.org%2F2001
%2F10%2Fsynthesis%20http%3A%2F%2Fwww.w3.org%2FTR%2Fspeech-synthesis
%2Fsynthesis.xsd%22%20xml%3Alang%3D%22en-US%22%3E
That%20is%20a%20%3Cemphasis%3E%20big%20%3C%2Femphasis%3E%20car!
%3C%2Fspeak%3E
```

SSML used:

```
<?xml version="1.0"?>
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
      http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
      xml:lang="en-US">
  That is a <emphasis>big </emphasis> car!
</speak>
```

Audio synthesized as, "That is a BIG car!".

More information on SSML can be found at: <http://www.w3.org/TR/speech-synthesis/>.

Math Markup Language (MathML)

MathML tags are used to display and represent mathematical statements. This option is disabled by default but can be requested by emailing sales@ispeech.org.

Remember to set “library” to “libmath” so that the MathML processor loads your text as MathML.

Example HTTP GET Request (Setting library to libmath)
<pre>http://api.ispeech.org/api/rest?action=convert &apikey=YOUR_API_KEY_HERE&library=libmath &text=%3Cfmath%3E%3Cmrow%3E%3Cmtext%3E%3Csup%3E%2B%3C%2Fsup%3E%3C%2Fmtext%3E %3Cmn%3E7%3C%2Fmn%3E%3C%2Fmrow%3E%3C%2Ffmath%3E</pre>
<p>MathML used: <code><mrow><mtext><sup>+</sup></mtext><mn>7</mn></mrow></code></p> <p>Audio synthesized as, "positive 7" instead of plus 7.</p>

More information on MathML can be found at: <https://developer.mozilla.org/en-US/docs/MathML>

The following table lists MathML tags supported by the iSpeech API.

MathML Tag	Purpose
mfrac	Used to display fractions
mi	Rendered as an identifier such as function names, variables or symbolic constants.
mn	A numeric literal which is normally a sequence of digits with a possible separator (a dot or a comma). However, it is also allowed to have arbitrary text in it which is actually a numeric quantity, for example "eleven".
mo	An operator in a broad sense. Besides operators in strict mathematical meaning, this element also includes "operators" like parentheses, separators like comma and semicolon, or "absolute value" bars.
mroot	Displays roots with an explicit index. Two arguments are accepted, which leads to the syntax: <code><mroot> base index </mroot></code> .
mrow	Groups sub-expressions, which usually contain one or more operators with their respective operands (such as <code><mi></code> and <code><mn></code>). This element renders as a horizontal row containing its arguments.
msqrt	Displays square roots (no index is displayed). The square root accepts only one argument, which leads to the following syntax: <code><msqrt> base </msqrt></code> .
msup, sup	Attaches a superscript to an expression.
mtext	Renders arbitrary text with <i>no</i> notational meaning, such as comments or annotations.
mspan, span	Used for highlighting text or just general styling of an equation
mtable, table	Creates tables or matrices. Inside a <code><mtable></code> , only <code><mtr></code> and <code><mtd></code> elements may appear. These elements are similar to <code><table></code> , <code><tr></code> and <code><td></code> elements of HTML.

Example Transactions

Summary

The following examples are packet captures from TCP connections that used the HTTP protocol. You can compare your network traffic to these transactions to debug code. Wireshark can be used to analyze network connections. A REST client can be used to make these HTTP requests.

HTTP POST URL encoded request for Text to Speech

HTTP POST Request and Reply
<pre>POST /api/rest HTTP/1.1 Content-Length: 71 Content-Type: text/plain; charset=UTF-8 Host: api.ispeech.org Connection: Keep-Alive apikey=developerdemokeydeveloperdemokey&action=convert&text=hello+world</pre>
<pre>HTTP/1.0 200 OK Connection: close Server: iSpeech Cloud/1.2 Accept-Ranges: none X-Time-Length: 3853 X-Content-Hash: e969ef3dd0dc0e9c417f31f7ffbd10ed Content-Length: 23760 Content-Type: audio/mpeg Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-revalidate, no-transform Pragma: no-cache [mp3 binary audio data]</pre>

HTTP POST JSON request for Text to Speech

HTTP POST, JSON Request and Reply
<pre>POST /api/json HTTP/1.1 Content-Length: 11 Content-Type: application/json; charset=UTF-8 Host: api.ispeech.org Connection: Keep-Alive {"apikey":"developerdemokeydeveloperdemokey","action":"convert","text":"hello world","voice":"usenglishfemale"}</pre>

```
Connection: close
Server: iSpeech Cloud/1.2
Accept-Ranges: none
X-Time-Length: 3853
X-Content-Hash: e969ef3dd0dc0e9c417f31f7ffbd10ed
Content-Length: 23760
Content-Type: audio/mpeg
Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-
revalidate, no-transform
Pragma: no-cache

[mp3 audio binary data]
```

HTTP POST XML request for Text to Speech

HTTP POST, XML Request and Reply

```
POST /api/xml HTTP/1.1
Content-Length: 150
Content-Type: application/xml; charset=UTF-8
Host: api.ispeech.org
Connection: Keep-Alive

<data>
<apikey>developerdemokeydeveloperdemokey</apikey>
<action>convert</action>
<text>hello world</text>
<voice>usenglishfemale</voice>
</data>
```

```
HTTP/1.0 200 OK
Connection: close
Server: iSpeech Cloud/1.2
Accept-Ranges: none
X-Time-Length: 3853
X-Content-Hash: 4affe15913fccd851ebf08a7e2650955
Content-Length: 23760
Content-Type: audio/mpeg
Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-
revalidate, no-transform
Pragma: no-cache

[mp3 audio binary data]
```

Example of a text-to-speech network transaction with an error

Responses with an error message return HTTP status response code “HTTP/1.0 202 Accepted”.

HTTP GET, URL Encoded Request and Reply (Error: misspelled voice)
<pre>GET /api/rest?apikey=developerdemokeydeveloperdemokey&action=convert& text=something&output=rest&voice=usenglishfemal HTTP/1.1 Host: api.ispeech.org Connection: keep-alive User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.58 Safari/535.2 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate,sdch Accept-Language: en-US,en;q=0.8 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3</pre>
<pre>HTTP/1.0 202 Accepted Server: iSpeech Cloud/1.2 Connection: close Content-Length: 41 Content-Type: text/plain Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-revalidate, no-transform Pragma: no-cache result=error&code=8&message=Invalid+voice</pre>

Example network transactions containing MathML

Summary

The following examples are packet captures from TCP connections that used the HTTP protocol. You can compare your network traffic to these transactions to debug code. Wireshark can be used to analyze network connections. A REST client can be used to make these HTTP requests.

More information on MathML is available on <http://www.w3.org/TR/MathML2/> and <https://developer.mozilla.org/en-US/docs/MathML>

HTTP GET URL encoded Text-to-Speech request containing MathML

Request information
Text: "+7" (says, positive seven)
MathML used: $\langle \text{mrow} \rangle \langle \text{mtext} \rangle \langle \text{sup} \rangle + \langle \text{sup} \rangle \langle \text{mtext} \rangle \langle \text{mn} \rangle 7 \langle \text{mn} \rangle \langle \text{mrow} \rangle$

HTTP GET, URL Encoded Request and Reply, +7 (says positive 7)
<pre>GET /api/rest?action=convert&voice=usenglishfemale &apikey=YOUR_API_KEY_HERE&library=libmath&format=mp3 &text=%3Cmath%3E%3Cmrow%3E%3Cmtext%3E%3Csup%3E%2B%3C%2Fsup%3E%3C%2Fmtext%3E %3Cmn%3E7%3C%2Fmn%3E%3C%2Fmrow%3E%3C%2Fmath%3E HTTP/1.1 Host: api.ispeech.org Connection: keep-alive Cache-Control: no-cache Pragma: no-cache Accept-Encoding: identity;q=1, *;q=0 Accept: */* Accept-Language: en-US,en;q=0.8 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3</pre>
<pre>HTTP/1.0 200 OK Connection: close Access-Control-Allow-Origin: * Server: iSpeech Cloud/1.2 Accept-Ranges: none X-Time-Length: 927 X-Content-Hash: 7d125717aa5ac91de5b2b9f2ad6de873 Content-Length: 6264 Content-Type: audio/mpeg Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy- revalidate, no-transform Pragma: no-cache [...mp3 binary data...]</pre>

HTTP POST JSON Text-to-Speech request containing MathML

Request Information
Text: "-3/5" (says, negative three fifths)
MathML used: <pre><mrow> <mtext><sup>-</sup></mtext> <table><tbody> <tr><td class="fm-num-frac fm-inline"><mn>3</mn></td></tr> <tr><td class="fm-den-frac fm-inline"><mn>5</mn></td></tr> </tbody></table> </mrow></pre>

HTTP POST JSON Request and Reply, -3/5 (says negative three fifths)
<pre>POST /api/json HTTP/1.1 Content-Length: 381 Content-Type: text/plain; charset=UTF-8 Host: api.ispeech.org Connection: Keep-Alive Expect: 100-Continue {"apikey":"YOUR_API_KEY_HERE", "action":"convert", "text":"<mrow><mtext><sup>-</sup></mtext> <table><tbody> <tr><td class=\"fm-num-frac fm-inline\"><mn>3</mn></td></tr> <tr><td class=\"fm-den-frac fm-inline\"><mn>5</mn></td></tr> </tbody></table> </mrow>", "voice":"usenglishfemale", "library":"libmath"}</pre>
<pre>HTTP/1.0 200 OK Connection: close Access-Control-Allow-Origin: * Server: iSpeech Cloud/1.2 Accept-Ranges: none X-Time-Length: 1438 X-Content-Hash: 01338a58a7006a20e2b83e5536b34d3a Content-Length: 9288 Content-Type: audio/mpeg Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy- revalidate, no-transform Pragma: no-cache [...mp3 binary data...]</pre>

Last revised: 3/13/2013

Copyright © 2007 – 2013 iSpeech Incorporated
<http://www.ispeech.org/developers/>

HTTP POST XML Text-to-Speech request containing MathML

The text: "+7" gets spoken as "positive seven".

Request Information
Text: "10^3" (says, 10 to the power of 3)
Math ML: $\langle \text{msup} \rangle \langle \text{mn} \rangle 10 \langle \text{mn} \rangle \langle \text{span class="fm-script fm-inline"} \rangle \langle \text{mn} \rangle 3 \langle \text{mn} \rangle \langle \text{span} \rangle \langle \text{msup} \rangle$

HTTP POST XML Request and Reply, 10^3 (says 10 to the power of 3)
<pre>POST /api/xml HTTP/1.1 Content-Length: 294 Content-Type: text/plain; charset=UTF-8 Host: api.ispeech.org Connection: Keep-Alive Expect: 100-Continue <data> <apikey>YOUR_API_KEY_HERE</apikey> <action>convert</action> <text>&lt;msup&gt;&lt;mn&gt;10&lt;/mn&gt;&lt;span class=&quot;fm-script fm- inline&quot;&gt;&lt;mn&gt;3&lt;/mn&gt;&lt;/span&gt;&lt;/msup&gt;</text> <library>libmath</library> <voice>usenglishfemale</voice> </data></pre>
<pre>HTTP/1.0 200 OK Connection: close Access-Control-Allow-Origin: * Server: iSpeech Cloud/1.2 Accept-Ranges: none X-Time-Length: 1356 X-Content-Hash: e563c115ba8c4c3e20b57fff4f4bae5e Content-Length: 8856 Content-Type: audio/mpeg Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy- revalidate, no-transform Pragma: no-cache [...mp3 binary data...]</pre>

Error codes for Text-to-Speech and General Errors

Code	Summary
1	Invalid API key
2	Could not convert text
3	Not enough credits
4	No action specified
5	Invalid text
6	Too many words
7	Invalid text entry
8	Invalid voice
12	Invalid file format
13	Invalid speed
14	Invalid dictionary
15	Invalid bitrate
16	Invalid frequency
30	Option not enabled for your account. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
32	Invalid pitch
100	This evaluation account has exceeded its trial period. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to upgrade your license.
101	Your key has been disabled. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
997	No api access
998	Unsupported output type
999	Invalid request
1000	Invalid Request Method POST Required
3000	SSML error

Section 3

Automated Speech Recognition

Transaction Types and URL Formats

There are currently three transaction types available for use with the iSpeech API. All transactions must use the appropriate URL.

TRANSACTION TYPE	INPUT TYPE	URL FORMAT
HTTP GET/POST	URL Encoded	http://api.ispeech.org/api/rest
HTTP GET/POST	XML	http://api.ispeech.org/api/xml
HTTP GET/POST	JSON	http://api.ispeech.org/api/json

Request Parameters

PARAMETER	VALUE	EXAMPLE
Apikey	32 character hex integer	abcdef1234567890abcdef1234567890
Locale	String	en-US
Action	String	recognize
Content-Type	String	audio/x-wav, audio/amr, audio/speex
Audio	String (base64, remove \r\n)	UkIGRgAKAQBsl/j2+sa0dR [...]
Output (optional)	String	xml, json, rest (default: rest)
Freeform (optional)	Integer (0 to 7)	3
Model (optional)	String	assistant
Speexmode (optional)	Integer (1 to 3)	1 (speex codec only)

Locales

Standard Locales

Name	Alias	Support
English (United States)	locale=en-US (default)	freeform & command list
English (Canada)	locale=en-CA	freeform & command list
English (United Kingdom)	locale=en-GB	freeform & command list
English (Australia)	locale=en-AU	freeform & command list
Spanish (Spain)	locale=es-ES	freeform & command list
Spanish (Mexico)	locale=es-MX	freeform & command list
Italian (Italy)	locale=it-IT	freeform & command list
French (France)	locale=fr-FR	freeform & command list
French (Canada)	locale=fr-CA	freeform & command list
Polish (Poland)	locale=pl-PL	freeform & command list
Portuguese (Portugal)	locale=pt-PT	freeform & command list
Catalan (Catalan)	locale=ca-ES	command list
Chinese (Taiwan)	locale=zh-TW	command list
Danish (Denmark)	locale=da-DK	command list
German (Germany)	locale=fr-FR	command list
Finnish (Finland)	locale=it-IT	command list
Japanese (Japan)	locale=ja-JP	command list
Korean (Korea)	locale=ko-KR	command list
Dutch (Netherlands)	locale=nl-NL	command list
Norwegian (Norway)	locale=nb-NO	command list
Portuguese (Brazil)	locale=pt-BR	command list
Russian (Russia)	locale=ru-RU	command list
Swedish (Sweden)	locale=sv-SE	command list
Chinese (People's Republic of China)	locale=zh-CN	command list

Last revised: 3/13/2013

Copyright © 2007 – 2013 iSpeech Incorporated
<http://www.ispeech.org/developers/>

Chinese (Hong Kong S.A.R.)	locale=zh-HK	command list
----------------------------	--------------	--------------

Custom Locales

Contact sales@ispeech.org for details.

Speech Recognition Models

Statistical speech recognition models are used to increase the probability of a correct result. Models with fewer word choices are faster and more accurate than the freeform models. For example, in the food model the words, “7 up” would be recognized as, “7up”. Another example is with a food model would recognize the audio from “ice cream” as “ice cream” instead of “i scream”.

Standard Freeform Models

Name	Value	Use Case
SMS / General Messages	freeform=1	Text Messages
Voice mail	freeform=2	Voice Mail
Dictation	freeform=3	Normal speech
Message (coming soon)	freeform=4	Email
Instant Message (coming soon)	freeform=5	Instant Message
Transcript (coming soon)	freeform=6	
Memo (coming soon)	freeform=7	Memorandum
Web (coming soon)	freeform=8	Web search

Standard Non-Freeform Models

Name	Value	Use Case
Assistant	model=assistant	Personal Assistant
Date	model=date	Date
NFL	model=nfl	Football teams
NBA	model=nba	Basketball teams
US Money	model=usmoney	US Money
Numbers to 9	model=numbersto9	Numbers, 0 to 9
Numbers to 99	model=numbersto99	Numbers, 0 to 99
Numbers to 999	model=numbersto999	Numbers, 0 to 999
Time	model=time	Time
Phone number	model=phonenumbers	Phone number
Streets	model=streets	Streets
Sports Team	model=sportsteam	Sports Teams
City/State	model=citystate	US City/States

Custom Models

Contact sales@ispeech.org for details.

Speex Modes

The speexmode variable tells the server which format your Speex data is encoded in for improved speech recognition quality. It is highly recommended you include this parameter when using Speex encoding.

Name	Value
Narrowband (8khz)	speexmode=1
Wideband (16khz) - Recommended	speexmode=2

Example Transactions for Freeform Speech

Format of Examples

The following examples are packet captures from TCP connections that used the HTTP protocol. You can compare your network traffic to these transactions to debug code. Wireshark can be used to analyze network connections.

HTTP REST transaction for Speech Recognition

HTTP REST Request and Response

```
POST /api/rest HTTP/1.1
Content-Length: 34875
Content-Type: text/plain; charset=UTF-8
Host: api.ispeech.org
Connection: Keep-Alive

apikey=developerdemokeydeveloperdemokey&action=recognize&freeform=1&content-
type=audio/x-wav&output=rest&locale=en-us&audio=[base64 encoded something.wav
without \r\n characters]

HTTP/1.0 200 OK
Connection: close
Content-Length: 59
Content-Type: text/plain
Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-
revalidate, no-transform
Pragma: no-cache

text=something&confidence=0.0216270890086889&result=success
```

HTTP JSON transaction for Speech Recognition

HTTP JSON Request and REST Reply

```
POST /api/json HTTP/1.1
Content-Length: 34897
Content-Type: text/plain; charset=UTF-8
Host: api.ispeech.org
Connection: Keep-Alive

{"apikey":"developerdemokeydeveloperdemokey","action":"recognize",
"freeform":"1", "locale":"en-us", "content-type":"audio/x-wav",
"output":"rest", "audio":"[base64 encoded something.wav without \r\n
```

```
characters"]"}}
```

```
HTTP/1.0 200 OK
Connection: close
Content-Length: 59
Content-Type: application/json
Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-
revalidate, no-transform
Pragma: no-cache

text=something&confidence=0.0134419081732631&result=success
```

HTTP XML network transaction for Speech Recognition

HTTP XML Request and Reply

```
POST /api/xml HTTP/1.1
Content-Length: 34953
Content-Type: text/plain; charset=UTF-8
Host: api.ispeech.org
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.0.1 (java 1.5)
Expect: 100-Continue

<data>
<apikey>developerdemokeydeveloperdemokey</apikey>
<action>recognize</action>
<freeform>1</freeform>
<locale>en-us</locale>
<content-type>audio/x-wav</content-type>
<output>xml</output>
<audio>[base64 encoded something.wav without \r\n characters]</audio>
</data>
```

```
HTTP/1.0 200 OK
Connection: close
Content-Length: 140
Content-Type: text/xml
Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-
revalidate, no-transform
Pragma: no-cache

<?xml version="1.0" encoding="UTF-8"?>
<data>
<text>something</text>
<confidence>0.0216270890086889</confidence>
<result>success</result>
</data>
```

Command Lists

Command lists are used to limit the possible values returned during speech recognition. For example, if the command list contains only “yes” and “no”, the result will be either “yes” or “no”.

Example Transactions for Command Lists

Formatting of Examples

The following examples are packet captures of TCP connections that use the HTTP protocol. You can compare your network traffic with these transactions to debug code. Wireshark can be used to analyze network connections. A REST client can be used to make these HTTP requests.

HTTP XML network transaction to detect commands from a list

If a user speaks "say yes", "say maybe", or "say no" it will be successfully recognized.

HTTP XML Request and Response
<pre>POST /api/xml HTTP/1.1 Content-Length: 80941 Content-Type: text/xml; charset=UTF-8 Host: api.ispeech.org Expect: 100-Continue <data> <apikey>developerdemokeydeveloperdemokey</apikey> <action>recognize</action> <locale>en-US</locale> <output>xml</output> <alias>command1 YESNOMAYBE</alias> <YESNOMAYBE>yes no maybe</YESNOMAYBE> <command1>say %YESNOMAYBE%</command1> <content-type>audio/x-wav</content-type> <audio>[base64 encoded say_yes.wav without \r\n characters]</audio> </data></pre>
<pre>HTTP/1.0 200 OK Connection: close Content-Length: 137 Content-Type: text/xml Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy- revalidate, no-transform Pragma: no-cache <?xml version="1.0" encoding="UTF-8"?> <data> <text>say yes</text> <confidence>0.726751327514648</confidence> <result>success</result> </data></pre>

HTTP REST network transaction to detect commands from a list

If a user speaks "call john", "call anna", or "call mary" it will be successfully recognized.

HTTP REST Request and Response
<pre>POST /api/rest/ HTTP/1.1 Content-Length: 72682 Content-Type: text/plain; charset=UTF-8 Host: api.ispeech.org Expect: 100-Continue apikey=developerdemokeydeveloperdemokey&action=recognize&locale=en- us&content-type=audio%2Fwav&output=rest&alias=command1 NAMES &NAMES=john mary anna&command1=call%20%25NAMES%25&audio=[base64 encoded wav without \r\n characters]</pre>
<pre>HTTP/1.0 200 OK Connection: close Content-Length: 58 Content-Type: text/plain Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy- revalidate, no-transform Pragma: no-cache text=call+mary&confidence=0.672464966773987&result=success</pre>

HTTP POST JSON request to detect commands from a list

If a user speaks "say yes", "say maybe", or "say no" it will be successfully recognized.

HTTP POST JSON Request and REST Response
<pre>POST /api/json/ HTTP/1.1 Content-Length: 22788 Content-Type: text/plain; charset=UTF-8 Host: api.ispeech.org Expect: 100-Continue {"apikey":"developerdemokeydeveloperdemokey","action":"recognize","locale":"en-US","alias":"command1 YESNOMAYBE","YESNOMAYBE":"yes no maybe","command1":"say %YESNOMAYBE%","content-type":"audio/x-wav","output":"rest","audio":"[base64 encoded say_yes.wav without \r\n characters]"} HTTP/1.0 200 OK Connection: close Content-Length: 56 Content-Type: application/json Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-revalidate, no-transform Pragma: no-cache text=say+yes&confidence=0.726751327514648&result=success</pre>

Advanced Example, HTTP POST XML request to detect multiple audio commands from multiple lists

If a user speaks "set monitor on", "set monitor off", or "set dynamite explode", etc. it will be successfully recognized.

HTTP XML Request and Response
<pre>POST /api/xml HTTP/1.1 Content-Length: 91393 Content-Type: text/xml; charset=UTF-8 Host: api.ispeech.org Connection: Keep-Alive Expect: 100-Continue <data> <apikey>developerdemokeydeveloperdemokey</apikey> <action>recognize</action> <locale>en-us</locale> <content-type>audio/x-wav</content-type> <output>xml</output> <alias>command1 command2 MONITORACTIONS COLORLIST DYNAMITEACTIONS OBJECTLIST</alias> <MONITORACTIONS>on off reset</MONITORACTIONS> <COLORLIST>blue green red yellow purple orange black white cyan</COLORLIST> <DYNAMITEACTIONS>explode fizzle out</DYNAMITEACTIONS> <OBJECTLIST>monitor %MONITORACTIONS% color %COLORLIST% dynamite %DYNAMITEACTIONS%</OBJECTLIST> <command1>set %OBJECTLIST%</command1> <command2>quit menu</command2> <audio>[base64 encoded set_dynamite_explode.wav without \r\n characters]</audio> </data></pre>
<pre>HTTP/1.0 200 OK Connection: close Content-Length: 150 Content-Type: text/xml Cache-Control: no-cache, no-store, must-revalidate, max-age=0, proxy-revalidate, no-transform Pragma: no-cache <?xml version="1.0" encoding="UTF-8"?><data><text>set dynamite explode</text><confidence>0.589247465133667</confidence> <result>success</result></data></pre>

Error Codes for Speech Recognition and General Errors

Code	Summary
1	Invalid API key
3	Not enough credits
4	No action specified
12	Invalid file format
14	Invalid dictionary
17	Invalid alias list
18	Alias missing
19	Invalid content type
20	Alias list too complex
21	Could not recognize
23	Invalid locale
24	Bad audio data
25	Model not supported or disabled
26	Selected model does not support desired locale
28	Locale not supported
30	Option not enabled for your account. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
100	This evaluation account has exceeded its trial period. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to upgrade your license.
101	Your key has been disabled. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
997	No api access
998	Unsupported output type
999	Invalid request
1000	Invalid Request Method POST Required

Section 4

Position Markers

Transaction Types and URL Formats

TRANSACTION TYPE	INPUT FORMAT	URL
HTTP GET/POST	URL Encoded	http://api.ispeech.org/api/rest
HTTP GET/POST	XML	http://api.ispeech.org/api/xml
HTTP GET/POST	JSON	http://api.ispeech.org/api/json

Request Parameters

PARAMETER	DATA TYPE	EXAMPLE VALUE
Apikey	32 character hex integer	abcdef1234567890abcdef1234567890
Action	String	markers
Text	String	Hello World
Voice (optional)	String	usenglishfemale (default)
Format (optional)	String	mp3 (default)
Speed (optional)	Integer	-10 to 10 (default: 0)
Startpadding (optional)	Integer (seconds)	5 (default: 0)
Endpadding (optional)	Integer (seconds)	5 (default: 0)

Introduction

Position markers provide information regarding word boundaries to allow applications to visually display the current location in spoken audio. It is similar to how a karaoke system would display lyrics.

This is accomplished by first retrieving audio from the iSpeech API (see section 2 for more details), then making a second request for an XML document which contains word boundary information.

Example Transactions for Position Markers

Note: Marker data is currently only presented in XML form.

To obtain marker information from the iSpeech API, you query the server in the same manner as a normal text-to-speech request. The only difference between a TTS request and a marker request is the “action” parameter, which is set to “convert” for audio, and “markers” for marker information.

HTTP GET network transaction to retrieve position markers

HTTP GET Request and XML Response	
<code>http://api.ispeech.org/api/rest?apikey=YOUR_API_KEY&action=markers &text=hello+world</code>	
<pre><?xml version="1.0" encoding="UTF-8"?> <markers> <text>hello world</text> <voice>usenglishfemale</voice> <length>894</length> <words>2</words> <word> <start>70</start> <end>339</end> <index>1</index> <length>270</length> <text>hello</text> </word> <word> <start>340</start> <end>894</end> <index>2</index> <length>555</length> <text>world</text> </word> </markers></pre>	<pre><- Text sent in request <- Voice parameter sent in request <- Approximate audio length in ms <- Amount of words and frames <- Start offset in milliseconds <- End offset in milliseconds <- Index of frame <- Length of frame <- Word to highlight</pre>

Marker Information Usage Technique

Once you have obtained an audio file and the respective marker information XML document, you are ready to highlight text.

There are many methods to processing iSpeech marker information; the following outlines the most basic of those methods. Use the following steps as a baseline implementation. **Implementations will vary greatly depending on the platform.**

Media Player Considerations

Your media player must support “location”, “position”, or must notify you of its current progress periodically. For example, in Flash, we set a timer to poll for the audio position every 250 milliseconds. Highlighting will be more accurate with a low interval.

Implementation

If your media player supports retrieval of “current position” or similar, you can follow these basic steps:

1. Retrieve audio
2. Retrieve marker information xml
3. Parse xml into enumerable container/object
4. Load audio into media player and start playing
5. Create a timer and set it's interval to 250 milliseconds
6. Inside of the newly created timer, at every interval query the media player's current position
7. Convert the position to milliseconds (if you have number such as 1.343, simply multiply by 1000)
8. Move to first (or next) “word” node inside of marker information xml document
9. Check to see if current position is greater than or equal to the value of “start” AND ALSO current position is less than or equal to the value of “end”, highlight the specified “text”
10. If current position is greater than “word” “end” value go to step 8

You can follow the above steps until the audio file is exhausted.

Notes

Due to the overhead of creation, we **strongly** recommend using short sentences instead of large blocks of text when requesting marker information.

The same parameters must be sent in the markers request as the original TTS audio request. For example, if you pass a “speed” parameter during audio conversion, you must also send this parameter in your marker information request. If you fail to do so, the marker information will not line up correctly.

File type affects audio length. A MP3 file is always longer than a WAV file due to compression padding. The API will modify the file length accordingly.

Error codes for Position Markers and General Errors

Code	Summary
1	Invalid API key
2	Could not convert text
3	Not enough credits
4	No action specified
5	Invalid text
6	Too many words
7	Invalid text entry
8	Invalid voice
12	Invalid file format
13	Invalid speed
14	Invalid dictionary
15	Invalid bitrate
16	Invalid frequency
30	Option not enabled for your account. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
33	Invalid text or markers not supported for the selected voice
34	Markers do not support audio padding or other option.
100	This evaluation account has exceeded its trial period. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to upgrade your license.
101	Your key has been disabled. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
997	No api access
998	Unsupported output type
999	Invalid request
1000	Invalid Request Method POST Required

Section 5

Visemes

Transaction Types and URL Formats

TRANSACTION TYPE	INPUT FORMAT	URL
HTTP GET/POST	URL Encoded	http://api.ispeech.org/api/rest
HTTP GET/POST	XML	http://api.ispeech.org/api/xml
HTTP GET/POST	JSON	http://api.ispeech.org/api/json

Request Parameters

PARAMETER	DATA TYPE	EXAMPLE VALUE
Apikey	32 character hex integer	abcdef1234567890abcdef1234567890
Action	String	viseme
Text	String	Hello World
Voice (optional)	String	usenglishfemale (default)
Format (optional)	String	mp3 (default)
Speed (optional)	Integer	-10 to 10 (default: 0)
Startpadding (optional)	Integer (seconds)	5 (default: 0)
Endpadding (optional)	Integer (seconds)	5 (default: 0)

Introduction

Visemes provide information regarding the mouth position and time interval of spoken audio which allows applications to visually pronounce audio.

This is accomplished by first retrieving audio from the iSpeech API (see section 2 for more details), then making a second request for an XML document which contains viseme information.

Example Transaction for Viseme Retrieval

Note: Viseme data is currently only presented in XML form.

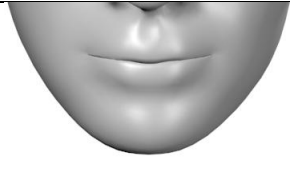
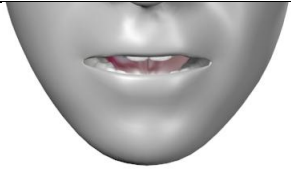
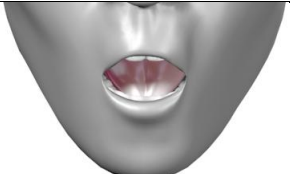



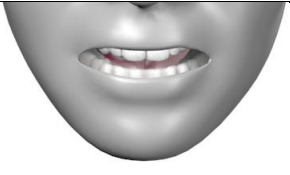


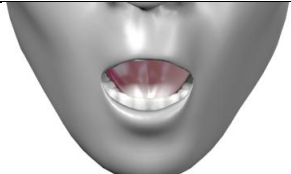
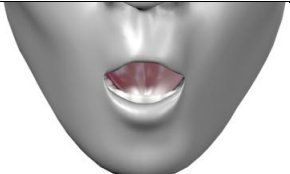




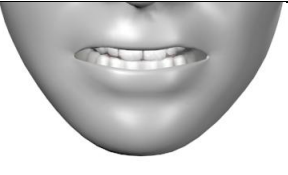


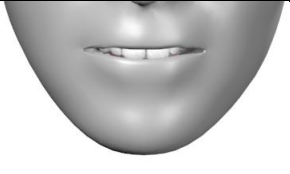


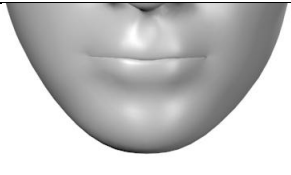
To obtain viseme information from the iSpeech API, you query the server in the same manner as a normal text-to-speech request. The only difference between a TTS request and a marker request is the “action” parameter, which is set to “convert” for audio, and “viseme” for viseme information.

HTTP GET network transaction to retrieve viseme positions

HTTP GET Request and XML Response	
<code>http://api.ispeech.org/api/rest?apikey=YOUR_API_KEY&action=viseme &text=hello+world</code>	
<pre><visemes> <text>hello world</text> <voice>usenglishfemale</voice> <length>753</length> <frames>9</frames> <viseme> <start>1</start> <end>74</end> <index>1</index> <length>74</length> <mouth>12</mouth> </viseme> <viseme> <start>75</start> <end>102</end> <index>2</index> <length>28</length> <mouth>4</mouth> </viseme> <viseme> <start>103</start> <end>182</end> <index>3</index> <length>80</length> <mouth>14</mouth> </visemes></pre>	<pre><- Text sent in request <- Voice parameter sent in request <- Approximate audio length in ms <- Amount of frames <- Start offset in milliseconds <- End offset in milliseconds <- Index of frame <- Length of frame <- Mouth position (0-21 maps to the 1-22 standard)</pre>

<pre> </viseme> <viseme> <start>183</start> <end>269</end> <index>4</index> <length>87</length> <mouth>8</mouth> </viseme> <viseme> <start>270</start> <end>359</end> <index>5</index> <length>90</length> <mouth>7</mouth> </viseme> <viseme> <start>360</start> <end>470</end> <index>6</index> <length>111</length> <mouth>5</mouth> </viseme> <viseme> <start>471</start> <end>660</end> <index>7</index> <length>190</length> <mouth>14</mouth> </viseme> <viseme> <start>661</start> <end>752</end> <index>8</index> <length>92</length> <mouth>19</mouth> </viseme> <viseme> <start>753</start> <end>753</end> <index>9</index> <length>1</length> <mouth>0</mouth> </viseme> </visemes> </pre>	
--	--

Viseme Chart

			
Mouth 0	Mouth 1	Mouth 2	Mouth 3
			
Mouth 4	Mouth 5	Mouth 6	Mouth 7
			
Mouth 8	Mouth 9	Mouth 10	Mouth 11
			
Mouth 12	Mouth 13	Mouth 14	Mouth 15
			
Mouth 16	Mouth 17	Mouth18	Mouth 19
			
Mouth 20	Mouth 21		

Viseme Usage Technique

Once you have obtained an audio file and the respective viseme information XML document, you are ready to animate mouth positions to simulate speaking.

There are many methods to processing iSpeech viseme information; the following outlines the most basic of those methods. Use the following steps as a baseline implementation. **Implementations will vary greatly depending on the platform.**

Media Player Considerations

Your media player must support “location”, “position”, or must notify you of its current progress periodically. For example, in Flash, we set a timer to poll for the audio position every 250 milliseconds. Mouth positioning will be more accurate with a low interval.

Implementation

If your media player supports retrieval of “current position” or similar, you can follow these basic steps:

1. Retrieve audio
2. Retrieve viseme information xml
3. Parse xml into enumerable container/object
4. Load audio into media player and start playing
5. Create a timer and set it's interval to 250 milliseconds
6. Inside of the newly created timer, at every interval query the media player's current position
7. Convert the position to milliseconds (if you have number such as 1.343, simply multiply by 1000)
8. Move to first (or next) “word” node inside of marker information xml document
9. Check to see if current position is greater than or equal to the value of “start” AND ALSO current position is less than or equal to the value of “end”, highlight the specified “text”
10. If current position is greater than “word” “end” value go to step 8

You can follow the above steps until the audio file is exhausted.

Notes

Due to the overhead of creation, we strongly recommend using short sentences instead of large blocks of text when requesting viseme information.

The same parameters must be sent in the viseme request as the original TTS audio request. For example, if you pass a “speed” parameter during audio conversion, you must also send this parameter in your marker information request. If you fail to do so, the viseme will not line up correctly.

File type affects audio length. A MP3 file is always longer than a WAV file due to compression padding. The API will modify the file length accordingly.

Error codes for Visemes and General Errors

Code	Summary
1	Invalid API key
2	Could not convert text
3	Not enough credits
4	No action specified
5	Invalid text
6	Too many words
7	Invalid text entry
8	Invalid voice
12	Invalid file format
13	Invalid speed
14	Invalid dictionary
15	Invalid bitrate
16	Invalid frequency
29	Viseme not supported for the selected voice.
30	Option not enabled for your account. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
32	Invalid pitch
100	This evaluation account has exceeded its trial period. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to upgrade your license.
101	Your key has been disabled. Please contact iSpeech sales at +1 (917) 338-7723 or at sales@ispeech.org to modify your license.
997	No api access
998	Unsupported output type
999	Invalid request
1000	Invalid Request Method POST Required