WebSpeak Web Speech Synthesizer Specification

WebSpeak is a web service enabling a simple integration of synthesised speech into a broad set of clients.

A client, which can be either web site or an application, sends a text to a HTTP server. The server converts the text to speech, and sends it as an audio format back to the client to broadcast.

This document describes the set of functions of WebSpeak, their parameters and returned replies. It is targeting readers with basic knowledge of HTTP protocol.

## Changes

1.0 Initial version

1.1 Extra parameters can be passed using POST method now.

 Basic user credentials (Authorization header) are no longer required.

 speak2 function introduced.

2.0 Client authentication using API key

 Pre-recorded audio managing discontinued

2.1 info function requires API key too

# Table of Contents

[Changes 1](#_Toc43105362)

[Table of Contents 2](#_Toc43105363)

[General Purpose Functions 3](#_Toc43105364)

[*info* Function 3](#_Toc43105365)

[Synthetic Speech Functions 4](#_Toc43105366)

[*speak* Function 4](#_Toc43105367)

[*speak2* Function 4](#_Toc43105368)

[*retrieve* Function 5](#_Toc43105369)

# General Purpose Functions

## info Function

Provides general information about speech server version and server installed voices. The client usually calls this function first, which enables it to display user a list of available voices in its user interface (optional).

### Request

* Parameters:
	+ f=info
	+ k=<API key>
* Method: GET
* Relevant request headers:
	+ Accept-Language: client preferred error description language

### Response

* Status: 200 OK, or an error status
* Type: application/json, or text/plain; charset=UTF-8 in case of error
* Content:

{

"version" : <server version, string>,

"voices" : [

{

"id" : <voice 1 ID, string>,

"name" : <voice 1 name, string>,

"lang" : <language, string, ISO 639-1>,

"gender" : <gender, "m"|"f">,

"age" : <age, "child"|"teen"|"adult"|"senior">

},

{

"id" : <voice 2 ID, string>,

"name" : <voice 2 name, string>,

"lang" : <language, string, ISO 639-1>,

"gender" : <gender, "m"|"f">,

"age" : <age, "child"|"teen"|"adult"|"senior">

}, ... ]

}

# Synthetic Speech Functions

## speak Function

Translates text to speech. When technically feasible, the server starts sending chunks of audio as soon as possible (Chunked transfer HTTP/1.1).

POST method is preferred for longer text.

### Request

* Parameters:
	+ f=speak (always GET)
	+ t=<text to synthesise UTF-8 encoded>
	+ v=<voice ID, one of the voices returned by info function>
	+ o=<response audio format requested: mp3 = MP3, ogg = OGG Vorbis>
	+ k=<API key>
* Method: POST or GET
* Relevant request headers:
	+ Content-Type: application/x-www-form-urlencoded for POST requests
	+ Accept-Language: client preferred error description language
	+ Origin: website URL, if the API key is domain based. It must contain <protocol>://<hostname>[:<port>] only (e.g. https://ebralec.si)

### Response

* Status: 200 OK, or an error status
* Type: audio/mpeg or audio/ogg, depends on the o parameter, or text/plain; charset=UTF-8 in case of error
* Content: audio file in requested format, or error description in the language specified by Accept-Language request header

## speak2 Function

Translates text to speech. This function prepares synthesized audio in the cache and returns its ID. This function may be used when text is too long for GET method, and a direct play is required on a player that does not support POST method and may download audio using GET method only (e.g. HTML5 <audio>).

Audio synthesised using this function is saved on the server for a limited time only. The server is cleaning its cache regularly. For immediate play or persistent audio storage, one should use the speak function, or speak2 function immediately followed by retrieve function.

The speak2 function may be used in a CMS to trigger synthesis of a text in advance to reduce audio delivery for initial listener when text is published. However, one must use the speak2 function on the web site immediately before the retrieve function. This ensures the server will resynthesise the audio if required in case it was cleaned from the cache already.

### Request

* Parameters:
	+ f=speak2 (always GET)
	+ t=<text to synthesise UTF-8 encoded >
	+ v=<voice ID, one of the voices returned by info function>
	+ k=<API key>
* Method: POST or GET
* Relevant request headers:
	+ Content-Type: application/x-www-form-urlencoded for POST requests
	+ Accept-Language: client preferred error description language
	+ Origin: website URL, if the API key is domain based. It must contain <protocol>://<hostname>[:<port>] only (e.g. https://ebralec.si)

### Response

* Status: 200 OK, or an error status
* Type: application/json, or text/plain; charset=UTF-8 in case of error
* Content: string with audio ID for retrieve function, or error description in the language specified by Accept-Language request header.

## retrieve Function

Downloads synthesized audio prepared using speak2 function.

### Request

* Parameters:
	+ f=retrieve (always GET)
	+ i=<audio ID returned by speak2 function>
	+ o=<response audio format requested: mp3 = MP3, ogg = OGG Vorbis>
	+ k=<API key>
* Method: POST or GET
* Relevant request headers:
	+ Content-Type: application/x-www-form-urlencoded for POST requests
	+ Accept-Language: client preferred error description language
	+ Origin: website URL, if the API key is domain based. It must contain <protocol>://<hostname>[:<port>] only (e.g. https://ebralec.si)

### Response

* Status: 200 OK, or an error status
* Type: audio/mpeg or audio/ogg, depends on the o parameter, or text/plain; charset=UTF-8 in case of error
* Content: audio file, or error description in the language specified by Accept-Language request header