SpeechAce API

1.0 Summary

SpeechAce provides an REST style application programming interface (API). It contains the following functions:

1. Score audio of text (section 2.0)
2. Score audio of phone list (section 3.0)
3. Validate text exists in Speechace dictionary (section 4.0)
4. Generate reference audio (section (5.0)

2.0 Score audio of text

Verb and Url

```
POST
https://api.speechace.co/api/scoring/text/v0.1/json?key={develope
r product key}&dialect={dialect to score against}&user_id={user
identifier}
```

Multi-part Post body

```
text={a word, phrase, or sentence}
user_audio_file={wav or mp3 audio file}
```

Description

Use this function to score an audio with respect to a piece of text and a dialect. It is possible that some of the word in the text has multiple pronunciations. For example, read can "r iy d" or "r eh d". If SpeechAce scoring algorithm determines that an audio scores higher with "r eh d", SpeechAce returns the score based on that.

Parameters

Name	Description	Example
text	a word, phase, or sentence.	`That's Mr. Smith, isn't it?`
user_audio_file	a wav or mp3 audio file with sampling rate 16kHz or higher. The file size must be 20 seconds or less.	

Name	Description	Example
dialect	Which dialect should the audio be scored against. Currently, en-us and en-gb are supported.	en-us
user_id	This can be any unique user identifier. Ideally, it should be non personal information of a user.	1234
key	This is a key issued by SpeechAce. If two products are billed together, they share the same product key. If two products are billed separately, they use two different keys.	

Response

When the scoring is successful,

```
{
    "status": "success",
    "quota_remaining": <number or ,-1 if user has unlimited quota>,
    "text_score": {
        "text": <the input text>,
        "quality_score": <floating point number between 0 and 100>,
        "word_score_list": [
            <word score of the first word in the input text>,
            <word score of the second word in the input text>,
            ...
            <word score of the last word in the input text>
        ],
        "word_intonation_list": [
            <word intonation of the first word in the input text>,
            <word intonation of the second word in the input text>,
            …
            <word intonation of the last word in the input text>
        ]
    }
}
```

Each <word score> has the following content:

```
{
    "word": <the input word>,
    "quality_score": <floating point number between 0 and 100>,
    "phone_score_list": [
        <phone score of the first phone in the input word>,
        <phone score of the second phone in the input word>,
```

```
            ...
            <phone score of the last phone in the input word>
    ],
    "syllable_score_list": [
            <syllable score of the first syllable in the input word>,
            <syllable score of the second syllable in the input word>,
            ...
            <syllable score of the last syllable in the input word>
    ]
}
```

Each <phone score> has the following content:

```
{
    "phone": <For en-us or en-gb, it is one of the phones in ARPABET notation>,
    "quality_score": <Overall quality score: floating point number between 0 and 100>,
    "stress_level": <Expected stress level: null for consonant; 0 - unstressed, 1 -
primary, 2 - secondary>,
    "stress_score": <Stress score: Floating point number between 0 and 100>,
    "extent": [
        <start time of the phone in the audio in 10 milliseconds>,
        <end time of the phone in the audio in 10 milliseconds>
    ],
    "sound_most_like": <which phone does the extent of the audio sound most like>,
    "child_phones": <missing if no extra phone is inserted, or it is a list of phone score>
}
```

Each <syllable score> has the following content:

```
{
    "letters": <the letters in the input word which constitutes this syllable>,
    "quality_score": <Overall quality score: floating point number between 0 and 100>,
    "stress_level": <Expected stress level: null for consonant; 0 - unstressed, 1 -
primary, 2 - secondary>,
    "stress_score": <Stress score: Floating point number between 0 and 100>,
    "extent": [
        <start time of the syllable in the audio in 10 milliseconds>,
        <end time of the syllable in the audio in 10 milliseconds>
    ],
    "phone_count": <number of phones in this syllable>
}
```

Each <word intonation> has the following content:

```
{
    "word": <the input word>,
    "syllable_intonation_list": [
        <syllable intonation of the first syllable of the word>,
        <syllable intonation of the second syllable of the word>,
```

```
            …
            <syllable intonation of the last syllable of the word>
        ]
}
```

Each <syllable intonation> has the following content:

```
[
        <secondary intonation information>,
        <primary intonation information>
]
```

<primary intonation information> indicates whether the pitch of a word rises, falls, or more or less unchanged. If we can't recognize the syllable (due to error from our side or due to user not saying the syllable), the value is null. If we can recognize the syllable but couldn't determine the pitch (due to error from our side or due to user reducing the sound to unvoiced), the value is reduced. To summarize, the valid values are: "RISE", "FALL", "FLAT", "REDUCED", null.

<secondary intonation information> is null unless one of the following cases occurs:

- If the pitch of the syllable falls, but the starting pitch of the syllable is higher than the ending pitch of previous syllable, secondary intonation is RISE while primary intonation is FALL.
- If the pitch of the syllable rises, but the starting pitch of the syllable is lower than the ending pitch of the previous syllable, secondary intonation is FALL while primary intonation is RISE.

When the scoring fails,

```
{
    "status": "error",
    "short_message": <a short error message>,
    "detail_message": <a detailed error message>
}
```

Some of the errors are:

1. short message: error_audio_too_long
   detail message: The audio file is too long. It must be 20 seconds or shorter.

2. short message: error_timeout
   detail message: The request time out because the server is too busy.

3. short message: error_unknown_words
   detail message: The server could not identify the following words: <list of unknown words separated by |>

4. short message: error_out_of_quota
   detail message: The user has run out of quota

Example

Request:

```
POST
https://api.speechace.co/api/scoring/text/v0.1/json?key={developer
product key}&dialect=en-us&user_id=1234

    text=That's%20Mr.%20Smith%2C%20isn't%20it%3F
    user_audio_file={wav or mp3 audio file}
```

Response:


```
{
    "quota_remaining": -1,
    "saved_file_path": "e2e_test/That_s_Mr__Smith__isn_t_it_-1/uo3vQOt6XX3XjV8pxaJMiUSAGqR0ovy5B-
JSchhvhO4=.wav",
    "status": "success",
    "text_score": {
        "quality_score": 95.0,
        "text": "That's Mr. Smith, isn't it?",
        "word_intonation_list": [
            {
                "syllable_intonation_list": [
                    [
                        null,
                        "RISE"
                    ]
                ],
                "word": "That's"
            },
            {
                "syllable_intonation_list": [
                    [
                        null,
                        "FALL"
                    ],
                    [
                        null,
                        "REDUCED"
                    ]
                ],
                "word": "Mr."
            },
            {
                "syllable_intonation_list": [
                    [
                        "RISE",
                        "FALL"
                    ]
                ],
                "word": "Smith"
            },
            {
                "syllable_intonation_list": [
                    [
                        "RISE",
                        "FALL"
                    ],
                    [
                        null,
                        "RISE"
                    ]
                ],
                "word": "isn't"
            },
            {
```

```
        "syllable_intonation_list": [
            [
                null,
                "RISE"
            ]
        ],
        "word": "it"
    }
],
"word_score_list": [
    {
        "phone_score_list": [
            {
                "extent": [
                    1,
                    5
                ],
                "phone": "dh",
                "quality_score": 99.0,
                "sound_most_like": "b",
                "stress_level": null
            },
            {
                "child_phones": [
                    {
                        "extent": [
                            5,
                            16
                        ],
                        "quality_score": 90.06060606060606,
                        "sound_most_like": "ae"
                    },
                    {
                        "extent": [
                            16,
                            17
                        ],
                        "quality_score": 41.666666666666664,
                        "sound_most_like": "ah"
                    }
                ],
                "extent": [
                    5,
                    17
                ],
                "phone": "ae",
                "quality_score": 86.02777777777777,
                "sound_most_like": "ah",
                "stress_level": 1,
                "stress_score": 100.0
            },
            {
                "extent": [
                    17,
                    19
                ],
                "phone": "t",
                "quality_score": 90.0,
                "sound_most_like": "ah",
                "stress_level": null
            },
            {
                "extent": [
                    19,
                    31
                ],
                "phone": "s",
                "quality_score": 99.0,
                "sound_most_like": "z",
```

```
            "stress_level": null
        }
    ],
    "quality_score": 94.0,
    "syllable_score_list": [
        {
            "extent": [
                1,
                31
            ],
            "intonation": [
                null,
                "RISE"
            ],
            "letters": "that's",
            "phone_count": 4,
            "pitch_range": [
                152.84,
                203.19
            ],
            "quality_score": 94.0,
            "stress_level": 1,
            "stress_score": 100.0
        }
    ],
    "word": "That's"
},
{
    "phone_score_list": [
        {
            "extent": [
                31,
                39
            ],
            "phone": "m",
            "quality_score": 100.0,
            "sound_most_like": "m",
            "stress_level": null
        },
        {
            "extent": [
                39,
                46
            ],
            "phone": "ih",
            "quality_score": 100.0,
            "sound_most_like": "ih",
            "stress_level": 1,
            "stress_score": 96.75098630772801
        },
        {
            "extent": [
                46,
                51
            ],
            "phone": "s",
            "quality_score": 98.0,
            "sound_most_like": "z",
            "stress_level": null
        },
        {
            "extent": [
                51,
                57
            ],
            "phone": "t",
            "quality_score": 92.5,
            "sound_most_like": "d",
            "stress_level": null
```

```
            },
            {
                "extent": [
                    57,
                    62
                ],
                "phone": "er",
                "quality_score": 53.8,
                "sound_most_like": "ih",
                "stress_level": 0,
                "stress_score": 96.75098630772801
            }
        ],
        "quality_score": 89.0,
        "syllable_score_list": [
            {
                "extent": [
                    31,
                    46
                ],
                "intonation": [
                    null,
                    "FALL"
                ],
                "letters": "mr",
                "phone_count": 2,
                "pitch_range": [
                    102.17,
                    91.8
                ],
                "quality_score": 100.0,
                "stress_level": 1,
                "stress_score": 96.75098630772801
            },
            {
                "extent": [
                    46,
                    62
                ],
                "intonation": [
                    null,
                    "REDUCED"
                ],
                "letters": ".",
                "phone_count": 3,
                "quality_score": 81.0,
                "stress_level": 0,
                "stress_score": 96.75098630772801
            }
        ],
        "word": "Mr."
    },
    {
        "phone_score_list": [
            {
                "extent": [
                    62,
                    76
                ],
                "phone": "s",
                "quality_score": 98.42857142857143,
                "sound_most_like": "ah",
                "stress_level": null
            },
            {
                "extent": [
                    76,
                    85
                ],
```

```
            "phone": "m",
            "quality_score": 100.0,
            "sound_most_like": "m",
            "stress_level": null
        },
        {
            "extent": [
                85,
                98
            ],
            "phone": "ih",
            "quality_score": 100.0,
            "sound_most_like": "ih",
            "stress_level": 1,
            "stress_score": 100.0
        },
        {
            "child_phones": [
                {
                    "extent": [
                        98,
                        120
                    ],
                    "quality_score": 88.36363636363636,
                    "sound_most_like": "z"
                },
                {
                    "extent": [
                        120,
                        121
                    ],
                    "quality_score": 35.0,
                    "sound_most_like": "n"
                },
                {
                    "extent": [
                        121,
                        122
                    ],
                    "quality_score": 41.666666666666664,
                    "sound_most_like": "hh"
                }
            ],
            "extent": [
                98,
                122
            ],
            "phone": "th",
            "quality_score": 84.19444444444444,
            "sound_most_like": "z",
            "stress_level": null
        }
    ],
    "quality_score": 96.0,
    "syllable_score_list": [
        {
            "extent": [
                62,
                122
            ],
            "intonation": [
                "RISE",
                "FALL"
            ],
            "letters": "smith",
            "phone_count": 4,
            "pitch_range": [
                204.12,
                142.72
```

```
            ],
            "quality_score": 96.0,
            "stress_level": 1,
            "stress_score": 100.0
        }
    ],
    "word": "Smith"
},
{
    "phone_score_list": [
        {
            "extent": [
                132,
                140
            ],
            "phone": "ih",
            "quality_score": 93.125,
            "sound_most_like": "b",
            "stress_level": 1,
            "stress_score": 95.02320783909232
        },
        {
            "extent": [
                140,
                146
            ],
            "phone": "z",
            "quality_score": 100.0,
            "sound_most_like": "z",
            "stress_level": null
        },
        {
            "extent": [
                146,
                150
            ],
            "phone": "ah",
            "quality_score": 100.0,
            "sound_most_like": "ah",
            "stress_level": 0,
            "stress_score": 95.02320783909232
        },
        {
            "extent": [
                150,
                154
            ],
            "phone": "n",
            "quality_score": 100.0,
            "sound_most_like": "n",
            "stress_level": null
        },
        {
            "extent": [
                154,
                161
            ],
            "phone": "t",
            "quality_score": 98.57142857142857,
            "sound_most_like": "ah",
            "stress_level": null
        }
    ],
    "quality_score": 98.0,
    "syllable_score_list": [
        {
            "extent": [
                132,
                140
```

```
            ],
            "intonation": [
                "RISE",
                "FALL"
            ],
            "letters": "i",
            "phone_count": 1,
            "pitch_range": [
                188.69571428571405,
                108.12
            ],
            "quality_score": 93.0,
            "stress_level": 1,
            "stress_score": 95.02320783909232
        },
        {
            "extent": [
                140,
                161
            ],
            "intonation": [
                null,
                "RISE"
            ],
            "letters": "sn't",
            "phone_count": 4,
            "pitch_range": [
                126.69,
                179.90550000000005
            ],
            "quality_score": 100.0,
            "stress_level": 0,
            "stress_score": 95.02320783909232
        }
    ],
    "word": "isn't"
},
{
    "phone_score_list": [
        {
            "extent": [
                161,
                168
            ],
            "phone": "ih",
            "quality_score": 99.85714285714286,
            "sound_most_like": "ih",
            "stress_level": 1,
            "stress_score": 100.0
        },
        {
            "extent": [
                168,
                190
            ],
            "phone": "t",
            "quality_score": 96.36363636363636,
            "sound_most_like": "l",
            "stress_level": null
        }
    ],
    "quality_score": 98.0,
    "syllable_score_list": [
        {
            "extent": [
                161,
                190
            ],
            "intonation": [
```

```
                        null,
                        "RISE"
                    ],
                    "letters": "it",
                    "phone_count": 2,
                    "pitch_range": [
                        179.90550000000007,
                        208.56
                    ],
                    "quality_score": 98.0,
                    "stress_level": 1,
                    "stress_score": 100.0
                }
            ],
            "word": "it"
        }
    ]
  }
}
```


3.0 Score audio of phone list

Verb and Url

```
POST
https://api.speechace.co/api/scoring/phone_list/v0.1/json?key={de
veloper product key}&dialect={dialect to score
against}&user_id={user identifier}
```

Multi-part Post body

```
phone_list={list of phones separated by |}
user_audio_file={mp3 audio file}
```

Description

Use this function to score an audio with respect to a list of phones and a dialect.

Parameters

Name	Description	Example
phone_list	a list of phone, separated by |.	`b|r|ah`
user_audio_file	a mp3 audio file with sampling rate 16kHz or higher. The file size must be 20 seconds or less.	
dialect	Which dialect should the audio be scored against. Currently, en-us and en-gb are supported.	`en-us`

Name	Description	Example
user_id	This can be any unique user identifier. Ideally, it should be non personal information of a user.	1234
key	This is a key issued by SpeechAce. If two products are billed together, they share the same product key. If two products are billed separately, they use two different keys.	

Response

When the scoring is successful,

```
{
    "status": "success",
    "quota_remaining": <number or ,-1 if user has unlimited quota>,
    "word_score": {
        "word": <the input phone list>,
        "quality_score": <floating point number between 0 and 100>,
        “phone_score_list”: [
            <phone score of the first phone in the input phone list>,
            <phone score of the second phone in the input phone list>,
            ...
            <phone score of the last phone in the input phone list>
        ]
    }
}
```

Each <phone score> has the following content:

```
{
    "phone": <For General American, it is one of the phones in ARPABET notation>,
    "quality_score": <floating point number between 0 and 100>,
    "stress_level": <null for consonant; 0 - unstressed, 1 - primary, 2 - secondary>,
    "extent": [
       <start time of the phone in the audio in 10 milliseconds>,
       <end time of the phone in the audio in 10 milliseconds>
    ],
    "sound_most_like": <which phone does the extent of the audio sound most like>,
    "child_phones": <missing if no extra phone is inserted, or it is a list of phone score>
}
```

When the scoring fails,

```
{
    "status": "error",
    "short_message": <a short error message>,
```

```
    "detail_message": <a detailed error message>
}
```

Some of the errors are:

1. short message: error_audio_too_long
   detail message: The audio file is too long. It must be 20 seconds or shorter.

2. short message: error_timeout
   detail message: The request time out because the server is too busy.

3. short message: error_unknown_words
   detail message: The server could not identify the following words: <list of unknown words separated by |>

4. short message: error_out_of_quota
   detail message: The user has run out of quota

Example

Request:

```
POST
https://api.speechace.co/api/scoring/phone_list/v0.1/json?key={develop
er product key}&dialect=general_american&user_id=1234

    phone_list=ae|p|ah|l
    user_audio_file={mp3 audio file}
```

Response:


```
{
    "quota_remaining": -1,
    "status": "success",
    "word_score": {
        "phone_score_list": [
            {
                "extent": [
                    9,
                    26
                ],
                "phone": "ae",
                "quality_score": 99.0,
                "sound_most_like": "ae",
                "stress_level": null
            },
            {
                "extent": [
                    26,
                    36
                ],
                "phone": "p",
                "quality_score": 100.0,
                "sound_most_like": "p",
                "stress_level": null
```

```
            },
            {
                "extent": [
                    36,
                    40
                ],
                "phone": "ah",
                "quality_score": 99.0,
                "sound_most_like": "ah",
                "stress_level": null
            },
            {
                "extent": [
                    40,
                    54
                ],
                "phone": "l",
                "quality_score": 99.71428571428571,
                "sound_most_like": "k",
                "stress_level": null
            }
        ],
        "quality_score": 99.0,
        "word": "ae|p|ah|l"
    }
}
```


4.0 Validate text exists in Speechace lexicon

Verb and Url

```
POST
https://api.speechace.co/api/validating/text/v0.1/json?key={developer product key}&dialect={dialect to validate against}&user_id={user identifier}&text={text to validate}
```

Description

Use this function to validate whether all words in a text exists in the Speechace lexicon for a specific dialect.

Parameters

Name	Description	Example
text	a list of words	`Oblivious Schwartz Fusion`
dialect	Which lexicon to use for validation. Currently, en-us and en-gb are supported.	`en-us`
user_id	This can be any unique user identifier. Ideally, it should be non personal information of a user.	1234

Name	Description	Example
key	This is a key issued by SpeechAce. If two products are billed together, they share the same product key. If two products are billed separately, they use two different keys.	

Response

When the all words are found to be within lexicon

```
{
    "quota_remaining": -1,
    "status": "success"
}
```

When out of lexicon words are found

```
{
    "detail_message": "smithosonian,strategery",
    "short_message": "error_unknown_words",
    "status": "error"
}
```

5.0 Generate reference audio

Verb and Url

```
POST
https://api.speechace.co/api/ttsing/text/v0.1/wav?key={developer
product key}&dialect={dialect to use for audio
generation}&user_id={user identifier}&text={text to use for
generation}
```

Description

Use this function to generate automated text to speech audio to use as reference audio for a given text.

Parameters

Name	Description	Example
text	a word, phrase or sentence.	`Good afternoon, Oklahoma.`

Name	Description	Example
dialect	Dialect to use for audio generation. Currently, en-us and en-gb are supported.	en-us
user_id	This can be any unique user identifier. Ideally, it should be non personal information of a user.	1234
key	This is a key issued by SpeechAce. If two products are billed together, they share the same product key. If two products are billed separately, they use two different keys.	

Response

On success the POST operation returns a .wav file with the audio.

When out of lexicon words are found

{"status":"error","short_message":"error_unknown_words","detail_message":"Strategery"}