

# Microsoft Azure Cognitive Services- Text Translator API

## API REST Specification

Document Number: 1

December 2020

<b>Release: Translator API 2020</b>	<b>Document Status: Final Draft</b>
<b>Version: 1.0.0</b>	

### Abstract

Translate text in real time using **Microsoft Azure Cognitive Services – Translator API** which is an AI service for real-time text translation. The four different end points defined in the application use more than 70 languages and language codes defined by the cognitive services for text translation to desired language, detection of the language for the given text, suggesting alternate translations for provided text and translation language; and getting the length of the input sentence consisting number of characters and spaces in each sentence.

*Keywords:* Microsoft, Azure, Cognitive Services, AI, real-time translation, translator, API

## Microsoft Azure Cognitive Services- Text Translator API

### API REST Specification

Microsoft Azure Cognitive Services- Text Translator uses Neural Machine Translation (NMT) for high- quality AI-powered machine translations.

Using this Text Translator AI, we pass our own parameters to the application already registered to the Azure Cognitive service and structure and utilize the response from this third-party API according to our own requirements for the application.

### API Operation Templates:

Sr. No.	Operation	Uniform API Operation	Description	Example Payload
1.	/api/language_code/:scope	GET Resource	GET must be used to obtain the language code to be used in each of the following scopes. It takes query parameter ' <b>scope</b> ' with any one of the following values: <ul style="list-style-type: none"> <li>translation</li> <li>transliteration</li> <li>dictionary</li> </ul>	GET /api/language_code/translation
2.	/api/translate	POST Resource	POST must be used to pass the input <b>text</b> to be translated	POST: /api/translate { "text": "string",

			to the <u>language code</u> mentioned in the <b>to</b> input parameter of the request body.	“to”: “string” }
3.	/api/detect	POST Resource	POST must be used to detect the <u>language code</u> of the input property <b>text</b>	POST: /api/detect { “text”: “string” }
4.	/api/break_sentence	POST Resource	POST must be used to identify the sentence breaks occurring in the given input <b>text</b> in the given body	POST: /api/break_sentence { “text”: “string” }
5.	/api/transliterate	POST Resource	POST must be used to phonetically translate input <b>text</b> from once script to another supported by the <u>transliteration table</u>	POST: /api/transliterate { “text”: “string”, “language”: “string”, “fromScript”: “string”, “toScript”: “string” }
6.	/api/alt_translations	POST Resource	POST must be used to get alternative	POST: /api/alt_translations

			translations for a word and a small number of idiomatic phrases.	{ “text”: “string “, “from”: “string”, “to”: “string“ }
--	--	--	--	---

**Swagger Documentation:**<http://64.225.59.48:3000/docs/#/>

**Application API End points****GET API/LANGUAGE\_CODE/{SCOPE}**Description:

- This REST endpoint is used to obtain a set of languages and codes currently supported by each of the following REST API operations.
- /api/language\_code/{scope} will output a list of languages and their corresponding language codes which then can be used in its respective context. That is, the result of this API with scope: translation will be used as an input suggestion for /api/translate and for result verification for /api/detect; scope: transliteration will be used as an input suggestion for /api/transliterate and scope: dictionary for /api/alt\_translations

Request URL:Send a GET request to:

[http://64.225.59.48:3000/api/language\\_code/{scope}](http://64.225.59.48:3000/api/language_code/{scope})

Request body:Attributes Required for POST request:

Sr. No.	Attribute Name	Mandatory	Rule
1.	scope	YES	String query parameter.  Allowed values are `translation`,

Sr. No.	Attribute Name	Mandatory	Rule
			`transliteration` and `dictionary`

Response body:

- A successful response is a JSON array with a 200 Success status.
- The **scope** query parameter is used to define which group of languages is required by the user.
- scope: translation provides languages and their codes supported by the <http://64.225.59.48:3000/api/translate> api.
- scope: transliteration provides capabilities of converting text in one language from one script to another script used by the <http://64.225.59.48:3000/api/transliterate> api.
- scope: dictionary provides language pairs from the Dictionary returns data to be implemented by [http://64.225.59.48:3000/api/alt\\_translations](http://64.225.59.48:3000/api/alt_translations) api.
- An unsuccessful request returns the error with a 400 Bad Request status.

Response Status Codes:

<u>Status Code</u>	<u>Description</u>
200	Success
400	Invalid or missing query parameter. Correct query parameter before sending the request

Example Requests:

- **Scope: translation**

<b>URL</b>
<a href="http://64.225.59.48:3000/api/language_code/translation">http://64.225.59.48:3000/api/language_code/translation</a>
<b>REQUEST</b>
GET API/LANGUAGE_CODE/TRANSLATION Content-type: application/json
<b>RESPONSE</b>
200 Content-Type: application/json  <pre>{   "translation": {     //... set of languages supported to translate text     (scope=translation)   }, }</pre>



## POST API/TRANSLATE

### Description:

- This REST endpoint is used to translate the provided input 'text' to the desired language which is provided as an input body parameter in 'to'.
- /api/translate will auto-detect the language of the text provided in the body and indicates the confidence score of the detected language.

### Request URL:

Send a POST request to:

<http://64.225.59.48:3000/api/translate>

### Request body:

- The body of the request is a JSON object.
- The JSON object has a string property named '**text**', which represents the string to translate.
- The other string property which is an input param in the JSON request body is '**to**' which specifies the language of the output text.
- The target language must be one of the [supported language codes](#) or [set of languages supported to translate text](#)

- included in the translation scope.

For example, use "to": "es" to translate the input text to Spanish.

- It is possible to translate to multiple languages simultaneously by passing an array to the 'to' input param.

For example, use "to": ["it", "de"] to translate the same input text to Italian and German simultaneously.

### Attributes Required for POST request:

Sr. No.	Attribute Name	Mandatory	Rule
1.	text	YES	String input.

Sr. No.	Attribute Name	Mandatory	Rule
			Cannot exceed 10,000 characters including spaces
2.	to	YES	String input. Must be one of the language code supported in the Microsoft Azure Cognitive Services Text Translation <a href="#">document</a> . OR output refer <a href="http://64.225.59.48:3000/api/language_code/translation">http://64.225.59.48:3000/api/language_code/translation</a>

Example JSON Request Body:

JSON
<pre>{   "text": "hello",   "to": ["it", "de"] }</pre>

The following limitations apply <sup>[1]</sup>:

- The entire text included in the request cannot exceed 10,000 characters including spaces.

Response body:

- A successful response is a JSON array with a 200 Success status.
- A result object includes the following properties <sup>[1]</sup>:
  - detectedLanguage: An object that describes the detected language with the following properties:
    - language: A string representing the code or codes of the detected language.
    - score: A score is a float value that indicates the confidence in the result of the language detected. This score is between zero(low) and one(high). A lower score indicates lower confidence in the result of the detected language.

- translations: The JSON array represents the result of each target language translation specified through the 'to' request body parameter. Each element in the array includes the following properties:
  - text: A string output which gives the translation of the input text to the target language code. With the 'profanityMarker' enabled, profanities are handled by replacing with '\*\*\*' sign in the response target language translation.
  - to: A string that represents the language code of the output target language.
- An unsuccessful request returns the error with a 400 Bad Request status.

Response Status Codes:

<u>Status Code</u>	<u>Description</u>
200	Success
400	One of the request body parameters is missing or not valid. Correct request parameters before sending the request

Example Requests:

- **Translate input text to single language**

URL
<a href="http://64.225.59.48:3000/api/translate">http://64.225.59.48:3000/api/translate</a>
REQUEST
POST API/TRANSLATE Content-type: application/json <pre>{   "text": "What's your name ",   "to": "ar" }</pre>
RESPONSE
200 Content-Type: application/json <pre>[   {     "detectedLanguage": {       "language": "en",       "score": 1     },     "translations": [       {         "text": "ما هو اسمك",         "to": "ar"       }     ]   } ]</pre>

- Translate input text to multiple languages simultaneously.

URL
<a href="http://64.225.59.48:3000/api/translate">http://64.225.59.48:3000/api/translate</a>
REQUEST
POST API/TRANSLATE Content-type: application/json <pre>{   "text": "hello ",   "to": ["it", "de"] }</pre>
RESPONSE
200 Content-Type: application/json <pre>[   {     "detectedLanguage": {       "language": "en",       "score": 1     },     "translations": [       {         "text": "Ciao ",         "to": "it"       },       {         "text": "Hallo ",         "to": "de"       }     ]   } ]</pre>

- Translate a text containing profanities

URL
<a href="http://64.225.59.48:3000/api/translate">http://64.225.59.48:3000/api/translate</a>
REQUEST
POST API/TRANSLATE Content-type: application/json <pre>{   "text": "You are a jackass",   "to": "es" }</pre>
RESPONSE
200 Content-Type: application/json <pre>[   {     "detectedLanguage": {       "language": "en",       "score": 1     },     "translations": [       {         "text": "Eres un ***.",         "to": "es"       }     ]   } ]</pre>

- Translate a text with missing or invalid request body parameters

URL
<a href="http://64.225.59.48:3000/api/translate">http://64.225.59.48:3000/api/translate</a>
REQUEST
POST API/TRANSLATE Content-type: application/json <pre>{   "text": "Good to see you",   "to": "" }</pre>
RESPONSE
400 Content-Type: application/json  "Request failed with status code 400"

## POST API/DETECT

### Description:

- This REST endpoint identifies the language code of a piece of text passes to it as a request input parameter.
- /api/detect will auto-detect the language code of the text, provide a confidence score for the detected language, and list alternative languages with their confidence score.

### Request URL:

Send a POST request to:

<http://64.225.59.48:3000/api/detect>

### Request body:

- The body of the request is a JSON object.
- The JSON object has a string property names '**text**', which represents the string whose language code is to be detected.
- The language auto-detection works better with longer input text. <sup>[2]</sup>

### Attributes Required for POST request:

Sr. No.	Attribute Name	Mandatory	Rule
1.	text	YES	String input. Cannot exceed 10,000 characters including spaces

### Example JSON Request Body:

JSON
<pre>{   "text": "Good to see you" }</pre>



The following limitations apply <sup>[2]</sup>:

- The entire text included in the request cannot exceed 50,000 characters including spaces.

Response body:

- A successful response is a JSON array with a 200 Success status.
- The result object includes the following properties:
  - language: A string representing the code or output code of the detected language.
  - score: A score is a float value that indicates the confidence in the result of the language detected. This score is between zero(low) and one(high). A lower score indicates lower confidence in the result of the detected language.
  - alternatives: This is a JSON array of other possible languages detected from the input 'text' body parameter. Each element in the array includes the following properties:
    - language: Same as listed above
    - score: Same as listed above
- An unsuccessful request returns the error with a 400 Bad Request status.

Response Status Codes:

<u>Status Code</u>	<u>Description</u>
200	Success
400	One of the request body parameters is missing or not valid. Correct request parameters before sending the request

Example Requests:

- Detect Language Code of the uniform input text

URL
<a href="http://64.225.59.48:3000/api/detect">http://64.225.59.48:3000/api/detect</a>
REQUEST
POST API/DETECT Content-type: application/json  { "text": "Good to see you" }
RESPONSE
200 Content-Type: application/json  [ { "language": "en", "score": 1, "alternatives": [ { "language": "", "score": "" } ] } ]

- Detect Language Code of a mixed input text

URL
<a href="http://64.225.59.48:3000/api/detect">http://64.225.59.48:3000/api/detect</a>
REQUEST
POST API/DETECT Content-type: application/json  { "text": "你好 Good to see you" }
RESPONSE
200 Content-Type: application/json  [ { "language": "en", "score": 0.67, "alternatives": [ { "language": "zh-Hans", "score": 0.33 } ] } ]

## POST API/BREAK\_SENTENCE

### Description:

- This REST endpoint is used to identify the positioning of sentence boundaries for the input text provided.
- /api/break\_sentence auto-detects the language code of the provided input text and outputs each sentence's length from the given input text.

### Request URL:

Send a POST request to:

[http://64.225.59.48:3000/api/break\\_sentence](http://64.225.59.48:3000/api/break_sentence)

### Request body:

- The body of the request is a JSON Object.
- The JSON object has a string property named '**text**', which represents the string whose value is used to compute the sentence boundaries.

### Attributes Required for POST request:

Sr. No.	Attribute Name	Mandatory	Rule
1.	text	YES	String input. Cannot exceed 10,000 characters including spaces

Example JSON Request Body:

JSON
<pre>{   "text": "Hello, how are you? Hope you are doing great!" }</pre>

The following limitations apply <sup>[3]</sup>:

- The text value of JSON object cannot exceed 50,000 characters including spaces.
- The entire text included in the request cannot exceed 50,000 characters including spaces.

Response body:

- A successful response is a JSON array with a 200 Success status.
- A result object includes the following properties <sup>[3]</sup>:
  - sentLen: An array of representing the sentence number and length of each sentence present in the text element. Also, array length represents the number of sentences present in the given input text.
  - detectedLanguage: A JSON object which describes the detected language with the following properties:
    - language: A string representing the code of the detected language.
    - score: A score is a float value that indicates the confidence in the result of the language detected. This score is between zero(low) and one(high). A lower score indicates lower confidence in the result of the detected language.
- When an input text consists of mixed languages, the /api/break\_sentence will detect the language that has a greater number of words and state it's confidence score.
- An unsuccessful request returns the error with a 400 Bad Request status.

Response Status Codes:

<u>Status Code</u>	<u>Description</u>
200	Success
400	One of the request body parameters is missing or not valid. Correct request parameters before sending the request

Example Requests:

- **Sentence Break with uniform input text**

URL
<a href="http://64.225.59.48:3000/api/break_sentence">http://64.225.59.48:3000/api/break_sentence</a>
REQUEST
POST API/BREAK_SENTENCE Content-type: application/json  { "text": "Hello, how are you? Hope you are doing great! Have a good time" }
RESPONSE
200 Content-Type: application/json  [ { "sentLen": [ "Sentence 1 : 20", "Sentence 2 : 26", "Sentence 3 : 16" ], "detectedLanguage": { "language": "en", "score": 1 } } ]

- **Sentence Break with mixed input text**

URL
<a href="http://64.225.59.48:3000/api/break_sentence">http://64.225.59.48:3000/api/break_sentence</a>
REQUEST
POST API/BREAK_SENTENCE Content-type: application/json <pre>{   "text": "Hello, how are you? Hope you are doing great! Хорошо тебе провести время" }</pre>
RESPONSE
200 Content-Type: application/json <pre>[   {     "sentLen": [       "Sentence 1 : 20",       "Sentence 2 : 26",       "Sentence 3 : 26"     ],     "detectedLanguage": {       "language": "en",       "score": 0.69     }   } ]</pre>

## POST API/TRANSLITERATE

### Description:

- This REST endpoint converts text in one language from one script to another script, that is, it gives a phonetic translation from of the input text from one language to another.
- /api/transliterate supports the transliteration of input text from one script to another based on the rules in the [document](#).

### Request URL:

Send a POST request to:

<http://64.225.59.48:3000/api/transliterate>

### Request body:

- The body of the request is a JSON object.
- The JSON object has a string property named '**text**', which represents the string which needs a phonetic translation, that is, needs transliteration.
- Another required string property is '**language**' which indicates the [language code](#) mentioned in the [transliteration table](#) or [supported codes](#).
- The request body takes two other string properties '**fromScript**' and '**toScript**' which indicates the name of the [script](#) supported by the [transliteration table](#) or [supported scripts](#).

### Attributes Required for POST request:

Sr. No.	Attribute Name	Mandatory	Rule
1.	text	YES	String input. Cannot exceed 10,000 characters including spaces
2.	language	YES	String input.



Sr. No.	Attribute Name	Mandatory	Rule
			Must be one of the language code supported in the Microsoft Azure Cognitive Services Text Transliteration <u>document</u> or <u>supported codes</u> .
3.	fromScript	YES	String input. Must be one of the <b>from script</b> included in the To/From Script pair supported in the Microsoft Azure Cognitive Services Text Transliteration <u>document</u> or <u>supported fromScripts</u> .
4.	toScript	YES	String input. Must be one of the <b>to script</b> included in the To/From Script pair supported in the Microsoft Azure Cognitive Services Text Transliteration <u>document</u> or <u>supported toScripts</u> .

Example JSON Request Body:

JSON
<pre>{   "text": "สวัสดี",   "language": "th",   "fromScript": "Thai",   "toScript": "Latn" }</pre>

The following limitations apply <sup>[4]</sup>:

- The text value of JSON object cannot exceed 1,000 characters including spaces.
- The entire text included in the request cannot exceed 5,000 characters including spaces.

Response body:

- A successful response is a JSON array with a 200 Success status.
- The result object includes the following properties <sup>[4]</sup>:
  - text: The output string which is a phonetic translation or transliteration of the input text in the script passed in the 'toScript' property of the request body.
  - script: The script which is used to transliterate the input text to, that is, the value of the script mentioned in the 'toScript' property. This property is used to verify the script of the transliterated output text.
- An unsuccessful request returns the error with a 400 Bad Request status.

Response Status Codes:

<u>Status Code</u>	<u>Description</u>
200	Success
400	One of the request body parameters is missing or not valid. Correct request parameters before sending the request

Example Requests:

- Transliterate input text to script according to the transliteration table.

URL
<a href="http://64.225.59.48:3000/api/transliterate">http://64.225.59.48:3000/api/transliterate</a>
REQUEST
POST API/TRANSLITERATE Content-type: application/json <pre>{   "text": "สวัสดี",   "language": "th",   "fromScript": "Thai",   "toScript": "Latn" }</pre>
RESPONSE
200 Content-Type: application/json <pre>[   {     "text": "sawatdi",     "script": "Latn"   } ]</pre>

- Transliterate input text to script not supported in transliteration table.

URL
<a href="http://64.225.59.48:3000/api/transliterate">http://64.225.59.48:3000/api/transliterate</a>
REQUEST
POST API/TRANSLITERATE Content-type: application/json <pre>{   "text": "สวัสดี",   "language": "th",   "fromScript": "Thai",   "toScript": "Hans" }</pre>
RESPONSE
400 Content-Type: application/json  "Request failed with status code 400"

## POST API/ALT\_TRANSLATIONS

### Description:

- This REST endpoint provides alternate translations for a word or certain idiomatic phrases in the input language provided.
- /api/alt\_translations responses with a list of alternate translations for the provided input text, in the input language provided and also lists the part-of-speech of the listed text in the translated language along with a list of back-translations in the original language of the text to provide context.

### Request URL:

Send a POST request to:

[http://64.225.59.48:3000/api/alt\\_translations](http://64.225.59.48:3000/api/alt_translations)

### Request body:

- The body of the request is a JSON object.
- The JSON object has a string property names '**text**', which represents the sting which is to be translated.
- The '**from**' string property specifies the language code of the input text. It must be one of the language codes included in the [dictionary scope](#).
- The '**to**' string property specifies the language code of the target output text i.e the language to which the input text is to be translated. It must be one of It must be one of the language codes included in the [dictionary scope](#).

### Attributes Required for POST request:

Sr. No.	Attribute Name	Mandatory	Rule
1.	text	YES	String input.

Sr. No.	Attribute Name	Mandatory	Rule
			Cannot exceed 100 characters including spaces.
2.	from	YES	String input. Must be one of the language code supported in the Microsoft Azure Cognitive Services Text Translation <u>document</u> or <u>languages supported for alternative translations</u>
3.	to	YES	String input. Must be one of the language code supported in the Microsoft Azure Cognitive Services Text Translation <u>document</u> or <u>languages supported for alternative translations</u>

Example JSON Request Body:

JSON
<pre>{   "text": "house",</pre>

JSON
<pre>"from": "en", "to": "es" }</pre>

The following limitations apply <sup>[5]</sup>:

- The text value of JSON object cannot exceed 100 characters including spaces.

Response body:

- A successful response is a JSON array with a 200 Success status.
- If the term in the input text is not defined in the dictionary, the response is 200 (OK) but the result array will be empty.
- The result object includes the following properties <sup>[5]</sup>:
  - normalizedTarget: A string with the normalized form of the output text translated in the provided input language.
  - displayTarget: A string which is a better representation of the normalizedTarget term for best end-user display.
  - postTag: This property indicates the part-of-speech of the input text in the translated language. The following Tags represent the given part-of-speech:

Tag name	Description
ADJ	Adjectives
ADV	Adverbs
CONJ	Conjunctions

DET	Determiners
MODAL	Verbs
NOUN	Nouns
PREP	Prepositions
PRON	Pronouns
VERB	Verbs
OTHER	Other

- confidence: A float value that indicates the confidence in the result of the translated output text. This score is between zero(low) and one(high). A lower score indicates lower confidence in the translation pair.
- prefixWord: A string value, which is a gendered determination of nouns, in the provided language for translation if it has any gendered determinations. If there is no prefix, the output for this property is an empty string.
- backTranslations: This output a list of backtranslations of the text in context to the normalizedTarget value in the original language of the input text. Each element of the backTranslations list is an object that has the following properties:
  - normalizedText: A string with the normalized form of the source input text term which is backtranslated in the original language of the input text.
  - displayText: A string which is a better representation of the normalizedText term for best end-user display.



- numExamples: An integer value that represents the number of examples that are available, for the given translation pair of the input text and the target translated text, in the training model.
  - frequencyCount: An integer value which represents the frequency of the given translation pair of the input text and the output translated text. This count can be used to sort the back-translations to get the most frequent pairs.
- An unsuccessful request returns the error with a 400 Bad Request status.

#### Response Status Codes:

<u>Status Code</u>	<u>Description</u>
200	Success
400	One of the request body parameters is missing or not valid. Correct request parameters before sending the request

#### Example Requests:

- **Alternate Translation for input text listed in the dictionary with available alternate translations in the target language**

<b>URL</b>
<a href="http://64.225.59.48:3000/api/alt_translations">http://64.225.59.48:3000/api/alt_translations</a>
<b>REQUEST</b>
POST API/ALT_TRANSALTE Content-type: application/json <pre>{   "text": "shark",   "from": "en",   "to": "es" }</pre>

**RESPONSE**

200

Content-Type: application/json

```
[
  {
    "normalizedTarget": "tiburón",
    "displayTarget": "tiburón",
    "posTag": "OTHER",
    "confidence": 0.8182,
    "prefixWord": "",
    "backTranslations": [
      {
        "normalizedText": "shark",
        "displayText": "shark",
        "numExamples": 0,
        "frequencyCount": 45
      }
    ]
  },
  {
    "normalizedTarget": "escualo",
    "displayTarget": "escualo",
    "posTag": "NOUN",
    "confidence": 0.1818,
    "prefixWord": "",
    "backTranslations": [
      {
        "normalizedText": "shark",
        "displayText": "shark",
        "numExamples": 1,
        "frequencyCount": 10
      }
    ]
  }
]
```

- **Alternate Translation for input text not listed in the dictionary**
  - Usually long phrases and sentences do not have listed examples of alternate translations.

URL
<u><a href="http://64.225.59.48:3000/api/alt_translations">http://64.225.59.48:3000/api/alt_translations</a></u>
REQUEST
POST API/TRANSLATE Content-type: application/json <pre>{   "text": "It's a wonderful day!",   "from": "en",   "to": "es" }</pre>
RESPONSE
200 Content-Type: application/json <pre>[]</pre>

- **Alternate Translations with missing attribute in the request body**

URL
<a href="http://64.225.59.48:3000/api/alt_translations">http://64.225.59.48:3000/api/alt_translations</a>
REQUEST
POST API/TRANSLATE Content-type: application/json <pre>{   "text": "It's a wonderful day!",   "from": "",   "to": "es" }</pre>
RESPONSE
400 Content-Type: application/json  "Request failed with status code 400: Missing attribute 'from'"

## References

- [1] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-translate>
- [2] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-detect>
- [3] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-break-sentence>
- [4] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-transliterate>
- [5] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-dictionary-lookup>
- [6] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/translator-info-overview>
- [7] <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-languages>