

Endpoints

Translate Text

Translation converts text from one language to another while preserving its meaning. For Example: 'मैं ऑफिस जा रहा हूँ' translates to 'I am going to the office' in English, where the script and language change, but the original meaning remains the same.

Available languages:

en-IN : English

hi-IN : Hindi

bn-IN : Bengali

gu-IN : Gujarati

kn-IN : Kannada

ml-IN : Malayalam

mr-IN : Marathi

od-IN : Odia

pa-IN : Punjabi

ta-IN : Tamil

te-IN : Telugu

For hands-on practice, you can explore the notebook tutorial on [**Translate API Tutorial**](#).

POST <https://api.sarvam.ai/d/translate>

Try it

Endpoints > Translate Text

```
curl -X POST -H "Content-Type: application/json" -d '{
  --data '{
    "input": "<string>",
    "source_language_code": "en-IN",
    "target_language_code": "en-IN",
    "speaker_gender": "Female",
    "mode": "formal",
    "model": "mayura:v1",
    "enable_preprocessing": false,
    "output_script": "roman",
    "numerals_format": "international"
  }'
```

200 400 403 422 429 500

```
{
  "request_id": "<string>",
  "translated_text": "<string>"
}
```

Headers

api-subscription-key string default:

Your unique subscription key for authenticating requests to the Sarvam AI Speech-to-Text API.

[Here are the steps to get your api key](#)

Body

application/json

input string **required**

The text you want to translate. This is the input text that will be processed by the translation model.

Endpoints > Translate Text

The language code of the input text. This specifies the source language for translation.

Note: The source language should either be an Indic language or English. As we supports both Indic-to-English and English-to-Indic translation.

Available options: `en-IN` , `hi-IN` , `bn-IN` , `gu-IN` , `kn-IN` , `ml-IN` , `mr-IN` , `od-IN` , `pa-IN` , `ta-IN` , `te-IN`

target_language_code `enum<string>` **required**

The language code of the translated text. This specifies the target language for translation.

Note:The target language should either be an Indic language or English. As we supports both Indic-to-English and English-to-Indic translation.

Available options: `en-IN` , `hi-IN` , `bn-IN` , `gu-IN` , `kn-IN` , `ml-IN` , `mr-IN` , `od-IN` , `pa-IN` , `ta-IN` , `te-IN`

speaker_gender `enum<string>`

Please specify the gender of the speaker for better translations. This feature is only supported for the code-mixed translation models currently.

Available options: `Male` , `Female`

mode `enum<string>`

Specifies the tone or style of the translation. Choose between formal, classic-colloquial and modern-colloquial translations. Default is formal.

Available options: `formal` , `modern-colloquial` , `classic-colloquial` , `code-mixed`

model `enum<string>`

Specifies the translation model to use. Currently, only one model is supported.Note:- This parameter is optional but will be deprecated by the end of January; avoid including it in your requests.

Endpoints > Translate Text

This will enable custom preprocessing of the input text which can result in better translations.

output_script enum<string> | null

output_script: This is an optional parameter which controls the transliteration style applied to the output text.

Transliteration: Converting text from one script to another while preserving pronunciation.

We support transliteration with four options:

null (default): No transliteration applied.

roman : Transliteration in Romanized script.

fully-native : Transliteration in the native script with formal style.

spoken-form-in-native : Transliteration in the native script with spoken style.

Example:

English: Your EMI of Rs. 3000 is pending.

Default modern translation: आपका Rs. 3000 का EMI pending है (when **null** is passed).

With postprocessing enabled, we provide the following style of outputs:

roman output: aapka Rs. 3000 ka EMI pending hai.

fully-native output: आपका रु. 3000 का ई.एम.ऐ. पेंडिंग है।

spoken-form-in-native output: आपका श्री थाउजेंड रूपीस का ईएमअइ पेंडिंग है।

Available options: **roman** , **fully-native** , **spoken-form-in-native**

numerals_format enum<string>

numerals_format is an optional parameter with two options:

international (default): Uses regular numerals (0-9).

native : Uses language-specific native numerals.

Example:

Available options: `international` , `native`

Response 200 application/json

Successful Response

`request_id` string | null **required**

`translated_text` string **required**

Translated text result in the requested target language.