# Translate Text

POST https://api.sarvam.ai/translate

Content-Type: application/json

\*\*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:

- \*\*`bn-IN`\*\*: Bengali

- \*\*`en-IN`\*\*: English

- \*\*`gu-IN`\*\*: Gujarati

- \*\*`hi-IN`\*\*: Hindi

- \*\*`kn-IN`\*\*: Kannada

- \*\*`ml-IN`\*\*: Malayalam

- \*\*`mr-IN`\*\*: Marathi

- \*\*`od-IN`\*\*: Odia

- \*\*`pa-IN`\*\*: Punjabi

- \*\*`ta-IN`\*\*: Tamil

- \*\*`te-IN`\*\*: Telugu

### Newly added languages:

- \*\*`as-IN`\*\*: Assamese

- \*\*`brx-IN`\*\*: Bodo

- \*\*`doi-IN`\*\*: Dogri

- \*\*`kok-IN`\*\*: Konkani

- \*\*`ks-IN`\*\*: Kashmiri

- \*\*`mai-IN`\*\*: Maithili

- \*\*`mni-IN`\*\*: Manipuri (Meiteilon)

- \*\*`ne-IN`\*\*: Nepali

- \*\*`sa-IN`\*\*: Sanskrit

- \*\*`sat-IN`\*\*: Santali

- \*\*`sd-IN`\*\*: Sindhi

- \*\*`ur-IN`\*\*: Urdu

For hands-on practice, you can explore the notebook tutorial on [Translate API Tutorial](https://github.com/sarvamai/sarvam-ai-cookbook/blob/main/notebooks/translate/Translate\_API\_Tutorial.ipynb).

Reference: https://docs.sarvam.ai/api-reference-docs/text/translate

## OpenAPI Specification

```yaml

openapi: 3.1.1

info:

title: Translate Text

version: endpoint\_text.translate

paths:

/translate:

post:

operationId: translate

summary: Translate Text

description: >-

\*\*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:

- \*\*`bn-IN`\*\*: Bengali

- \*\*`en-IN`\*\*: English

- \*\*`gu-IN`\*\*: Gujarati

- \*\*`hi-IN`\*\*: Hindi

- \*\*`kn-IN`\*\*: Kannada

- \*\*`ml-IN`\*\*: Malayalam

- \*\*`mr-IN`\*\*: Marathi

- \*\*`od-IN`\*\*: Odia

- \*\*`pa-IN`\*\*: Punjabi

- \*\*`ta-IN`\*\*: Tamil

- \*\*`te-IN`\*\*: Telugu

### Newly added languages:

- \*\*`as-IN`\*\*: Assamese

- \*\*`brx-IN`\*\*: Bodo

- \*\*`doi-IN`\*\*: Dogri

- \*\*`kok-IN`\*\*: Konkani

- \*\*`ks-IN`\*\*: Kashmiri

- \*\*`mai-IN`\*\*: Maithili

- \*\*`mni-IN`\*\*: Manipuri (Meiteilon)

- \*\*`ne-IN`\*\*: Nepali

- \*\*`sa-IN`\*\*: Sanskrit

- \*\*`sat-IN`\*\*: Santali

- \*\*`sd-IN`\*\*: Sindhi

- \*\*`ur-IN`\*\*: Urdu

For hands-on practice, you can explore the notebook tutorial on

[Translate API

Tutorial](https://github.com/sarvamai/sarvam-ai-cookbook/blob/main/notebooks/translate/Translate\_API\_Tutorial.ipynb).

tags:

- - subpackage\_text

parameters:

- name: api-subscription-key

in: header

required: true

schema:

type: string

responses:

'200':

description: Successful Response

content:

application/json:

schema:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslationResponse'

'400':

description: Bad Request

content: {}

'403':

description: Forbidden

content: {}

'422':

description: Unprocessable Entity

content: {}

'429':

description: Quota Exceeded

content: {}

'500':

description: Internal Server Error

content: {}

requestBody:

content:

application/json:

schema:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslationRequest'

components:

schemas:

Sarvam\_Model\_API\_TranslateSourceLanguage:

type: string

enum:

- value: auto

- value: bn-IN

- value: en-IN

- value: gu-IN

- value: hi-IN

- value: kn-IN

- value: ml-IN

- value: mr-IN

- value: od-IN

- value: pa-IN

- value: ta-IN

- value: te-IN

- value: as-IN

- value: brx-IN

- value: doi-IN

- value: kok-IN

- value: ks-IN

- value: mai-IN

- value: mni-IN

- value: ne-IN

- value: sa-IN

- value: sat-IN

- value: sd-IN

- value: ur-IN

Sarvam\_Model\_API\_TranslateTargetLanguage:

type: string

enum:

- value: bn-IN

- value: en-IN

- value: gu-IN

- value: hi-IN

- value: kn-IN

- value: ml-IN

- value: mr-IN

- value: od-IN

- value: pa-IN

- value: ta-IN

- value: te-IN

- value: as-IN

- value: brx-IN

- value: doi-IN

- value: kok-IN

- value: ks-IN

- value: mai-IN

- value: mni-IN

- value: ne-IN

- value: sa-IN

- value: sat-IN

- value: sd-IN

- value: ur-IN

Sarvam\_Model\_API\_TranslateSpeakerGender:

type: string

enum:

- value: Male

- value: Female

Sarvam\_Model\_API\_TranslateMode:

type: string

enum:

- value: formal

- value: modern-colloquial

- value: classic-colloquial

- value: code-mixed

Sarvam\_Model\_API\_TranslateModel:

type: string

enum:

- value: mayura:v1

- value: sarvam-translate:v1

Sarvam\_Model\_API\_TransliterateMode:

type: string

enum:

- value: roman

- value: fully-native

- value: spoken-form-in-native

Sarvam\_Model\_API\_NumeralsFormat:

type: string

enum:

- value: international

- value: native

Sarvam\_Model\_API\_TranslationRequest:

type: object

properties:

input:

type: string

source\_language\_code:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslateSourceLanguage'

target\_language\_code:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslateTargetLanguage'

speaker\_gender:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslateSpeakerGender'

mode:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslateMode'

model:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslateModel'

enable\_preprocessing:

type: boolean

output\_script:

oneOf:

- $ref: '#/components/schemas/Sarvam\_Model\_API\_TransliterateMode'

- type: 'null'

numerals\_format:

$ref: '#/components/schemas/Sarvam\_Model\_API\_NumeralsFormat'

required:

- input

- source\_language\_code

- target\_language\_code

Sarvam\_Model\_API\_TranslationResponse:

type: object

properties:

request\_id:

type:

- string

- 'null'

translated\_text:

type: string

source\_language\_code:

type: string

required:

- request\_id

- translated\_text

- source\_language\_code

```

## SDK Code Examples

```python

from sarvamai import SarvamAI

client = SarvamAI(

api\_subscription\_key="YOUR\_API\_SUBSCRIPTION\_KEY",

)

client.text.translate(

input="input",

source\_language\_code="auto",

target\_language\_code="bn-IN",

)

```

```typescript

import { SarvamAIClient } from "sarvamai";

const client = new SarvamAIClient({ apiSubscriptionKey: "YOUR\_API\_SUBSCRIPTION\_KEY" });

await client.text.translate({

input: "input",

source\_language\_code: "auto",

target\_language\_code: "bn-IN"

});

```

```go

package main

import (

"fmt"

"strings"

"net/http"

"io"

)

func main() {

url := "https://api.sarvam.ai/translate"

payload := strings.NewReader("{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}")

req, \_ := http.NewRequest("POST", url, payload)

req.Header.Add("api-subscription-key", "<apiKey>")

req.Header.Add("Content-Type", "application/json")

res, \_ := http.DefaultClient.Do(req)

defer res.Body.Close()

body, \_ := io.ReadAll(res.Body)

fmt.Println(res)

fmt.Println(string(body))

}

```

```ruby

require 'uri'

require 'net/http'

url = URI("https://api.sarvam.ai/translate")

http = Net::HTTP.new(url.host, url.port)

http.use\_ssl = true

request = Net::HTTP::Post.new(url)

request["api-subscription-key"] = '<apiKey>'

request["Content-Type"] = 'application/json'

request.body = "{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}"

response = http.request(request)

puts response.read\_body

```

```java

HttpResponse<String> response = Unirest.post("https://api.sarvam.ai/translate")

.header("api-subscription-key", "<apiKey>")

.header("Content-Type", "application/json")

.body("{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}")

.asString();

```

```php

<?php

$client = new \GuzzleHttp\Client();

$response = $client->request('POST', 'https://api.sarvam.ai/translate', [

'body' => '{

"input": "foo",

"source\_language\_code": "auto",

"target\_language\_code": "bn-IN"

}',

'headers' => [

'Content-Type' => 'application/json',

'api-subscription-key' => '<apiKey>',

],

]);

echo $response->getBody();

```

```csharp

var client = new RestClient("https://api.sarvam.ai/translate");

var request = new RestRequest(Method.POST);

request.AddHeader("api-subscription-key", "<apiKey>");

request.AddHeader("Content-Type", "application/json");

request.AddParameter("application/json", "{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}", ParameterType.RequestBody);

IRestResponse response = client.Execute(request);

```

```swift

import Foundation

let headers = [

"api-subscription-key": "<apiKey>",

"Content-Type": "application/json"

]

let parameters = [

"input": "foo",

"source\_language\_code": "auto",

"target\_language\_code": "bn-IN"

] as [String : Any]

let postData = JSONSerialization.data(withJSONObject: parameters, options: [])

let request = NSMutableURLRequest(url: NSURL(string: "https://api.sarvam.ai/translate")! as URL,

cachePolicy: .useProtocolCachePolicy,

timeoutInterval: 10.0)

request.httpMethod = "POST"

request.allHTTPHeaderFields = headers

request.httpBody = postData as Data

let session = URLSession.shared

let dataTask = session.dataTask(with: request as URLRequest, completionHandler: { (data, response, error) -> Void in

if (error != nil) {

print(error as Any)

} else {

let httpResponse = response as? HTTPURLResponse

print(httpResponse)

}

})

dataTask.resume()

```

# Language Identification

POST https://api.sarvam.ai/text-lid

Content-Type: application/json

Identifies the language (e.g., en-IN, hi-IN) and script (e.g., Latin, Devanagari) of the input text, supporting multiple languages.

Reference: https://docs.sarvam.ai/api-reference-docs/text/identify-language

## OpenAPI Specification

```yaml

openapi: 3.1.1

info:

title: Language Identification

version: endpoint\_text.identifyLanguage

paths:

/text-lid:

post:

operationId: identify-language

summary: Language Identification

description: >-

Identifies the language (e.g., en-IN, hi-IN) and script (e.g., Latin,

Devanagari) of the input text, supporting multiple languages.

tags:

- - subpackage\_text

parameters:

- name: api-subscription-key

in: header

required: true

schema:

type: string

responses:

'200':

description: Successful Response

content:

application/json:

schema:

$ref: >-

#/components/schemas/Sarvam\_Model\_API\_LanguageIdentificationResponse

'400':

description: Bad Request

content: {}

'403':

description: Forbidden

content: {}

'422':

description: Unprocessable Entity

content: {}

'429':

description: Quota Exceeded

content: {}

'500':

description: Internal Server Error

content: {}

requestBody:

content:

application/json:

schema:

$ref: >-

#/components/schemas/Sarvam\_Model\_API\_LanguageIdentificationRequest

components:

schemas:

Sarvam\_Model\_API\_LanguageIdentificationRequest:

type: object

properties:

input:

type: string

required:

- input

Sarvam\_Model\_API\_LanguageIdentificationResponse:

type: object

properties:

request\_id:

type:

- string

- 'null'

language\_code:

type:

- string

- 'null'

script\_code:

type:

- string

- 'null'

required:

- request\_id

```

## SDK Code Examples

```python

from sarvamai import SarvamAI

client = SarvamAI(

api\_subscription\_key="YOUR\_API\_SUBSCRIPTION\_KEY",

)

client.text.identify\_language(

input="input",

)

```

```typescript

import { SarvamAIClient } from "sarvamai";

const client = new SarvamAIClient({ apiSubscriptionKey: "YOUR\_API\_SUBSCRIPTION\_KEY" });

await client.text.identifyLanguage({

input: "input"

});

```

```go

package main

import (

"fmt"

"strings"

"net/http"

"io"

)

func main() {

url := "https://api.sarvam.ai/text-lid"

payload := strings.NewReader("{\n \"input\": \"foo\"\n}")

req, \_ := http.NewRequest("POST", url, payload)

req.Header.Add("api-subscription-key", "<apiKey>")

req.Header.Add("Content-Type", "application/json")

res, \_ := http.DefaultClient.Do(req)

defer res.Body.Close()

body, \_ := io.ReadAll(res.Body)

fmt.Println(res)

fmt.Println(string(body))

}

```

```ruby

require 'uri'

require 'net/http'

url = URI("https://api.sarvam.ai/text-lid")

http = Net::HTTP.new(url.host, url.port)

http.use\_ssl = true

request = Net::HTTP::Post.new(url)

request["api-subscription-key"] = '<apiKey>'

request["Content-Type"] = 'application/json'

request.body = "{\n \"input\": \"foo\"\n}"

response = http.request(request)

puts response.read\_body

```

```java

HttpResponse<String> response = Unirest.post("https://api.sarvam.ai/text-lid")

.header("api-subscription-key", "<apiKey>")

.header("Content-Type", "application/json")

.body("{\n \"input\": \"foo\"\n}")

.asString();

```

```php

<?php

$client = new \GuzzleHttp\Client();

$response = $client->request('POST', 'https://api.sarvam.ai/text-lid', [

'body' => '{

"input": "foo"

}',

'headers' => [

'Content-Type' => 'application/json',

'api-subscription-key' => '<apiKey>',

],

]);

echo $response->getBody();

```

```csharp

var client = new RestClient("https://api.sarvam.ai/text-lid");

var request = new RestRequest(Method.POST);

request.AddHeader("api-subscription-key", "<apiKey>");

request.AddHeader("Content-Type", "application/json");

request.AddParameter("application/json", "{\n \"input\": \"foo\"\n}", ParameterType.RequestBody);

IRestResponse response = client.Execute(request);

```

```swift

import Foundation

let headers = [

"api-subscription-key": "<apiKey>",

"Content-Type": "application/json"

]

let parameters = ["input": "foo"] as [String : Any]

let postData = JSONSerialization.data(withJSONObject: parameters, options: [])

let request = NSMutableURLRequest(url: NSURL(string: "https://api.sarvam.ai/text-lid")! as URL,

cachePolicy: .useProtocolCachePolicy,

timeoutInterval: 10.0)

request.httpMethod = "POST"

request.allHTTPHeaderFields = headers

request.httpBody = postData as Data

let session = URLSession.shared

let dataTask = session.dataTask(with: request as URLRequest, completionHandler: { (data, response, error) -> Void in

if (error != nil) {

print(error as Any)

} else {

let httpResponse = response as? HTTPURLResponse

print(httpResponse)

}

})

dataTask.resume()

```

# Transliterate Text

POST https://api.sarvam.ai/transliterate

Content-Type: application/json

\*\*Transliteration\*\* converts text from one script to another while preserving the original pronunciation. For example, \*\*'नमस्ते'\*\* becomes \*\*'namaste'\*\* in English, and \*\*'how are you'\*\* can be written as \*\*'हाउ आर यू'\*\* in Devanagari. This process ensures that the sound of the original text remains intact, even when written in a different script.

Transliteration is useful when you want to represent words phonetically across different writing systems, such as converting \*\*'मैं ऑफिस जा रहा हूँ'\*\* to \*\*'main office ja raha hun'\*\* in English letters.

\*\*Translation\*\*, on the other hand, converts text from one language to another while preserving the meaning rather than pronunciation. For example, \*\*'मैं ऑफिस जा रहा हूँ'\*\* translates to \*\*'I am going to the office'\*\* in English, changing both the script and the language while conveying the intended message.

### Examples of \*\*Transliteration\*\*:

- \*\*'Good morning'\*\* becomes \*\*'गुड मॉर्निंग'\*\* in Hindi, where the pronunciation is preserved but the meaning is not translated.

- \*\*'सुप्रभात'\*\* becomes \*\*'suprabhat'\*\* in English.

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 [Transliterate API Tutorial](https://github.com/sarvamai/sarvam-ai-cookbook/blob/main/notebooks/transliterate/Transliterate\_API\_Tutorial.ipynb).

Reference: https://docs.sarvam.ai/api-reference-docs/text/transliterate

## OpenAPI Specification

```yaml

openapi: 3.1.1

info:

title: Transliterate Text

version: endpoint\_text.transliterate

paths:

/transliterate:

post:

operationId: transliterate

summary: Transliterate Text

description: >-

\*\*Transliteration\*\* converts text from one script to another while

preserving the original pronunciation. For example, \*\*'नमस्ते'\*\* becomes

\*\*'namaste'\*\* in English, and \*\*'how are you'\*\* can be written as \*\*'हाउ

आर यू'\*\* in Devanagari. This process ensures that the sound of the

original text remains intact, even when written in a different script.

Transliteration is useful when you want to represent words phonetically

across different writing systems, such as converting \*\*'मैं ऑफिस जा रहा

हूँ'\*\* to \*\*'main office ja raha hun'\*\* in English letters.

\*\*Translation\*\*, on the other hand, converts text from one language to

another while preserving the meaning rather than pronunciation. For

example, \*\*'मैं ऑफिस जा रहा हूँ'\*\* translates to \*\*'I am going to the

office'\*\* in English, changing both the script and the language while

conveying the intended message.

### Examples of \*\*Transliteration\*\*:

- \*\*'Good morning'\*\* becomes \*\*'गुड मॉर्निंग'\*\* in Hindi, where the

pronunciation is preserved but the meaning is not translated.

- \*\*'सुप्रभात'\*\* becomes \*\*'suprabhat'\*\* in English.

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

[Transliterate API

Tutorial](https://github.com/sarvamai/sarvam-ai-cookbook/blob/main/notebooks/transliterate/Transliterate\_API\_Tutorial.ipynb).

tags:

- - subpackage\_text

parameters:

- name: api-subscription-key

in: header

required: true

schema:

type: string

responses:

'200':

description: Successful Response

content:

application/json:

schema:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TransliterationResponse'

'400':

description: Bad Request

content: {}

'403':

description: Forbidden

content: {}

'422':

description: Unprocessable Entity

content: {}

'429':

description: Quota Exceeded

content: {}

'500':

description: Internal Server Error

content: {}

requestBody:

content:

application/json:

schema:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TransliterationRequest'

components:

schemas:

Sarvam\_Model\_API\_TransliterateSourceLanguage:

type: string

enum:

- value: auto

- value: bn-IN

- value: en-IN

- value: gu-IN

- value: hi-IN

- value: kn-IN

- value: ml-IN

- value: mr-IN

- value: od-IN

- value: pa-IN

- value: ta-IN

- value: te-IN

Sarvam\_Model\_API\_TranslatiterateTargetLanguage:

type: string

enum:

- value: bn-IN

- value: en-IN

- value: gu-IN

- value: hi-IN

- value: kn-IN

- value: ml-IN

- value: mr-IN

- value: od-IN

- value: pa-IN

- value: ta-IN

- value: te-IN

Sarvam\_Model\_API\_NumeralsFormat:

type: string

enum:

- value: international

- value: native

Sarvam\_Model\_API\_SpokenFormNumeralsFormat:

type: string

enum:

- value: english

- value: native

Sarvam\_Model\_API\_TransliterationRequest:

type: object

properties:

input:

type: string

source\_language\_code:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TransliterateSourceLanguage'

target\_language\_code:

$ref: '#/components/schemas/Sarvam\_Model\_API\_TranslatiterateTargetLanguage'

numerals\_format:

$ref: '#/components/schemas/Sarvam\_Model\_API\_NumeralsFormat'

spoken\_form\_numerals\_language:

$ref: '#/components/schemas/Sarvam\_Model\_API\_SpokenFormNumeralsFormat'

spoken\_form:

type: boolean

required:

- input

- source\_language\_code

- target\_language\_code

Sarvam\_Model\_API\_TransliterationResponse:

type: object

properties:

request\_id:

type:

- string

- 'null'

transliterated\_text:

type: string

source\_language\_code:

type: string

required:

- request\_id

- transliterated\_text

- source\_language\_code

```

## SDK Code Examples

```python

from sarvamai import SarvamAI

client = SarvamAI(

api\_subscription\_key="YOUR\_API\_SUBSCRIPTION\_KEY",

)

client.text.transliterate(

input="input",

source\_language\_code="auto",

target\_language\_code="bn-IN",

)

```

```typescript

import { SarvamAIClient } from "sarvamai";

const client = new SarvamAIClient({ apiSubscriptionKey: "YOUR\_API\_SUBSCRIPTION\_KEY" });

await client.text.transliterate({

input: "input",

source\_language\_code: "auto",

target\_language\_code: "bn-IN"

});

```

```go

package main

import (

"fmt"

"strings"

"net/http"

"io"

)

func main() {

url := "https://api.sarvam.ai/transliterate"

payload := strings.NewReader("{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}")

req, \_ := http.NewRequest("POST", url, payload)

req.Header.Add("api-subscription-key", "<apiKey>")

req.Header.Add("Content-Type", "application/json")

res, \_ := http.DefaultClient.Do(req)

defer res.Body.Close()

body, \_ := io.ReadAll(res.Body)

fmt.Println(res)

fmt.Println(string(body))

}

```

```ruby

require 'uri'

require 'net/http'

url = URI("https://api.sarvam.ai/transliterate")

http = Net::HTTP.new(url.host, url.port)

http.use\_ssl = true

request = Net::HTTP::Post.new(url)

request["api-subscription-key"] = '<apiKey>'

request["Content-Type"] = 'application/json'

request.body = "{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}"

response = http.request(request)

puts response.read\_body

```

```java

HttpResponse<String> response = Unirest.post("https://api.sarvam.ai/transliterate")

.header("api-subscription-key", "<apiKey>")

.header("Content-Type", "application/json")

.body("{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}")

.asString();

```

```php

<?php

$client = new \GuzzleHttp\Client();

$response = $client->request('POST', 'https://api.sarvam.ai/transliterate', [

'body' => '{

"input": "foo",

"source\_language\_code": "auto",

"target\_language\_code": "bn-IN"

}',

'headers' => [

'Content-Type' => 'application/json',

'api-subscription-key' => '<apiKey>',

],

]);

echo $response->getBody();

```

```csharp

var client = new RestClient("https://api.sarvam.ai/transliterate");

var request = new RestRequest(Method.POST);

request.AddHeader("api-subscription-key", "<apiKey>");

request.AddHeader("Content-Type", "application/json");

request.AddParameter("application/json", "{\n \"input\": \"foo\",\n \"source\_language\_code\": \"auto\",\n \"target\_language\_code\": \"bn-IN\"\n}", ParameterType.RequestBody);

IRestResponse response = client.Execute(request);

```

```swift

import Foundation

let headers = [

"api-subscription-key": "<apiKey>",

"Content-Type": "application/json"

]

let parameters = [

"input": "foo",

"source\_language\_code": "auto",

"target\_language\_code": "bn-IN"

] as [String : Any]

let postData = JSONSerialization.data(withJSONObject: parameters, options: [])

let request = NSMutableURLRequest(url: NSURL(string: "https://api.sarvam.ai/transliterate")! as URL,

cachePolicy: .useProtocolCachePolicy,

timeoutInterval: 10.0)

request.httpMethod = "POST"

request.allHTTPHeaderFields = headers

request.httpBody = postData as Data

let session = URLSession.shared

let dataTask = session.dataTask(with: request as URLRequest, completionHandler: { (data, response, error) -> Void in

if (error != nil) {

print(error as Any)

} else {

let httpResponse = response as? HTTPURLResponse

print(httpResponse)

}

})

dataTask.resume()

```