



# One Hour Translation API V2 - Developer Guide

v1.1

## Contents

One Hour Translation API V2 - Developer Guide .....	1
General .....	4
Request URLs: .....	4
Authentication: .....	4
Response Format: .....	4
Callbacks: .....	4
Development Friendly: .....	4
Methods Documentation.....	5
Account > Get account details .....	5
Resources > Create a text resource.....	5
Resources > Create a new file resource .....	5
Resources > Download resource .....	6
Tools > Get Quote .....	7
Tools > Get Word Count .....	7
Project > Create a new project.....	7
Proofreading (source and translated docs):.....	8
Transcription: .....	9
Project > Get project details .....	10
Project > Cancel project.....	10
Project > Comments > Get comments .....	11
Project > Comments > New comment.....	11
Machine Translation > Translate .....	12
Machine Translation > Detect Language.....	12
Discover > Supported Languages .....	13
Discover > Supported Language Pairs.....	13
Discover > Supported Expertise.....	14
One Hour Translation - Translation Memory (OHT - TM) .....	15
Concept:.....	15
Methods: .....	16
1. Create new Context: .....	16
2. List of Contexts .....	17

3.	Get details of a specific Context:.....	18
4.	Delete a specific context and its phrases .....	19
5.	Create phrase .....	20
6.	List Phrases (Search): .....	21
7.	Get specific phrase: .....	22
8.	Update specific phrase: .....	23
9.	Delete specific phrase:.....	24
10.	Request Context Translation .....	25
	Callbacks.....	26
	Callback Events: .....	26
	Content Formats.....	27
	Plain Text.....	27
	Parameters or Formatted Text .....	27
	XML .....	27
	Sandbox Usage.....	28
	Status and Error Codes .....	29
	Language Code.....	33
	Expertise Codes .....	37
	Developer Tools .....	37
	Client Libraries:.....	37
	Testing and Experimenting: .....	37

## **General**

One Hour Translation API expose access to various methods and services delivered by One Hour Translation.

### Request URLs:

API Base URL: <https://www.onehourtranslation.com/api/2>

API Sandbox URL: <https://sandbox.onehourtranslation.com/api/2>

### Authentication:

Each request must include the `public_key` and `secret_key` parameters, as supplied by One Hour Translation.

To get your API keys on the Production environment, please [click here](#)

To get your API keys on the Sandbox environment, please [click here](#)

### Response Format:

Responses are encoded according to the [JSON scheme](#), Long and complex values are further encoded with [Base64](#).

Each response will include the “`status`” entity which represent a success or failure in processing your request. The complete status codes list may be viewed [here](#).

Response example:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "errors": [ /*array of errors*/ ],
  "results": [ /*method-specific parameters as specified below*/ ]
}
```

### Callbacks:

Some requests allow you to specify a callback URL. This URL will be used on time critical events, for example, when a translation has been received for your project.

Read more about callbacks in the [“Callbacks” section below](#).

### Development Friendly:

We have developed some tools that may accelerate your development and testing dramatically.

Please consult the [“Developer Tools”](#) section.

## **Methods Documentation**

### Account > Get account details

GET: /account

Auth: full

Request Params:

Response Params:

- credits
- account\_id
- account\_username

### Resources > Create a text resource

POST: /resources/text

Auth: full

Request Params:

- text

Response Params:

- resource\_uuid

### Resources > Create a new file resource

POST: /resources/file

Auth: full

Request Params:

- file\_name
- file\_mime - optional
- @upload (http file upload)

Response Params:

- resource\_uuid

### Resources > Get resource

GET: /resources/<resource\_uuid>

Auth: full

Request Params:

- fetch - (optional) possible values:
  - false** - (default) do not fetch content
  - base64 - fetch content, base64 encoded
- project\_id - (optional) Project ID, needed when requesting a resource that was uploaded by another user - e.g. as a project's translation .

Response Params:

- type - text | file
- length
- file\_name - only for files
- file\_mime - only for files
- download\_url - url to download as file (currently str link to the API call "Download resource"...) )
- content - base64 encoded (only if fetch="base64")

### Resources > Download resource

GET: /resources/<resource\_uuid>/download

Auth: full

Request Params:

- **project\_id** = (optional) Project ID, needed when requesting a resource that was uploaded by another user - e.g. as a project's translation .

Response Params:

Return a "file download" response, not JSON. See [here](#).

File should have original name, text content should be downloaded as a file with filename: "oht\_<resource\_uuid>.txt"

### Tools > Get Quote

GET: /tools/quote

Auth: full

#### Request Params:

- resources = array of resource\_uuid
- wordcount = integer, words count
- expertise = optional, 0 | expertise\_slug
- proofreading = optional, 0 | 1
- currency = optional, **USD** | EUR

#### Response Params:

- currency - currency selected by user (or default)
- resources - array of results per resource
  - resource\_uuid
  - wordcount
  - credits
  - price
- total
  - wordcount - total words count
  - credits - sum of credits to charge
  - net\_price - price in selected currency, based on credits and discounts
  - transaction\_fee - price in selected currency, based on fee from payment vendors
  - price - total price in selected currency, based on net\_price and transaction\_fee.

### Tools > Get Word Count

GET: /tools/wordcount

Auth: full

#### Request Params:

- resources = array of resource\_uuid

#### Response Params:

- resources - array of results per resource
  - resource\_uuid
- total
  - wordcount - total words count

### Project > Create a new project

POST: /projects  
POST: /projects/<project\_type>  
Auth: full

*Translation Project:*

Request Params:

- project\_type - “**translation**” - default
- source\_language - [language code](#)
- target\_language - [language code](#)
- sources - comma separated list of Resource UUIDs
- wordcount - (optional) integer, if empty use automatic counting
- notes - (optional)
- expertise - (optional) See “[expertise codes](#)” section
- callback\_url - (optional), see “[Callbacks](#)” section
- custom0..9 - (optional)

Response Params:

- project\_id
- wordcount
- credits

*Proofreading (only source docs):*

Request Params:

- project\_type - “proof-general”
- source\_language- [language code](#)
- sources - comma separated list of Resource UUIDs
- wordcount - (optional) integer, if empty use automatic counting
- notes - (optional)
- callback\_url - (optional), see “[Callbacks](#)” section
- custom0..9 - (optional)

Response Params:

- project\_id
- wordcount
- credits

Proofreading (source and translated docs):

Request Params:

- project\_type - “proof-translated”
- source\_language - [language code](#)



- target\_language - [language code](#)
- sources - comma separated list of Resource UUIDs
- translations - comma separated list of Resource UUIDs
- wordcount - (optional) integer, if empty use automatic counting
- notes - (optional)
- callback\_url - (optional), see "[Callbacks](#)" section
- custom0..9 - (optional)

Response Params:

- project\_id
- wordcount
- credits

Transcription:

Request Params:

- project\_type - "transcription"
- source\_language - [language code](#)
- sources - comma separated list of Resource UUIDs
- length - (optional) integer of seconds, if empty use automatic counting
- notes - (optional)
- callback\_url - (optional), see "[Callbacks](#)" section
- custom0..9 - (optional)

Response Params:

- project\_id
- length
- credits

Project > Get project details

GET: /projects/<project\_id>

Auth: full

Request Params:

Response Params:

- project\_id
- project\_type
- source\_language - [language code](#)
- target\_language - [language code](#)
- resources
  - sources
  - translations | proofs | transcriptions
- wordcount | length (seconds, transcription projects only)
- custom 0..9

Project > Cancel project

DELETE: /projects/<project\_id>

Auth: full

Constraints:

- Available only before actual translation starts

Request Params:

Response Params:

Project > Comments > Get comments

GET: /projects/<project\_id>/comments  
Auth: full

Request Params:

Response Params:

- comments - array of comments
  - id
  - date
  - commenter\_name - Short representation of the user's name
  - commenter\_role - admin | customer | provider | potential-provider
    - admin = One Hour Translation support team
    - provider = The translator / proofreader / transcriber that is assigned to the project
    - potential-provider = A translator / proofreader / transcriber that was allowed to view the project before it was assigned
  - content - text content

Project > Comments > New comment

POST: /projects/<project\_id>/comments  
Auth: full

Request Params:

- content - text content

Response Params:

### Machine Translation > Translate

GET,POST: /mt/translate/text

GET,POST: /mt/translate/json

GET,POST: /mt/translate/xml

Auth: full

Request Params:

- source\_language - [language code](#)
- target\_language - [language code](#)
- source\_content

Response Params:

- translation - the translated content in the original format

### Machine Translation > Detect Language

GET,POST: /mt/detect/text

GET,POST: /mt/detect/json

GET,POST: /mt/detect/xml

Auth: full

Request Params:

- source\_content

Response Params:

- language - [language code](#)

### Discover > Supported Languages

GET: /discover/languages

Auth: half

Response Params:

- Array - ordered by name field (A->Z)
  - language\_name
  - language\_code - [language code](#)

**response example :**

```
[
  {
    "name" : "English",
    "code" : "en-us"
  },
  {
    "name" : "Arabic",
    "code" : "ar-sa"
  }
]
```

### Discover > Supported Language Pairs

GET: /discover/languagespairs

Auth: half

Response Params:

- Array - ordered by source name field (A->Z)
  - source -
    - name
    - code - [language code](#)
  - targets - Array of languages:
    - name
    - code - [language code](#)

Response example :

```
[ {
  "source" : { "name" : "English",
               "code" : "en-us"
             },
  "targets" : [ { "name" : "French",
                  "code" : "fr-fr"
                },
                ]
}
```

```
{ "name" : "Dutch",  
  "code" : "nl-nl"  
}  
]  
}
```

### Discover > Supported Expertise

GET: /discover/expertise

Auth: half

#### Request Params:

- source\_language - [language code](#) (optional, mandatory if target\_language is specified)
- target\_language - [language code](#) (optional, mandatory if source\_language is specified)

#### Response Params:

- Array - ordered by name field (A->Z)
  - expertise\_name
  - expertise\_code

#### Response example :

```
[  
  { "name" : "Automotive/Aerospace",  
    "code" : "330"  
  },  
  { "name" : "Business/Forex/Finance",  
    "code" : "331"  
  }  
]
```

# One Hour Translation - Translation Memory (OHT - TM)

## **Concept:**

Nowadays, customers worldwide utilize several distinct methods of localizing their apps, websites and softwares. Those methods include:

- Localization files (po, resx, xml, etc)
- Homemade internal localization systems
- Several open-source systems

Using the above methods cause huge overhead to customers that need to update their products often, by re-running the translation-implementation cycle over and over again and usually requires special in house personnel that needs to process new translation requirements, monitor the translation process and implement back in the product.

OHT-TM service is intended to help dynamic and fast growing customers by:

- Serving as a central repository for all product resources and translations
- Semi-automatic translation process that keeps all resources translated
- Simple import/export of resources
- Simple API for inserting new resources and fetching translations

On the practical level, OHT-TM is structured as a collection of “Contexts”.

Each Context is a collection of “Phrases” that typically belong to the same domain, each phrase contains a source string and its translations to multiple languages.

For example, when localizing an application, each string is represented as a Phrase and all phrases belong to a single Context. This makes it easy to add new strings continuously and export a complete translated repository whenever needed.

## **Methods:**

### **1. Create new Context:**

POST: /tm/context

Params:

- context\_name - optional
- parent\_context - optional

### **Examples:**

URL: <http://www.onehourtranslation.com/api/2/tm/context>

Params:

```
secret_key: <secret_key>
public_key: <public_key>
context_name: c1
```

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": {
    "acl": {
      "public_write": 0,
      "public_read": 0
    },
    "uuid": "c-773531d1ae26",
    "parent": "",
    "name": "c1"
  },
  "errors": []
}
```



## 2. List of Contexts

GET: /tm/context

Examples:

URL:

`http://www.onehourtranslation.com/api/2/tm/context?public_key=<public_key>&secret_key=<secret_key>`

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": [
    {
      "acl": {
        "public_write": 0,
        "public_read": 0
      },
      "uuid": "c-e8a7436b5cae",
      "parent": "c-3dfd4e6b5cad",
      "name": "c1"
    },
    {
      "acl": {
        "public_write": 0,
        "public_read": 0
      },
      "uuid": "c-0ab1a4e4a491",
      "parent": "c-e8a7436b5cae",
      "name": "c2"
    }
  ],
  "errors": []
}
```

**3. Get details of a specific Context:**

GET: /tm/context/<context\_uuid>

Example:

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26>

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": [
    {
      "acl": {
        "public_write": 0,
        "public_read": 0
      },
      "uuid": "c-773531d1ae26",
      "parent": "",
      "name": "c1"
    }
  ],
  "errors": []
}
```

#### 4. Delete a specific context and its phrases

DELETE: /tm/context/<context\_uuid>

Example:

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-e8a7436b5cae>

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "errors": [],
  "results": []
}
```

## 5. Create phrase

POST: /tm/context/<context\_uuid>/phrases

Params:

- source\_language
- source\_text
- phrase\_key - optional
- target\_language- optional
- target\_text - optional
- flags - **Not Ready**
- invalidate - relevant only when "target\_language" and "target\_text" specified.
- remarks

### Example:

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26/phrases>

Params:

```
secret_key: <secret_key>
public_key: <public_key>
source_lang: english
source_text: hello world
remarks: example
```

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": {
    "source_text": "hello world",
    "uuid": "p-93effa9e7620",
    "context": "c-773531d1ae26",
    "target": [],
    "remarks": "example",
    "flags": null,
    "source_language": "english",
    "phrase_key": "2aae6c35c94fcfb415dbe95f408b9ce91ee846ed"
  },
  "errors": []
}
```

## 6. List Phrases (Search):

GET: /tm/context/<context\_uuid>/phrases

GET: /tm/context/<context\_uuid>/phrases/<source\_language>

GET:

/tm/context/<context\_uuid>/phrases/<source\_language>/to/<target\_language>

GET: /tm/context/<context\_uuid>/phrases?source\_text=<source text>

### Examples:

URL:

<http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26/phrases/english>

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": [
    {
      "source_text": "Hello World",
      "uuid": "p-93effa9e7620",
      "context": "c-773531d1ae26",
      "target": [],
      "remarks": "example",
      "flags": null,
      "source_language": "english",
      "phrase_key":
"2aae6c35c94fcfb415dbe95f408b9ce91ee846ed"
    },
    {
      "source_text": "Good day!",
      "uuid": "p-f7250f042e6a",
      "context": "c-773531d1ae26",
      "target": [],
      "remarks": "",
      "flags": null,
      "source_language": "english",
      "phrase_key":
"3fd4f11f202ff82750dc4a133aa67747832388fe"
    }
  ],
  "errors": []
}
```

## 7. Get specific phrase:

GET: /tm/context/<context\_uuid>/phrase/<phrase\_key>

### Example:

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26/phrase/3fd4f11f202ff82750dc4a133aa67747832388fe>

### Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": {
    "source_text": "Good day",
    "uuid": "p-f7250f042e6a",
    "context": "c-773531d1ae26",
    "target": {
      "spanish": [
        "buen día"
      ]
    },
    "remarks": "",
    "flags": null,
    "source_language": "english",
    "phrase_key": "3fd4f11f202ff82750dc4a133aa67747832388fe"
  },
  "errors": []
}
```

## 8. Update specific phrase:

POST: /tm/context/<context\_uuid>/phrase/<phrase\_key>

Params:

- source\_text
- target\_language - optional
- target\_text - optional
- invalidate - relevant only when "target\_language" specified.
- remarks

Example:

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26/phrase/3fd4f11f202ff82750dc4a133aa67747832388fe>

Params:

```
secret_key: <secret_key>
public_key:<public_key>
target_text: يوم سعيد.
target_lang: arabic
```

Response:

```
{
  "status": {
    "code": 0,
    "msg": []
  },
  "results": {
    "source_text": "Good day",
    "uuid": "p-f7250f042e6a",
    "context": "c-773531d1ae26",
    "target": {
      "arabic": [
        "يوم سعيد."
      ],
      "spanish": [
        "buen día"
      ]
    },
    "remarks": "",
    "flags": null,
    "source_language": "english",
    "phrase_key":
"3fd4f11f202ff82750dc4a133aa67747832388fe"
  },
  "errors": []
}
```

**9. Delete specific phrase:**

DELETE: /tm/context/<context\_uuid>/phrase/<phrase\_key>

**Example:**

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26/phrase/2aae6c35c94fcfb415dbe95f408b9ce91ee846ed>

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": [],
  "errors": []
}
```



## 10. Request Context Translation

This request initiates a translation project to translate all non-translated strings. Alternatively, you can specify a list of phrases to translate.

POST: /tm/context/<context\_uuid>/translate/<source\_language>/to/<target\_language>

Params:

- retranslate = (optional, default=false) - when true, retranslate all, including translated phrases. Otherwise, translate only untranslated / invalidated phrases
- phrase\_keys[] - (optional) array of specific phrases to translate, automatic include translated
- callback\_url
- custom0..9

Example:

URL: <http://www.onehourtranslation.com/api/2/tm/context/c-773531d1ae26/translate/english/to/arabic>

Params:

```
secret_key:<secret_key>
public_key:<public_key>
retranslate: true
```

Response:

```
{
  "status": {
    "code": 0,
    "msg": "ok"
  },
  "results": {
    "project_id": "196",
    "word_count": 2,
    "credits": 14,
  },
  "errors": []
}
```

## **Callbacks**

Some methods allow you to specify a callback URL. This URL will be used on time critical events, for example, when a translation has been received on your project.

Callback URL must be a valid and web-accessible location and must return an HTTP "200 OK" status.

All callbacks are logged and you can view the log for a specific project on the project's page.

### **Callback Events:**

1. Project > Resource Submitted:

Method: POST

Request Params:

- event = project.resources.new
- project\_id - the specific project ID
- custom0..9 - custom fields that were submitted on project creation, if any.
- resource\_uuid - UUID of the newly submitted resource
- resource\_type = translation | proof | transcription

2. Project > Comment Submitted:

Method: POST

Request Params:

- event = project.comments.new
- project\_id - the specific project ID
- custom0..9 - custom fields that were submitted on project creation, if any.
- comment\_date
- commenter\_name - A short version of the commenter name
- commenter\_role - customer | translator | proofreader | admin
- comment\_content - text

## **Content Formats**

### Plain Text

The most simple, straightforward method of submitting text for translation is just by submitting the text itself.

The text will be received as-is and will not be parsed or processed in any way.

Once translated, the translation will be submitted to the customer as plain text as well.

To submit text for translation, first create a “text resource” with the “[Create a new resource \(text\)](#)” method and attach the created resource to a project.

### Parameters or Formatted Text

For translating parameterized content we recommend using [XLIFF](#) formatted text.

XLIFF is an XML-based format for representing translatable content. It has become an industry standard for exchanging source and translated contents and for translation localization purposes.

To submit XLIFF for translation, first create a “file resource” with the “[Create a new resource \(file\)](#)” method and attach the created resource to a project.

Please make sure to name your file with `.xliiff` extension.

More information about XLIFF standard can be found here: <http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html>

### XML

One Hour Translation supports translation of XML document of any type. Please note the following when submitting XML for translation:

1. We accept any XML format you submit and translate only the texts within the XML tags.
2. Tag names and attributes are not translated.
3. The translated XML will maintain the same format and hierarchy of the original XML.
4. Submitted XML must be well formed. If your XML contains non-standard or XML-reserved characters, we encourage you to wrap it with CDATA notation.
5. You may also submit HTML for translation. The best way to do so is to wrap it inside a XML CDATA element and add a ' type="text/html" ' attribute to the containing element.

To submit XML for translation, first create a “file resource” with the “[Create a new resource \(file\)](#)” method and attach the created resource to a project.

Please make sure to name your file with `.xm1` extension.

## **Sandbox Usage**

OneHourTranslation provides a separated, semi-functional version on One Hour Translation for testing and development purposes. The sandbox environment may be accessed here:

<http://sandbox.onehourtranslation.com>

In order to use One Hour Translation sandbox, please note following:

1. The sandbox is fully functional as a website and API interface but is completely non functional for translation purposes or any other professional services.
2. To get started, please [create a user account](#) on One Hour Translation sandbox. Once an account is established, create your API keys [here](#).
3. Every text you submit for translation will be translated within a few seconds using Google's machine translation service. XML formatted text will not be translated but will rather be copied and returned to your system as a translation.
4. Note: This behavior is limited to the sandbox only. Texts and XMLs you submit to OneHourTranslation.com will only be translated by real human translators.
5. The sandbox environment only supports **English to French** translation at the moment. Other language pair will invoke a "language pair is currently unsupported" (#201) error code.

## Status and Error Codes

Code	Message	Remarks/Error
0	"ok"	
<b>General Request Errors</b>		
100	"Malformed or not supported request"	
101	"Unauthorized request - please include all your authentication fields"	specific missing parameter : public/private key
102	"Forbidden - you are not allowed to perform this request or api key has been revoked"	
103	"Request validation failed"	Usually followed by specific field error messages (source/target lang)
104	"Item is missing or you are not authorized"	Usually followed by specific unauthorized item, (context_id
105	"action is forbidden"	Usually followed by a

	because of certain dependencies”	specific item that can't be deleted ( context_id)
106	“Comments have been disabled for this project status”	
107	“Cannot cancel project in this status”	
108	“Please select the same language as source and target on general proofreading projects”	
109	Wordcount is zero, cannot create project	
<b>Project Related Errors</b>		
200	"insufficient credits to fulfill your request - action required"	
201	"language pair is currently unsupported"	
202	"unknown project id or you are not	

	authorized to access this project"	
203	"translation has not been submitted for this project yet"	
204	"No phrases found for translation project"	override default by using retranslation=true
205	Language pair is currently not supported for requested expertise or expertise is not supported	
<b>Machine Translation</b>		
301	language detection failed	
302	"language pair is currently not supported for machine translation"	
<b>Resources Related Errors</b>		
401	Unable to upload file: file is empty	
402	Unable to upload file: file size is not supported	

<b>Internal Errors</b>		
900	"unexpected internal error"	



## Language Code

Language codes are encoded using the following mechanism:

1. Two letter [ISO 639.2](#) language code. This is the minimal language representation.
2. Two letter regional (country) code according to [ISO 3166](#) (optional). Most language are also represented using this code.
3. Three letter dialect representation according to the [SIL-List](#) (optional). Mostly used for Chinese dialects.
4. 's' or 't' for "Simplified" or "Traditional" Chinese (optional).

Language	Code
Albanian	sq-al
Arabic	ar-sa
Bengali	bn-bd
Bosnian	bs-ba
Bulgarian	bg-bg
Catalan	ca-es
Chinese Cantonese	zh-cn-yue
Chinese Mandarin-Simplified	zh-cn-cmn-s
Chinese Mandarin-Traditional	zh-cn-cmn-t
Croatian	hr-hr
Czech	cs-cz
Danish	da-dk
Dari	fa-af

Dutch	nl-nl
English UK	en-uk
English US	en-us
Estonian	et-ee
Farsi	fa-ir
Finnish	fi-fi
French	fr-fr
French Canadian	fr-ca
Georgian	ka-ge
German	de-de
Greek	el-gr
Hebrew	he-il
Hindi	hi-in
Hungarian	hu-hu
Indonesian	id-id
Italian	it-it
Japanese	jp-jp
Kazakh	kk-kz
Khmer	km-kh
Korean	ko-kp

Lao	lo-la
Latvian	lv-lv
Lithuanian	lt-lt
Macedonian	mk-mk
Malay	ms-my
Norwegian	no-no
Pashto	ps
Polish	pl-pl
Portuguese Brazil	pt-br
Portuguese Portugal	pt-pt
Romanian	ro-ro
Russian	ru-ru
Serbian	sr-rs
Slovak	sk-sk
Slovenian (Slovene)	sl-si
Spanish	es-es
Spanish Latin-America	es-ar
Swedish	sv-se
Tagalog	tl-ph
Tamil	ta-in

Thai	th-th
Turkish	tr-tr
Ukrainian	uk-ua
Urdu	ur
Uzbek	uz-uz
Vietnamese	vi-vn

## Expertise Codes

Name	Code
Automotive/Aerospace	automotive-aerospace
Business/Forex/Finance	business-finance
Software/IT	software-it
Legal/Certificates	legal-certificate
Marketing/Consumer/Media	marketing-consumer-media
CV	cv
Medical	medical
Patents	patents
Scientific	scientific
Technical/Engineering	technical-engineering
Gaming/Video Games	gaming-video-games
Ad-Words/Banners	ad-words-banners
Mobile Applications	mobile-applications

## Developer Tools

Here's a list of tools that will help you integrate with One Hour Translation API faster.

### Client Libraries:

PHP: <https://github.com/OneHourTranslation/API-2-Client-PHP>

### Testing and Experimenting:

1. [Postman](#): REST Client for Google Chrome browser.  
Download and import One Hour Translation client: