Worldwide Lexicon v2 Web API Documentation

This article describes how to get started with the latest version of the Worldwide Lexicon hybrid translation API, which is now fully REST oriented, and is tightly integrated with the SpeakLike professional translation platform. This API, which can be access directly, or via tools such as our Word Press translation plugin, makes it easy to embed best effort translation in a wide range of applications and publishing systems, and to blend machine, crowd and professional translation to optimize for quality, cost and turnaround time.

We are also developing an affiliate program for software developers which will enable you to earn a percentage of sales generated by any tools or applications you build around the open API. We track every professional translation and peer review job generated by your application and pay out a percentage of revenue. We will announce more details about this program in the near future.

This documentation is for the Spring 2011 release of the API, which is currently available for testing at worldwidelexicon2.appspot.com This will be migrated to the production service in April 2011, once compatibility testing is completed.

Table of Contents

Hello World : Requesting A Translation

Let's start out with a simple "Hello World" demonstration, to show just how easy it is to incorporate hybrid translation into your project. To request translations, you simply call the web API www.worldwidelexicon.org/t with the following parameters in a GET or POST form:

- sl : source language code
- tl : target language code
- st : source text (utf 8 encoding)
- url : URL of the source document the text came from (optional)
- allow_machine : use machine translation (y/n)
- ac : affiliate code (if you are building a third party tool, stamp requests with your affiliate code, e.g. abcsoft_joomla_plugin)
- output : json, text (JSON is the default output format)

If you wish to request a professional translation, simply include these extra parameters:

- lspusername : your SpeakLike login or email
- lsppw : your SpeakLike password or key
- callback_url : optional callback URL for SpeakLike to submit the translation to
- apikey : optional secret key to include in the callback
- output : output format (json by default)

The service will reply with a JSON record containing the best available translations, or an HTTP error. If you request a professional translation which has not yet been completed, and allow_machine is y, the service will respond with a placeholder machine translation until a professional translation is available. Thus you can instantly generate machine translations, which are then replaced by professional translations when they come in.

You can either poll the system for translations (please cache translations locally in memcached or a similar tool to avoid repetitive queries), or you can provide a callback URL for SpeakLike to submit the translation to when done. SpeakLike will mimic the /submit API POST request when calling you, so all you need to do is to implement the /submit API (described in the next section) to accept asynchronous callbacks.

Accepting Callback Submissions

If you include a callback_url field in your call to the /t web API, SpeakLike will submit the completed translation, and any followon edits, to your system at the URL you specify. The script that handles these requests should expect the following form fields:

- guid : globally unique record locator, assigned by SpeakLike
- sl : source language code
- tl : target language code
- st : source text (utf8)
- tt : translated text (utf8)
- url : URL the source text came from (optional)
- lsp : LSP name (e.g. speaklike)
- username : translator name or ID (optional)
- apikey : secret key provided in the initial request

Your service can then do whatever it needs to with the translation. If the submission is valid, reply with “ok”, otherwise reply with an HTTP error code.

We generally prefer that clients implement a callback mechanism if possible as this reduces the need to poll the translation server, and also provides you with immediate delivery as soon as the translation is completed.

Output Formats

We use JSON as the default output format when returning structured data, and encourage most developers to use this format (it is easy to convert into other formats within your system or applications). We do provide support for several translation specific output formats, notably PO (gettext) and XLIFF. We support PO files as an output format because they can be used as a drop in solution for applications that use gettext() to translate strings. We support XLIFF as well. We have deprecated support for other formats such as RSS.

Submitting Crowd / Community Translations

To submit a crowd or community translation to the translation memory, simply call the /submit API to do so, using the following parameters:

- sl : source language code
- tl : target language code
- st : source text (utf8)
- tt : translated text (utf8)
- url : URL text came from (optional)
- username : username
- facebookid : optional Facebook ID number (Facebook Connect is easy to implement and

is a good way to track submissions)
- profile_url : Facebook profile URL (optional)

The service will reply with “ok” or an error code. The service does not authenticate users, and assumes that you apply whatever access control policies you want prior to submitting to the translation memory.

Deleting Translations

Translations can be deleted by calling the /delete request handler. This is only allowed in two cases:

- The user who created the translation is requesting its deletion
- The user requesting deletion provided a secret key (e.g. is a trusted user)

The service responds with “ok” or an HTTP error.

Submitting User Generated Scores For Translations

To submit a score for a crowd / community translation, simply call the /scores API with a POST form containing the following fields:

- guid : globally unique record locator for the translation
- score : 0-5 (0=junk, 5=native quality)
- username : optional username or email for scorer

The service will reply with “ok” or an error message.

Request Peer Review For A Crowd Translation

You can automatically vet new translators on your system using the /lsp/score API call. This simple API enables you to submit a translation to be scored on a 0 to 5 scale by a professional translator. With a few scores for each new translator, you can develop an accurate profile of who the good and bad translators are and make your own allow/deny decisions based on this data. You can also use this to spot check translators over a longer period of time.

Simply call /lsp/score with the following parameters

- guid : unique record locator, generated by your system so you can match the returned score with the correct record
- sl : source language code

- tl : target language code
- st : source text (utf8)
- tt : translated text (utf8)
- url : URL text came from (optional)
- lspusername : your SpeakLike username/email
- lsppw : your SpeakLike password/key
- sla : service level agreement code (if other than standard level)
- callback_url : URL to submit the score to when done
- apikey : secret key to include in callback

The service will reply with “ok” or an error message (e.g. 401 : authentication failed). The service will generate a POST form to the callback_url with the following parameters when done:

- guid : record locator
- score : 0:5
- username : name/id of translator (optional)

You can then use this information to update the statistics for the user who generated the original translation.

Other Commands

The API calls mentioned above enable you to build a robust translation service with minimal effort. However, we provide additional tools for developers who want to dig deeper.

Fetching and Submitting Comments (/comments)

The /comments API enables you to fetch and submit comments about translations. It is simple to use. To request comments about a translation, just make an HTTP GET request to /comments with the fields:

- guid : translation record locator
- language : optional language code, to filter comments by language

It returns a JSON recordset with a list of comments, sorted newest to oldest.

To submit a comment about a translation, make an HTTP POST request to /comments with the fields:

- guid : translation record locator
- language : language code (language comment is written in)
- text : text of the comment

- username : optional username/email
- name : optional proper name of user

The service will reply with “ok” or an HTTP error code.

Requesting The Estimated Delivery Time For A Translation (/lsp/eta)

This API call enables you to obtain an estimated turnaround time for a project based on word count, language pair and service level agreement. We calculate the response time based on statistics for similar jobs that have been recently processed. We cannot guarantee that jobs will be completed within the estimated time, but in most cases, this will give you an idea of how long a particular job should take. Simply call /lsp/eta with the following fields in a GET request:

- lspusername : SpeakLike username or email
- lsppw : password or key
- sl : source language code
- tl : target language code
- words : word count
- sla : service level agreement code (if other than standard)

The service will reply with a JSON dictionary with the following fields:

- min_eta : minimum completion time
- average_eta : average completion time
- max_eta : maximum completion time

Fetching a Batch Of Completed Translations and Scores (/lsp/fetch)

If you application is using polling to retrieve completed jobs from the system, you can use the /lsp/fetch API to retrieve a batch of up to 100 completed translations or review scores per query. The service expects an HTTP GET with the following fields:

- lspusername : SpeakLike username/email
- lsppw : SpeakLike pw or key
- jobtype : optional job type (translation or score)
- sl : source language (optional)
- tl : target language (optional)

The service returns a JSON recordset with the following fields for each row:

- guid : record locator
- date : timestamp

- jobtype : ‘translation’ or ‘score’
- sl : source language
- tl : target language
- st : source text (utf8)
- tt : translated text (utf8)
- url : URL text came from
- username : translator name/id
- score : 0-5 (if jobtype=score, otherwise not present)

Accepting or Rejecting Translations

After you have fetched a list of translations or scores with the /lsp/fetch API, you then can signal whether to accept or reject translations.

- To accept a translation, you call /lsp/accept/{guid}
- To reject/redo a translation, you call /lsp/reject/{guid}

Deleting Translation and Score Requests

To abort a translation or score request, call /lsp/delete/{guid}

The service will reply with “ok” or an error message (for example if the job does not exist, is in progress, or has already been completed).

Fetching A List Of Pending Transactions (/lsp/queue)

To view a list of the pending transactions in your queue, simply call /lsp/queue with the following parameters in a GET request:

- lspusername : SpeakLike username/email
- lsppw : password or key
- jobtype : optional job type (‘translation’ or ‘score’)
- sl : source language
- tl : target language

The service will respond with a list of pending jobs, with the same JSON response as the /lsp/fetch command

Requesting An Instant Quote (/lsp/quote)

This API call enables you to get an instant quote based on language pair, word count and SLA. Simply call /lsp/quote with the following fields in a GET request:

- lspusername : SpeakLike username or email
- lsppw : password or key
- sl : source language
- tl : target language
- words : word count
- sla : SLA code, if other than standard

The service will respond with a JSON dictionary with the following fields:

- eta : estimated delivery time
- price : total price for job (USD)
- unit_price : per word price for job (USD)

Requesting Queue Statistics (/lsp/stats)

Under construction, this API call will provide summary statistics for your account, with an overview of currently queued jobs, completion times, and other metrics.

Revision History (/r)

You can request a revision history of translations for a specific source text by calling /r with the following fields in a GET request:

- sl : source language code
- tl : optional target language code
- st : source text (utf8)

The service will reply with a JSON recordset with the following fields for each row:

- guid : record locator
- sl : source language code
- tl : target language code
- st : source text
- tt : translated text
- url : URL text came from
- username : username/id of translator
- remote_addr : IP address of translator
- human : is human translation (True/False)
- professional : is professional translation (True/False)

Requesting A User's Score History (/scores/user)

You can request a user's score history, both averages and individual votes, by calling /scores/user/{username_or_ip_address}

The service will respond with a JSON recordset containing both average and individual scores submitted by other users and professional translators.

Requesting A Batch Of Translations Associated With A URL (/u)

This API enables you to request a batch of up to 500 translations associated with a URL, for example to bulk translate a web page that is being translated by users and/or professional translators. Simply call the API:

- /u/{target_language}/{url}
- or
- /u?tl={target_language}?url={url}

The service will reply with a JSON recordset that contains a list of translations matching the query.

NOTE: this service does not translate the web page, as machine translation engines do, but simply returns a list of human edited translations that are linked to that URL.