

Worldwide Lexicon

Language Service Provider API & Integration

This document explains how to integrate on demand professional translation services with the Worldwide Lexicon translation platform. The Spring 2011 release of the Worldwide Lexicon system includes a simple yet powerful API for requesting and managing professional translation and peer review. The system functions as an open network for LSPs, and enables users to build translation enabled applications and services using a single, common API that is shared across participating LSPs. The LSP API is very easy to implement, and should take an experienced developer a matter of hours to prototype and test.

The LSP API supports the following functionality:

- Request translations for text segments on demand, with optional placeholder machine translation by WWL
- Request scores for translations by professional reviewer
- Accept, delete or reject pending jobs in the translation/peer review queue
- Retrieve a list of completed jobs in the translation/peer review queue
- View a list of pending jobs in the translation/peer review queue

The API supports both polling and asynchronous callback, and thus enables developers to choose the mechanism that works best for their purposes and system capabilities.

Integrated Platform and Ecosystem

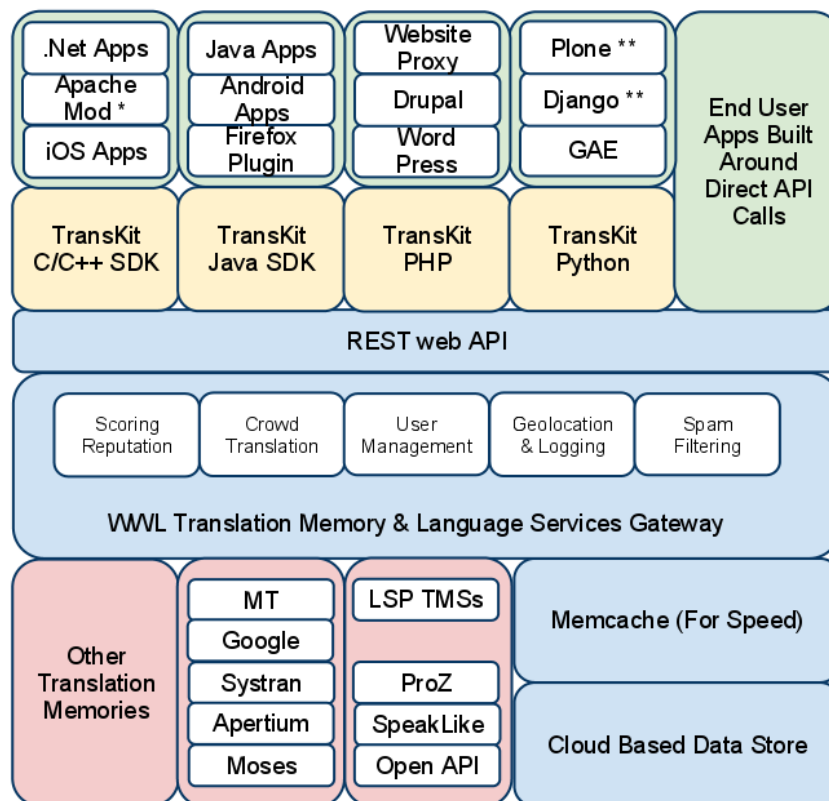
Integrating with Worldwide Lexicon makes your translation agency part of a global translation ecosystem and enables you to receive translation jobs via an automatic process. With WWL, users can request professional translations via any number of our translation modules and tools, including:

- Via TransKit, a family of libraries for popular programming languages that enable developers to embed translation in web, mobile and desktop applications
- CMS plugins, such as our Word Press translation plugin, stream requests for professional translation to WWL and participating LSPs as new content is published
- Translation proxy servers, which make any HTML compliant website translatable by humans and machines.
- Apache module (due out in Spring 2011) makes on demand translation an embedded

service in the Apache HTTP server, which powers more than 65% of websites and web applications worldwide.

Thus, with a few hours of work, you can make your translation services company part of a global translation grid that provides you with a constant stream of job requests, and also enables you to provide your customers with tools they can use to streamline their translation process.

WWL Stack Diagram



Hello World : Simplest Integration

For applications that request professional translations on demand, we provide a very simple interface for doing so. The web API at www.worldwidelexicon.org/t enables the client to request the best available translation for a text, while simultaneously initiating a request for professional translation if none is available. WWL returns the best available translation in its response, and if the user is requesting a professional translation that has not already been completed, it echoes the request to an identical API handler at the upstream LSP.

To support this interface, all an LSP needs to do is to mimic the behavior of the /t service at

Worldwide Lexicon. This is a REST API that expects the following parameters:

- guid : globally unique record locator (if omitted, you should generate one)
- sl : source language code
- tl : target language code
- st : source text (utf8)
- url : URL of parent document (optional)
- lspusername : customer username
- lspw : customer key or pw
- sla : optional service level agreement code
- queue : optional queue name
- allow_machine : allow machine translation for placeholder (y/n, default=y)
- callback_url : optional callback URL
- apikey : API key to include in callback

If the LSP has a completed translation in memory, it will reply with a JSON recordset containing the translation(s) and associated metadata. If no translation is on file or is in progress, it will return a 200 status code and empty recordset. If the request is invalid, it will return the appropriate HTTP error code and optional message. Well behaved client applications will cache the translation locally, and should only requery your system when the cache has expired (a simple way to detect post-edits and other changes in a reasonable time frame).

If the client includes a callback URL and apikey in the query, your system should submit the completed job to that URL, with the following fields in an HTTP POST form:

- guid : globally unique record locator
- sl : source language
- tl : target language
- st : source text (utf8)
- tt : translated text (utf8)
- url : URL of parent document (optional)
- lsp : LSP ID/nickname in network
- username : name or ID of translator (optional)

The request handler that answers this request will reply with OK or an HTTP error message.

Basic API Calls

An LSP that wishes to fully implement support for WWL translation services should implement the following API calls on their system. You can go to worldwidelexicon2.appspot.com to view a preview of the new system and examine the behavior of LSP related API calls there. You will need to implement the following services on your system:

- `/t` : simple REST call to request a translation and/or initiate a professional translation
- `/lsp/accept/{jobid}` : simple REST call to mark a job as accepted
- `/lsp/delete/{jobid}` : simple REST call to delete a job in the work queue
- `/lsp/fetch` : a simple REST call to fetch a recordset of completed jobs for inspection
- `/lsp/queue` : a simple REST call to view the current contents of the translation work queue
- `/lsp/quote` : a simple REST call to request a quote for a translation based on language pair, word count and service level agreement
- `/lsp/reject/{jobid}` : a simple REST call to reject a completed job and trigger a redo
- `/lsp/score` : a simple REST call to request a translator to score a translation

While this is a fairly simple API, it provides the basic functions needed to build a sophisticated and highly automated translation application, while also providing developers with a fast track to prototype and test using common software development tools. The API is based on the REST concept, and uses JSON as the data interchange format for structured recordsets (this is directly supported in Javascript, which makes it easy to build interactive “web 2.0” applications that interact with the API, as well as by common server side languages).

The API does not mandate how you store or process data internally, or how you manage the translation process. As long as you mimic the behavior and output generated by the dummy `/lsp/*` calls on the WWL server, everything will work fine.

Example Applications and Use Cases

On demand translation enables users to dramatically improve the efficiency and response time in their translation work flows. It is especially well suited to dynamic content, customer contact and communication services, and similar scenarios. The API eliminates the need to ship files back and forth, and makes the entire process of requesting and receive translations, scores, etc fully automated and dynamic.

Software Translation and Localization

Software and web applications can use our TransKit SDKs to request translations from the system, and its participating language service providers. This on demand approach eliminates the need to ship localization files (e.g. PO files), and enables applications, as well as the content they generate or display to be deeply integrated with translation services. A common example is our PHP library, which makes building multilingual applications in the PHP web scripting languages a trivial process.

On Demand Content Translation For CMS Platforms

We have developed translation filters for popular content management systems, including Word

Press. These tools, rather than translating pages as static content, translate pages as they are generated. Because the content published on the systems is often dynamic in nature, this approach enables the system to detect incremental changes, and request translations only for the parts of a document that have changed (a recent blog post for example). Our Word Press translator, which is fully integrated to provide machine, crowd and professional translation is an especially good example of the utility of this approach.

Customer Contact and Communication

On demand translation is a powerful tool for enabling multilingual customer communication and support services. For example, a CRM system can be integrated with WWL to provide near real-time translation for inbound and outbound customer communication, and enables a company to support customers in any number of languages. Furthermore companies can build solutions that blend machine, crowd and professional translation to optimize costs and quality (for example, a CRM system might use machine translation for a first pass, and then enable agents to request a professional translation as needed when the MT translation falls short).