



“ServiceAPI” to the WorldLingo System

Technical Summary

WorldLingo



Table of Contents

Table of Contents.....	2
Table of Figures.....	2
List of Tables	2
1. Purpose	3
2. Intended Audience and Reading Suggestions	3
3. References.....	3
4. API Description.....	3
5. Mime Types.....	6
6. Supported Languages	7
7. Autodetect.....	8
7.1 Usage	8
7.2 Return Values.....	8
7.3 Limitations and Best Practices	8
8. Supported Encodings	9
9. Improving Translation Quality.....	11
10. Access via SOAP	12
11. Testing and Security	13
12. Examples	13
13. Return codes	15
14. Revision History.....	16

Table of Figures

Figure 1: Schematic Calling Sequence	4
--	---

List of Tables

Table 1: List of Mandatory Parameters	5
Table 2: List of Optional Parameters	5
Table 3: Used Language Codes.....	7
Table 4: Supported Encodings	11
Table 5: Glossary Codes	11
Table 6: Return codes	15



1. Purpose

This document describes the *ServiceAPI*, which allows connection to the WORLDLINGO.COM SYSTEM and submitting requests, which are then processed. When finished the result can be retrieved by the client and used as required.

2. Intended Audience and Reading Suggestions

The content of this document is aimed at technically skilled persons, such as webmasters, network administrators or software developers. It requires certain knowledge of Internet techniques and protocols such as HTTP. It is recommended that you follow the references in section 3 and read the material provided in the listed locations. Since the way translated text is returned to the calling party is HTTP based, the calling system needs to implement scripts or programs that parse these results and effectively convert them into the desired end result, such as a web page. Hence, the user of the API needs detailed knowledge in these techniques.

3. References

The following Internet Specifications provide relevant information to the development and implementation of the ServiceAPI:

- ISO 639 Language Codes
- ISO 3166 Country Codes
- ISO 4217 Currency Codes
- RFC 1738 Uniform Resource Locators (URL)
- RFC 1808 Relative Uniform Resource Locators
- RFC 1945 Hypertext Transfer Protocol (HTTP/1.0)
- RFC 2046 MIME Part Two: Media Types

On-line versions of any of these RFC's (ie. "Request For Comments") can be located at <http://www.rfc-editor.org>. The World Wide Web Consortium (<http://www.w3.org>) is the definitive source of HTTP related information that affects this specification and its implementations.

4. API Description

The ServiceAPI is based on HTTP (Hypertext Transfer Protocol). This allows different programming environments to be used to access the API. One might use a PERL script, Java classes or C based CGI applications. Even simple Telnet connections to the server are possible, although it is the purpose of this API to integrate the services provided by WorldLingo into other applications.

Please refer to Figure 1 for a graphical representation of the communication involved in this API. The process works as follows:

1. The client sends a request to the WorldLingo server. This request must be a valid HTTP request in the form of

```
http://www.worldlingo.com/S000.1/api?parameterlist...
```

that might result in the following data being sent to the server

```
GET /S000.1/api?parameterlist... HTTP/1.1
Accept: */*
Accept-Language: en,de;q=0.7,ja;q=0.3
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT)
Host: www.worldlingo.com
Connection: Keep-Alive
```

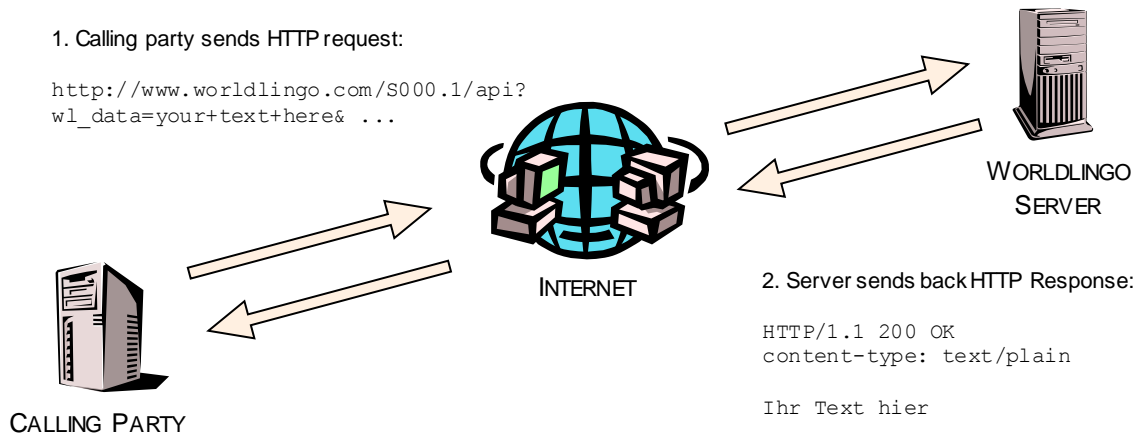
2. The server returns the translated text to the client. In this example it might look like the following

```
HTTP/1.1 200 OK
Date: Fri, 31 Mar 2000 03:21:18 GMT
Server: Apache
Content-Type: text/plain
```

translated text...

The result can then be processed on the calling side. It can be embedded in a resulting web page or sent off by e-mail.

Figure 1: Schematic Calling Sequence





The following table gives a comprehensive list of all mandatory request parameters, please be advised that you must url-encode the values of the parameters:

Parameter	Description
wl_data	The data to be translated. The format of the data is specified by the wl_mimetype parameter and defaults to plain text.
wl_password	The password of the calling party, issued by WorldLingo upon service creation.
wl_srclang	The selected source language for translation.
wl_trglang	The selected target language for translation.

Table 1: List of Mandatory Parameters

The following table gives a comprehensive list of all optional request parameters:

Parameter	Description
wl_mimetype	text/plain – The data is in plain text format (default) text/html – The data is in HTML format
wl_dictno	Specifies the custom dictionary to use. Refer to section 8 for further information.
wl_gloss	Use a context specific glossary. Refer to section 8 for further information.
wl_srcenc	Specify the input encoding of the data (see "wl_data" above). If the parameter is not given then the server assumes the data is encoded in the default encoding of "UTF-8"
wl_trgenc	Specify the output encoding for the translation. If the parameter is not given then the server returns the result encoded in the default encoding of "UTF-8"
wl_errorstyle	0 – The API will return a simple number, as the first line of the result, referencing the particular error. Refer to section 11 for error information. 1 – The error code will be placed in the http response header field "X-WL-ERRORCODE". Refer to section 11 for error information.

Table 2: List of Optional Parameters

The parameters that specify the encodings (wl_srcenc, wl_trgenc) have to be provided in the same way as done inside a HTML page within the "Content-Type" and its "charset" part. An example might be the following webpage encoded in a Japanese Shift JIS:

```
<html>
<head>
<title>Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=x-sjis">
</head>
<body bgcolor="#FFFFFF">
Japanese Text here...
</body>
</html>
```

The text of this web page that is sent to the WorldLingo Server must provide the following parameter as part of the request to get the correct results

```
...?wl_data=Japanese+text+here&wl_srclang=ja&wl_trglang=en&wl_trgenc=shift_
jis
```



5. Mime Types

The WorldLingo ServiceAPI currently handles two types of input data. The ServiceAPI is primarily designed to handle the translation of text. However, functionality for html translation has also been built into the system. The following example illustrates how the API can be used to translate HTML.

Request:

```
http://www.worldlingo.com/S000.1/api?wl_srclang=en&wl_trglang=de&wl_password=secret&wl_mimetype=text%2Fhtml&wl_data=%3Chtml%3E%0D%0A%3Chead%3E%0D%0A%3Ctitle%3Ethis+is+a+test%3C%2Ftitle%3E%0D%0A%3C%2Fhead%3E%0D%0A%3Cbody%3E%0D%0Athis+is+a+test+of+the+HTML+translation+via+the+API%0D%0A%3C%2Fbody%3E%0D%0A%3C%2Fhtml%3E%0D%0A
```

Response:

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html;charset=UTF-8
Content-Length: 137
```

```
0
<html>
<head><title>dieses ist ein Test</title></head>
<body>Dieses ist ein Test der HTML-Übersetzung über die API</body>
</html>
```



6. Supported Languages

Language Code	Language
af	Afrikaans
sq	Albanian
ar	Arabic
az	Azerbaijani
eu	Basque
be	Belarusian
bn	Bengali
bg	Bulgarian
ca	Catalan
zh_CN	Chinese (Simplified)
zh_TW	Chinese (Traditional)
hr	Croatian
cs	Czech
da	Danish
nl	Dutch
en	English
eo	Esperanto
et	Estonian
fa	Farsi
tl	Filipino
fi	Finnish
fr	French
gl	Galician
ka	Georgian
de	German
el	Greek
gu	Gujarati
ht	Haitian Creole
ha	Hausa
he	Hebrew
hi	Hindi
hu	Hungarian
is	Icelandic

Language Code	Language
id	Indonesian
ga	Irish
it	Italian
ja	Japanese
kn	Kannada
ko	Korean
la	Latin
lv	Latvian
lt	Lithuanian
mk	Macedonian
ms	Malay
mt	Maltese
no	Norwegian
pl	Polish
pt	Portuguese
ro	Romanian
ru	Russian
sr	Serbian
sk	Slovak
sl	Slovenian
so	Somali
es	Spanish
sw	Swahili
sv	Swedish
ta	Tamil
te	Telugu
th	Thai
tr	Turkish
uk	Ukrainian
ur	Urdu
vi	Vietnamese
cy	Welsh
yi	Yiddish

Table 3: Used Language Codes

We support multilingual language pairs between all of the above languages.



7. Autodetect

Users can submit text without knowing the source language. The autodetect feature will analyze the characters and ngrams in the submitted text and return scores for the highest probability matches. The highest probability is used as the detected source language.

Translation of the data is then performed using the detected source language.

7.1 Usage

The autodetect feature is primarily meant to be used with WorldLingo's Translation API. To use the autodetect feature, use “auto” as the value of `wl_srclang`.

7.2 Return Values

The translation response will be performed using the detected source language. The detected language code can optionally be inserted into the response header “X-WL-SRCLANG”. Contact WorldLingo if you would like the language code returned in the response header.

7.3 Limitations and Best Practices

Language autodetection works best with text larger than 10-15 words. Smaller data sets may return false language guesses.

8. Supported Encodings

The parameter "wl_srcenc" or "wl_trgenc" determines what the encoding (input or output) of the given text is. If the parameter is not given as part of the request it is assumed by using a default value of UTF-8.

Language Code	Default	Additional
af	UTF-8	
ar	UTF-8	ISO-8859-6
az	UTF-8	
be	UTF-8	
bg	UTF-8	
bn	UTF-8	
ca	UTF-8	
cs	UTF-8	
cy	UTF-8	
da	UTF-8	
de	UTF-8	ISO-8859-1
el	UTF-8	ISO-8859-7
en	UTF-8	ISO-8859-1
eo	UTF-8	
es	UTF-8	ISO-8859-1
et	UTF-8	
eu	UTF-8	
fa	UTF-8	
fi	UTF-8	
fr	UTF-8	ISO-8859-1
ga	UTF-8	
gl	UTF-8	
gu	UTF-8	
ha	UTF-8	
he	UTF-8	
hi	UTF-8	
hr	UTF-8	
ht	UTF-8	
hu	UTF-8	
id	UTF-8	
is	UTF-8	



it	UTF-8	ISO-8859-1
iw	UTF-8	
ja	UTF-8	Shift_JIS
ka	UTF-8	
kn,	UTF-8	
ko	UTF-8	EUC-KR
la	UTF-8	
lt	UTF-8	
lv	UTF-8	
mk	UTF-8	
ms	UTF-8	
mt	UTF-8	
nl	UTF-8	ISO-8859-1
no	UTF-8	
pl	UTF-8	
ps	UTF-8	
pt	UTF-8	ISO-8859-1
ro	UTF-8	
ru	UTF-8	KOI8-R, Cp1251
sk	UTF-8	
sl	UTF-8	
so	UTF-8	
sq	UTF-8	
sr	UTF-8	
sv	UTF-8	ISO-8859-1
sw	UTF-8	
ta	UTF-8	
te	UTF-8	
th	UTF-8	
tl	UTF-8	
tr	UTF-8	
uk	UTF-8	
ur	UTF-8	
vi	UTF-8	
yi	UTF-8	



zh_CN	UTF-8	GB2312
zh_TW	UTF-8	Big5

Table 4: Supported Encodings

9. Improving Translation Quality

To improve the quality of your translations you can use a custom dictionary of words and phrases either not to be translated or to be translated to a specific word or phrase. To create a custom dictionary, go to <http://www.worldlingo.com/login/dictionaries.html>. The dictionary number assigned is the value you need to pass to the API under the parameter 'wl_dictno'.

Another way to improve the translation quality is to specify the subject matter of the text by using one of our pre-defined glossaries.

Glossary Subject	Code
General	gl1
Automotive	gl2
Aviation/Space	gl3
Chemistry	gl4
Colloquial	gl5
Computers/IT	gl6
Earth Sciences	gl7
Economics/Business	gl8
Electronics	gl9
Food Science	gl10
Legal	gl11
Life Sciences	gl12
Mathematics	gl13
Mechanical Engineering	gl14
Medicine	gl15
Metallurgy	gl16
Military Science	gl17
Naval/Maritime	gl18
Photography/Optics	gl19
Physics/Atomic Energy	gl20
Political Science	gl21

Table 5: Glossary Codes



10. Access via SOAP

The ServiceAPI is also accessible via a SOAP interface; the WSDL for the SOAP interface is available at <http://www.worldlingo.com/soap/ServiceApi.wsdl>. SOAP can be used from within a number of languages including Visual Basic, C# & .NET

Example - Access from within a C# application

To access the SOAP interface from within a C# application you simply need to open your Visual Studio (.NET) project, go to the Project menu, and select Add Web Reference. In the URL field enter <http://www.worldlingo.com/soap/ServiceApi.wsdl>. It will immediately check the WSDL and display that it found 1 Service (ServiceApi) and list the method(s) the service has available. In the Web reference name field you need to enter in the name the object will be called in the actual code eg. WorldLingo. You can now click the Add Reference button.

To use the object in the application go to the class source code where you wish to access it from and create a new object of the type 'WorldLingo' (or the name you entered in as the 'Web reference name').

```
WorldLingo.ServiceApiService service = new WorldLingo.ServiceApiService();
```

Now to send a translation request you simply call the 'translate' method on this object. The translate method expects the following parameters:

serviceNo - your service number

password - your service password.

data - the data you wish to be translated.

mimeType - the mimetype of the data to be translated (text/plain or text/html).

srcLang - the source language.

trgLang - the target language.

srcEnc - the encoding the source data is encoded in. (Can be null, defaults to UTF8).

trgEnc - the encoding translation should return. (Can be null, defaults to UTF8).

dictionary - the number of the dictionary to be used. (Can be null).

glossary - the glossary to be used (Can be null).

```
WorldLingo.TranslateResult result = service.translate("S000.1", "secret",  
"text to be translated.", "text/plain", "en", "fr", "utf8", "utf8", null,  
null);
```

The result is returned as a 'TranslateResult' object that allows you easy access to the translation result. It also provides you with an error code, the result encoding and the mimetype.

If you find the result data you are getting is not readable you might need to change the encoding:

```
Encoding encoding = System.Text.Encoding.UTF8;  
new String(encoding.GetChars(encoding.GetBytes(result.data)));
```



11. Testing and Security

To allow a user of the API to test the services and evaluate its functionality there is a test mode built into it. The Service Number S000.1 (as used in the examples) in combination with the password “secret” will translate the given text from the given source language into a random target language. The target languages are always Latin based (ISO-8859-1 encoded), therefore excluding the Asian encodings. This ensures that most users of the API can handle the random result. Additionally the test account is limited 25 words per translation request.

This test mode delivers the same result as in a normal use of the API with an established service. But because the result is not predictable it is not useful in a normal production environment.

When you have been given access to the API, you will need to use your own service number (built into the url) and password. You are able to change the password for the service at any time. As an extra security feature, our API is accessible via https as well as http.

12. Examples

A number of examples on how the API can be used have been prepared; they include the following languages Java, PHP, Perl and HTML and are available at <http://www.worldlingo.com/services/service1?serviceaction=download>

The following examples give an overview of how to use the service. Please note, for ease of reading, this document only illustrates GET requests. You are also able to use POST requests.

1. Example

This example displays a valid request to the server, providing the text, language pair and authorization information.

Request:

```
http://www.worldlingo.com/S000.1/api?wl_data=This+is+a+test+translation&wl_srclang=en&wl_trglang=de&wl_password=secret
```

Response:

```
HTTP/1.1 200
Server: Apache
Content-Type: text/plain;charset=UTF-8
Content-Length: 34
```

0

```
Dieses ist eine Testübersetzung
```

2. Example

Here the user provided the wrong password for authorization.

Request:



`http://www.worldlingo.com/S000.1/api?wl_data=This+is+a+test+translation&wl_srclang=en&wl_trglang=de&wl_password=wrong`

Response:

```
HTTP/1.1 200
Server: Apache
Content-Type: text/plain; charset=UTF-8
Content-Length: 29
```

```
26
  This is a test translation
```

3. Example

Here is a valid request, this time using the request header error style (`wl_errorstyle=1`).

Request:

`http://www.worldlingo.com/S000.1/api?wl_data=This+is+a+test+translation&wl_srclang=en&wl_trglang=de&wl_password=secret&wl_errorstyle=1`

Response:

```
HTTP/1.1 200
Server: Apache
X-WL-ERRORCODE: 0
Content-Type: text/plain; charset=UTF-8
Content-Length: 32
```

Dieses ist eine Testübersetzung



13. Return codes

The following outline describes the possible return codes given by the API:

Status	Return code
Successful	0
Incorrect Password	26
Source language not subscribed to.	28
Target language not subscribed to.	29
Invalid Language Pair	176
No input data	177
Invalid Mime-type	502
Translation timed out	1176
TEngineErrorException. Occurs when single words or short phrases in isolation cause the translation engine to fail.	1181

Table 6: Return codes

Each translation provided via the API is accompanied by a Return Code, which you can choose to see in the response header or within the response body. In use, the return codes are referred to as “error codes”, though obviously they do not necessarily indicate that an error has occurred. Return/error codes are available to provide insight as to the status of the completed translation request. For example, Error Code 0 means that a translation was successful, while Error Code 1176 means that a translation timed-out, however, an 1176 doesn’t necessarily mean that no translations were returned and during initial testing it’s a good idea to familiarize yourself with what’s returned so you can decide if results other than Zero can be accepted as usable within your application.

If you find you are receiving an error code that is not listed above, or for more information about error codes in general please contact WorldLingo Technical Support <http://www.worldlingo.com/contact/>

To subscribe to WorldLingo’s ServiceAPI solution, please visit:
http://www.worldlingo.com/products/worldlingo_api.html



or contact us <http://www.worldlingo.com/contact/>

14. Revision History

Rev	Effective Date	Author	Controller	Authorizer	Description of Change
01	5/30/2014	Tom Amano	Tatiana Ferreira	Martin Pratt	Initial release of the document.
02	12/15/2014	Alejandro Castillo	Alejandro Castillo	Martin Pratt	Update Supported Languages, Encodings and add new section for Autodetect.
03	12/2/2015	Alejandro Castillo	Alejandro Castillo	Martin Pratt	Simplified language codes to improve clarity.