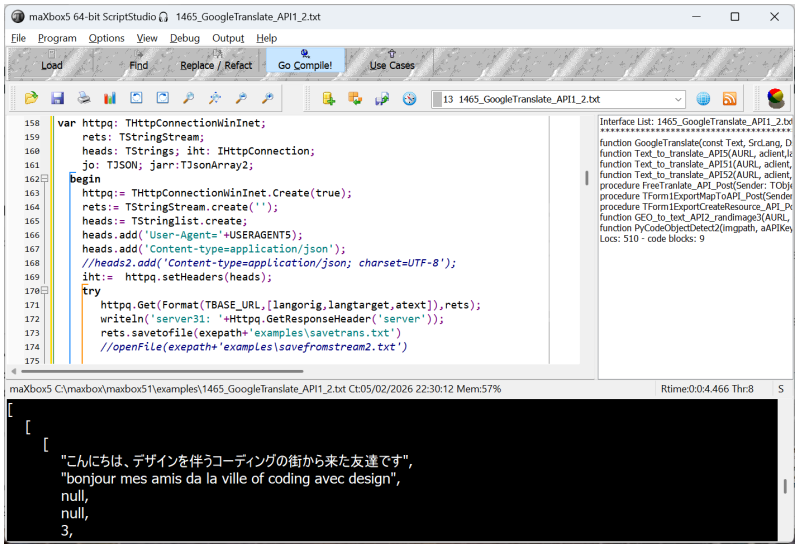
Free Google Translator API

February 6, 2026

While learning katakana through the source code of Google's Google Dictionary Chrome extension, which has support for translating via Google Translate, I found the web frontend endpoint they use in order to do just that.

We don't need an API-Key or an registration account and can call and use almost any Rest-Client as get or post:

URL: `https://clients5.google.com/translate_a/t?client=dict-chrome-ex&sl=auto&tl=en&q=bonjour`



https://sourceforge.net/projects/maxbox5/files/examples/1465_GoogleTranslate_API1_3.txt/download

`TStringStream` is tied to a *specific encoding* that has to be declared in the *constructor* before the stream is populated with data, and it cannot be changed dynamically. The default encoding is `TEncoding.Default`, which represents the OS default encoding. If the HTTP response uses a *different* encoding, the data will not decode to a `String` correctly.

Second, we have to make sure to seeking the stream's `Position` back to 0 before calling `ReadString()`. An easier way to retrieve a `TStringStream`'s

content as a decoded `String` is to use the `DataString` property instead, which ignores the `Position` property and returns the entire stream content as a whole or in my case to use the file-system store with SavetoFile and loadfromFile:

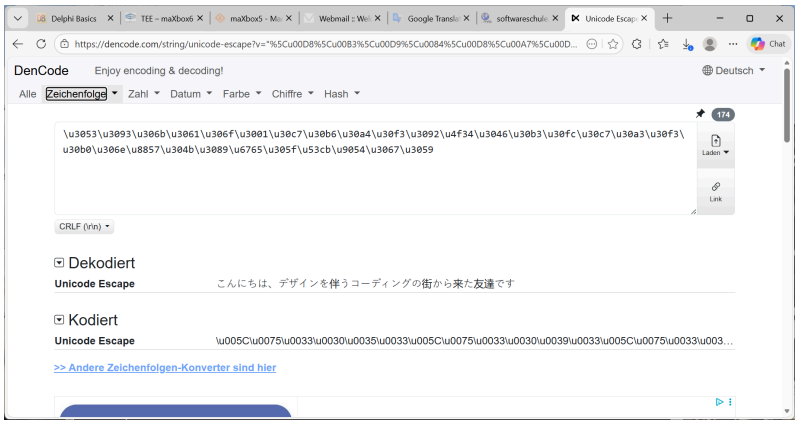
```
function Text_to_translate_API52(AURL,aclient,langorig,langta
var httpq: THttpConnectionWinInet;
    rets: TStringStream; heads: TStrings; iht: IHttpConnectio
    jo: TJSON; jarr:TJsonArray2;
  begin
    httpq:= THttpConnectionWinInet.Create(true);
    rets:= TStringStream.create('');
    heads:= TStringlist.create;
    heads.add('User-Agent='+USERAGENT5);
    heads.add('Content-type=application/json');
    iht:=  httpq.setHeaders(heads);
    try
       httpq.Get(Format(TBASE_URL,[langorig,langtarget,atext]
       writeln('server31: '+Httpq.GetResponseHeader('server')
       rets.savetofile(exepath+'examples\savetrans.txt')
       //openFile(exepath+'examples\savefromstream2.txt')

       jo:= TJSON.Create();
       jo.parse(loadfileUC(exepath+'examples\savetrans.txt'))
       jarr:= jo.JsonArray;
       if httpq.getresponsecode=200 Then
          result:=jarr[0].decode(loadfileUC(exepath+'examples
       else result:='Rest API Failed:'+
                    itoa(Httpq.getresponsecode)+Httpq.GetRespons
    except
       writeln('EWI_HTTP: '+ExceptiontoString(exceptiontype,e
    finally
       //httpq.free;
       httpq:= Nil;
       rets.Free;
       jo.free;
       heads.free;
    end;
  end;
```

I am sending an HTTP Get request to Google's Translator endpoint API, and I fill my StringStream with the response from the frontend-endpoint:

```
httpq:= THttpConnectionWinInet.Create(true);
    rets:= TStringStream.create('');
    heads:= TStringlist.create;
    heads.add('User-Agent='+USERAGENT5);
    heads.add('Content-type=application/json');
    iht:=  httpq.setHeaders(heads);
    try
       httpq.Get(Format(TBASE_URL,[langorig,langtarget,atext]
       writeln('server31: '+Httpq.GetResponseHeader('server')
       rets.savetofile(exepath+'examples\savetrans.txt')
```

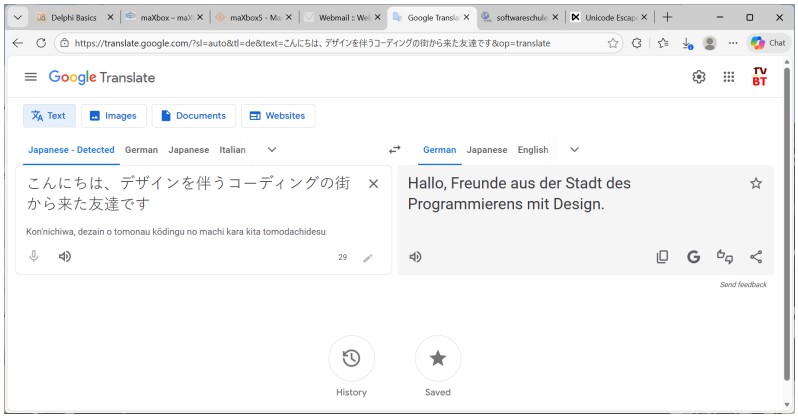
Special attention has to be made with Unicode and for example the Japanese Language, thus, the number of characters in Japanese is many more than 256 and thus cannot be encoded using a single byte – Japanese is thus encoded using two or more bytes, in a so-called "double byte" or "multi-byte" encoding. Problems that arise relate to transliteration and romanization, character encoding, and input of Japanese text.



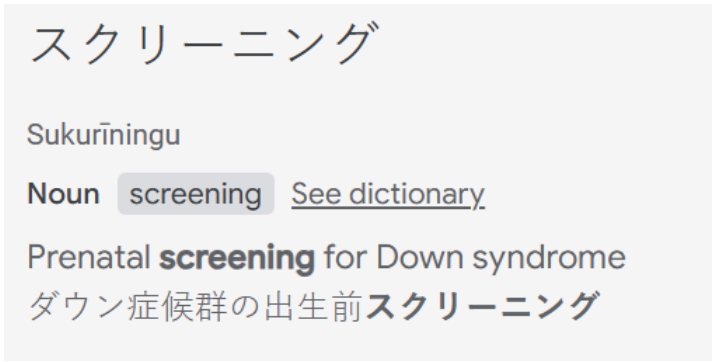
Unicode Escape Sequences in JSON Return

There are several standard methods to encode Japanese characters for use on a computer, including JIS, Shift-JIS, EUC, and Unicode. While mapping the set of kana is a simple matter, kanji has proven more difficult.

In the end we got an almost same result like the well known translator:



Google Translate



It depends on the language to call more functions lika a dictionary

Parameters and Errors

You do have several parameters to call the API:

Breakdown

Endpoint: `https://clients5.google.com/translate_a/t`

Query Parameters

Query Parameter	Default	Notes
client	`dict-chrome-ex`	Needs to be `dict-chrome-ex` or else you'll get a 403 error.
sl	`auto`	Designates the source language of the text to translate.
tl	(none)	Designates the destination language of the text to translate.
q	(none)	Text to translate

Written Japanese uses several different scripts: kanji (Chinese characters), 2 sets of *kana* (phonetic syllabaries) and roman letters. While kana and roman letters can be typed directly into a computer, entering kanji is a more complicated process as there are far more kanji than there are keys on most keyboards.

We call the function like the following:

```
const
  TBASE_URLS = 'https://clients5.google.com/translate_a/t?clie

atext:= 'Programmieren mit Design';
writeln((Text_to_translate_API52(TBASE_URLS,'dict-chrome-ex','
```

In the end you can trap mostly in two types of error:

1 (502) 875 err:20

EWI_HTTP: Exception: Can not send REST message https://translate.googleapis.com/translate_a/single?client=gtx&dt=t&sl=auto&tl=ja&q=bonjour mes amis da la ville of coding avec design, error: Bad Gateway (502)

2 debug: 222- 875 err:21
debug: 207-The requested header could not be located. 878 err:20

REQC_Err: Exception: The requested header could not be located.

First (Gateway) means you don't have a valid JSON [] or stream object () to send and the second (header) could be an invalid request header:

```
USERAGENT5 = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleW
                          ' Chrome/126.0.0.0 Safari/537.36 E
heads:= TStringlist.create;
    heads.add('User-Agent='+USERAGENT5);
    heads.add('Content-type=application/json');
    iht:=  httpq.setHeaders(heads);
```

It seems like a great discovery cause the API is simple to use and powerful in result with support of many languages and also the provided JSON has an automatic (parameter auto) language detection inside:

```
server31:  ESF
[
    [
        [
            "デザインを取り入れたプログラミング",
            "Programmieren mit Design",
```

```
            null,
            null,
            3,
            null,
            null,
            [
                [
                ],
                [
                ]
            ],
            [
                [
                    [
                        "ad3d68f0fc47c8c7a84213b3d785dc3a",
                        "de_en_2023q1.md"
                    ]
                ],
                [
                    [
                        "466914b2b9b759682681a550c00b67dd",
                        "en_ja_2023q1.md"
                    ]
                ]
            ]
        ]
    ],
    null,
    "de",
    null,
    null,
    null,
    1,
    [
    ],
    [
        [
            "de"
        ],
        null,
        [
            1
        ],
        [
            "de"
        ]
    ]
]
 mX5🐞 executed: 06/02/2026 16:06:12  Runtime: 0:0:2.710
Memload: 71% use
```


Conclusion

The Google Translate API is a powerful tool for developers and businesses looki ng to integrate translation capabilities into their applications. It offers two main versions: Basic (V2) and Advanced (V3), with the latter providing features like gl

ossaries for consistent translations and AutoML for custom translation models. Key features include:

- Glossaries: User-defined dictionaries for specific terms.
- Batch Translation: Efficient processing of multiple translations at once.
- Document Translation: Directly translating documents while preserving formatting.
  For a successful implementation, it's crucial to use IAM accounts for security and to plan for cost control and compliance with regulations like DSGVO. 2
- 2 Quellen
- Also a free API endpoint with no managment and support available.

Speak the Locomotion Languages

Posted in Code, Data Science, Japan, Languages, Machinelearning, maXbox, Tools

Leave a comment

Write a comment...

maxkleiner - Logged in via WordPress.com - Log out

Notify me of new posts
Receive web and mobile notifications for posts on this site.

Email me new posts

Instantly Daily Weekly

Email me new comments