

////////////////////////////////////

Google Translate

maXbox Starter 109 - Found a Google Translate endpoint that doesn't require an API key.

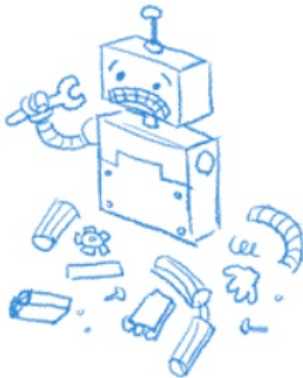
"Lost in translation - post to application".

While digging or diving through the source code of Google's Google Dictionary Chrome extension, which has support for translating via Google Translate, I found the endpoint they use in order to do just that. Since google translator frequently runs into 5xx errors, it might be useful to switch off to another endpoint, although this one is also annoyingly touchy with 403s error or 404s.



400. That's an error.

The server cannot process the request because it is malformed. It should not be retried. That's all we know.



pic:1202_transerror.png

So the endpoint we do invoke is:

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

Its a backbone from the link

https://translate.googleapis.com/translate_a/single

Now let's have a look at the app/script below with individual texts from your own data to translate. For this, we wrote two useful functions. The first one returns the text translated with the target language. The second one accepts one sentence as an argument with language detection as a param "auto". Then it will show the text in JSON or as file:

Query Parameters

Query Parameter	Default	Notes
client	dict-chrome-ex	Needs to be dict-chrome-ex or else you'll get a 403 error.

Query Parameter	Default	Notes
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

This seems like a great discovery!

This app allows you to translate or detect text from many different languages. That's why I want this endpoint to be seamlessly integrated into googletrans, with it switching between endpoints if one is facing 4xx/5xx errors.

```

Const AURLS = 'https://clients5.google.com/translate_a/t?client=dict-
chrome-ex&sl=%s&tl=%s&q=%s';

function Text_to_traslate_API2(AURL, aclient, langorig, langtarget, atext:
                                string):string;
var httpq: THttpConnectionWinInet;
    rets: TStringStream;
    heads: TStrings; iht:IHttpConnection2;
    jo: TJSON; jarr:TJsonArray2;
begin
    httpq:= THttpConnectionWinInet.Create(true);
    rets:= TStringStream.create('');
    try
        httpq.Get(Format(AURLS, [langorig, langtarget, atext]), rets);
        writeln('server: '+Httpq.GetResponseHeader('server'));

        jo:= TJSON.Create();
        jo.parse(rets.datastring)
        jarr:= jo.JsonArray;
        if httpq.getresponsecode=200 Then result:=jarr[0].stringify
            else result:='Failed: '+
                itoa(Httpq.getresponsecode)+Httpq.GetResponseHeader('message');
    except
        writeln('EWI_HTTP: '+ExceptionToString(exceptiontype, exceptionparam));
    finally
        httpq.free;
        httpq:= Nil;
        rets.Free;
        jo.free;
    end;
end;

```

Google's service, offered free of charge, instantly translates words, phrases, text and web pages between English and over 100 other languages. That's how we call the function:

```

atext:= 'bonjour mes amis da la ville';
writeln(utf8ToAnsi(Text_to_traslate_API2(AURL, 'dict-chrome-
                                ex', 'auto', 'es', atext)));

```

and the result:

```

server: ESF
["Hola mis amigos en la ciudad", "fr"]

```


for the query: https://translate.googleapis.com/translate_a/single?client=gtx&dt=t&sl=en&tl=ja&q=Hello, how are you today?

And something like this:

```
[["Bonjour", "Hello", null, null, 1]
]
, null, "en", null, null, null, null, [
]
```

String unicode (\uxxx) encoding and decoding.

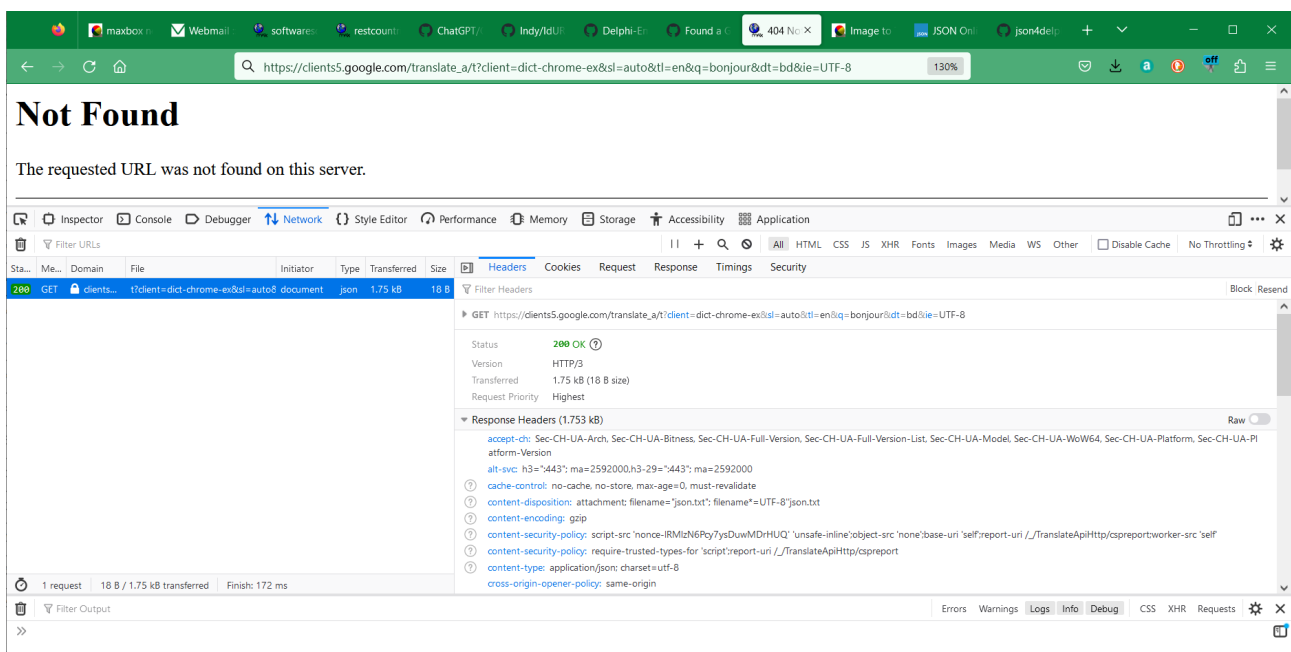
After some testing with request headers and F12 tools - Inspect (see below), I found the solution for the garbled text it can be. Simply set the User-Agent header to the one that Google Chrome uses.

Example:

```
import requests
word = 'لماذا تفعل هذا'
url = "https://clients5.google.com/translate_a/t?client=dict-chrome-ex&sl=auto&tl=en&q=" + word
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36(KHTML, like Gecko) Chrome/88.0.4324.104 Safari/537.36'}

try:
    request_result = requests.get(url, headers=headers).json()

    print(request_result)
    print('[In English]: ' + request_result['alternative_translations'][0]
['alternative'][0]['word_postproc'])
    print('[Language Dected]: ' + request_result['src'])
except:
    pass
```



Pic:1202_inspect_box_tutor109.png

Conclusion:

We should probably create a way to house all of these endpoints through one API, perhaps an interface that all of the endpoints implement. Also maybe try to URL encode the text before sending it (and use GET as it should not work with POST).

"Neural machine translation (NMT) systems have reached state of the art performance in translating text and are in wide deployment. Yet little is understood about how these systems function or how they break. A research paper suggests improving GPT-4 performance by 30% by having it consider why it was wrong.

The main part function opens connection with `HttpGet(EncodURL, mapStrm);`, invokes the API and results a stream which we convert to a datastring. A RESTful API needs to have one and exactly one entry point. If you don't use User-Agent, the response will have an incorrect encoding (ASCII). If you use a banned User-Agent like `curl/7.37.1`, you'll get a 403 error page; but if you use a web browser User-Agent, the response will have a correct encoding (UTF-8). About the POST request, it seems that the API endpoint allows using POST requests using query parameters instead of a body. This also works with PUT, PATCH, DELETE and OPTIONS (basically all common methods).

```
atext:= urlencode('A research paper suggests improving GPT-4 performance by 30%
by having it consider why it was wrong.');
```

```
writeln(utf8ToAnsi(Text_to_traslate_API2(AURL,'dict-chrome-
ex','auto','it',atext)));
```

```
["Un documento di ricerca suggerisce di migliorare le prestazioni del
GPT-4 del 30% facendo in modo che fosse sbagliato.", "en"]
```

Reference:

JSON Lib:

<https://github.com/rilyu/json4delphi/blob/master/src/Jsons.pas>

URI Builder:

<https://github.com/ghquant/Delphi-EmbeddedWB/blob/master/Source/EwbUrl.pas>

<https://github.com/skelter/Indy/blob/master/Lib/Protocols/IdURI.pas>

The script:

<http://www.softwareschule.ch/examples/gtranslate.txt>

Doc and Tool: <https://maxbox4.wordpress.com>

Script Ref: 1202_Google_Translate_API2.txt

Max Kleiner 30/03/2023