

```

1: *****
2:   maXbox Starter 95
3: *****
4:
5: Work with a Language Translator
6: -----
7: Max Kleiner
8:
9: //Zwei Worte werden Dir im Leben viele Türen öffnen - "ziehen" und "stossen".
10: Two words will open many doors for you in life - "pull" and "push".
11:
12: https://github.com/maxkleiner/argos-translate
13: https://github.com/LibreTranslate/LibreTranslate#mirrors
14:
15: Free and Open Source Machine Translation API, entirely self-hosted. Unlike other APIs, it
    doesnt rely on proprietary providers such as Google or Azure to perform translations. Instead,
    its translation engine is powered by the open source Argos Translate library. You can use the
    LibreTranslate API using many language bindings.
16:
17: Argos Translate uses OpenNMT for translations, SentencePiece for tokenization, Stanza for
    sentence boundary detection, and PyQt for GUI. Argos Translate can be used as either a Python
    library, command-line, or GUI application. LibreTranslate is an API and web-app built on top
    of Argos Translate.
18:
19: This is what we want to use in maXbox:
20:
21: const res = await fetch("https://libretranslate.com/translate", {
22:   method: "POST",
23:   body: JSON.stringify({
24:     q: "Hello!",
25:     source: "en",
26:     target: "es"
27:   }),
28:   headers: { "Content-Type": "application/json" }
29: });
30: console.log(await res.json());
31:
32: JSON.stringify() converts a value to JSON notation representing it: If the value has a
    toJSON() method, its responsible to define what data will be serialized.; Boolean, Number, and
    String objects are converted to the corresponding primitive values during stringification, in
    accord with the traditional conversion semantics.
33:
34: Then we use our Com-Object from OleVariant at late binding:
35:
36: function getPostTranslateLibre3(feedstream, fromlang, tolang: string): string;
37: var
38:   Url, aAPI_KEY, source: string;
39:   jo, locate: TJSONObject;
40:   httpReq, hr: Olevariant;
41:   strm: TStringStream;
42: begin
43:   httpReq:= CreateOleObject('WinHttp.WinHttpRequest.5.1');
44:   // Open the HTTPS req. connection.
45:   try
46:     hr:= httpReq.Open('POST', 'https://libretranslate.pussthecat.org/translate', false);
47:     httpReq.setRequestHeader('user-agent', CUSERAGENT );
48:     httpReq.setRequestHeader('content-type', 'application/x-www-form-urlencoded');
49:     if hr = S_OK then HttpReq.Send('q='+HTTPEncode(feedstream)+
50:                                   '&source='+fromlang+'&target='+tolang);
51:     If HttpReq.Status = 200 Then
52:       result:= HttpReq.responseText
53:     Else result:= 'Failed at getting response:'+itoa(HttpReq.Status)+HttpReq.responseText;
54:       //writeln('debug response '+HttpReq.GetAllResponseHeaders);
55:     finally
56:       httpreq:= unassigned;
57:     end;
58: end;
59:
60: For example from en to de:
61: Germany's conundrum over its ties with USA
62: Deutschlands Rätsel um seine Beziehungen zu den USA
63:
64: writeln(getPostDetectLang('con mucho cuidado'));
65: [{"confidence":94.0,"language":"es"}]
66:

```

```

67: writeln(utf8toAnsi(getPostTranslateLibre3('It is important to note this does nothing to
prevent someone from making an API request with your key','en','es'))); //}
68:   writeln(utf8toAnsi(getPostTranslateLibre3('Germany's conundrum over its ties with USA','
69:     'en','it')));
70:   writeln(utf8toAnsi(getPostTranslateLibre3('Germany's conundrum over its ties with USA','
71:     'en','fr')));
72:   writeln(utf8toAnsi(getPostTranslateLibre3('Conundrum','en','de')));
73:
74: {"translatedText":"Es importante se\u00f1alar que esto no hace nada para evitar que alguien
haga una solicitud de API con su clave,"}
75: {"translatedText":"Il conundrum tedesco sui suoi legami con gli Stati Uniti,"}
76: {"translatedText":"Le conundrum allemand sur ses liens avec les USA,"}
77: {"translatedText":"Conundrum"}
78:
79: What I did not resolve is the hex unicode: \u00f1 in Es importante se\u00f1alar que esto no hace
80: https://www.compart.com/en/unicode/U+00F1
81: U+00F1 is the unicode hex value of the character Latin Small Letter N with Tilde. Char U+00F1,
Encodings, HTML Entities:ñ,ñ,ñ, UTF-8 (hex), UTF-16 (hex), UTF-32 (hex).
82: This question relates to versions of Delphi below 2009 (ie without Unicode support built in).
I have a specification that requires me to transmit a Unicode encoded string over a TCP
connection.
83:
84: So the last word seems type unknown to argos: Definitions of conundrum
85: noun 1 a confusing and difficult problem or question.
86: "one of the most difficult conundrums for the experts"
87: Translations of conundrum - Part of speech Translation Reverse translations Frequency
88: help_outline:
89: noun das Rätsel as puzzle, mystery, riddle, enigma, conundrum, problem
90:
91: LibreTranslate supports per-user limit quotas, e.g. you can issue API keys to users so that
they can enjoy higher requests limits per minute (if you also set --req-limit). By default all
users are rate-limited based on --req-limit, but passing an optional api_key parameter to the
REST endpoints allows a user to enjoy higher request limits.
92:
93: Then we add some business goal to the service:
94: - a language translator/detector to fulfill a sentiment cluster analysis
95:
96: The idea behind POST microservices is that some types of applications become easier to build
and maintain when they are broken down into smaller, composable pieces which work together.
Each component is developed separately, and the application is then simply the sum of its
constituent components.
97: For example we get a list of the astronauts currently in space:
98:
99: //https://www.codeproject.com/Articles/5319146/How-to-Use-Rest-API-with-Python
100: procedure getAstronautAPI;
101: var res: string; i:integer;
102: begin
103: with TWinApiDownload.create do begin
104:   Useragent:= 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1';
105:   Url:= 'http://api.open-notify.org/astros.json';
106:   Download1(res);
107:   with TJSONObject.create4(res) do begin
108:     //writeln(tostring2(2,3))
109:     if getstring('message') = 'success' then
110:       for i:= 0 to getJSONArray('people').length-1 do
111:         writeln(itoa(i)+' ':'+getJSONArray('people').getJSONObject(i).getString('name'));
112:       free;
113:     end;
114:   free; //ApiDownload
115: end;
116: end;
117:
118: 0:Raja Chari
119: 1:Tom Marshburn
120: 2:Kayla Barron
121: 3:Matthias Maurer
122: 4:Oleg Artemyev
123: 5:Denis Matveev
124: 6:Sergey Korsakov
125: 7:Michael Lopez-Alegria
126: 8:Larry Connor
127: 9:Mark Pathy
128: 10:Eytan Stibbe
129:
130: Source of the translate script at:

```

