Package ‘deeplr’

May 29, 2018

Type Package

Title Interface to the 'DeepL' Translation API

Version 1.0.0

Description
A wrapper for the 'DeepL' API, a web service for translating texts between different languages. Access to the official API (see <https://www.deepl.com/translator>) is subject to a monthly fee. No authentication key is required for the undocumented DeepL JSON-RPC API. The package provides functions for both types of API calls.

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL <https://www.deepl.com/translator>

BugReports https://github.com/zumbov/deeplr/issues

RoxygenNote 6.0.1

Imports utf8, httr, tibble, rjson, purrr, tokenizers, stringr

Suggests dplyr

NeedsCompilation no

Author David Zumbach [aut, cre]

Maintainer David Zumbach <david.zumbach@gfzb.ch>

Repository CRAN

Date/Publication 2018-05-28 22:09:38 UTC

R topics documented:

detect ......................................................... 2
detect2 ....................................................... 3
pimp ......................................................... 3
pimp2 ....................................................... 4
toDutch ..................................................... 5
Detect the language of a text using the official DeepL Translator API

Description

detect detects the language of a text using the official DeepL Translator API. English, German, French, Spanish, Italian, Dutch and Polish are currently available. To use this service, an authentication key is required.

Usage

detect(text, auth_key = "your_key")

Arguments

text text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

auth_key DeepL authentication key which provides access to the API.

Examples

detect("My name is Hans.", auth_key = "my_key")
detect2

Detect the language of a text using DeepL

**Description**

detect2 detects the language of a text using the undocumented JSON-RPC DeepL API. English, German, French, Spanish, Italian, Dutch and Polish are currently available. No authentication key is required to use this service.

**Usage**

detect2(text)

**Arguments**

text
text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

**Examples**

detect2("My name is Hans.")

pimp

Fix and improve texts using the official DeepL Translator API

**Description**

pimp translates a text into a help language and then back into the original language using the official DeepL Translator API. To use this service, an authentication key is required.

**Usage**

pimp(text, help_lang, auth_key = "your_key")

**Arguments**

text
text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

help_lang
language used as a help language for reverse translation. Can be one of the following:

- EN English
- DE German
auth_key

DeepL authentication key which provides access to the API.

Examples

pimp("In former times I lived in Zurich", help_lang = "DE", auth_key = "my_key")

Description

pimp2 translates a text into a help language and then back into the original language using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

pimp2(text, help_lang)

Arguments

text
text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

help_lang
language used as a help language for reverse translation. Can be one of the following:

- EN English
- DE German
- FR French
- ES Spanish
- IT Italian
- NL Dutch
- PL Polish

Examples

pimp2("In former times I lived in Zurich", help_lang = "DE")
toDutch  

Translate texts into Dutch using the official DeepL Translator API

**Description**

`toDutch` translates a text from English, German, French, Spanish, Italian or Polish into Dutch using the official DeepL Translator API. To use this service, an authentication key is required.

**Usage**

```python
toDutch(text, source_lang = null, tag_handling = null, split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE, auth_key = "your_key")
```

**Arguments**

- **text**
  - text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

- **source_lang**
  - language of the text to be translated. Can be one of the following:
    - EN English
    - DE German
    - FR French
    - ES Spanish
    - IT Italian
    - PL Polish
    - If parameter is `null`, the API will try to detect the language of the source

- **tag_handling**
  - if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

- **split_sentences**
  - if `TRUE`, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to `FALSE` to prevent the engine from unintentionally splitting the sentence.

- **preserve_formatting**
  - if `TRUE`, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

- **get_detect**
  - if `TRUE`, the language detected for the source text is also included in the response. It corresponds to the value of the argument `source_lang` if it was specified. If `FALSE`, only the translated text is returned.

- **auth_key**
  - DeepL authentication key which provides access to the API.
Details

To get an authentication key, you need to register for a DeepL Pro account ([https://www.deepl.com/pro.html](https://www.deepl.com/pro.html)). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see [https://www.deepl.com/pro-pricing.html](https://www.deepl.com/pro-pricing.html)).

Value

If `get_detect` is set to `FALSE` a character vector containing the translation is returned. Otherwise, a `data.frame` (tibble::tibble) is returned with the following columns:

- `translation` the translated text.
- `source_lang` detected or specified language of the input text.

References

DeepL API documentations

Examples

```r
# Simple translation
toDutch("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
translator1 <- function(t) toDutch(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toDutch(text = t, get_detect = TRUE, auth_key = "x")
purrr::map_df(txt2, translator2)
```

toDutch2

Translates texts into Dutch using DeepL

Description

toDutch2 translates a text from English, German, French, Spanish, Italian or Polish into Dutch using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

toDutch2(text, source_lang = NULL, get_detect = FALSE)
Arguments

**text**

Text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

**source_lang**

Language of the text to be translated. Can be one of the following:

- EN English
- DE German
- FR French
- ES Spanish
- IT Italian
- PL Polish

If parameter `is.null`, the API will try to detect the language of the text.

**get_detect**

If TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument `source_lang` if it was specified. If FALSE, only the translated text is returned.

Value

If `get_detect` is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- **translation** the translated text(s).
- **source_lang** detected or specified language of the input text.

Examples

```r
# Simple translation
toDutch2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
purrr::map_chr(txt1, toDutch2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toDutch2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
```
toEnglish  
*Translate texts into English using the official DeepL Translator API*

**Description**

`toEnglish` translates a text from German, French, Spanish, Italian, Dutch or Polish into English using the official DeepL Translator API. To use this service, an authentication key is required.

**Usage**

```python
toEnglish(text, source_lang = NULL, tag_handling = NULL,
          split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE,
          auth_key = "your_key")
```

**Arguments**

- **text**  
  text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

- **source_lang**  
  language of the text to be translated. Can be one of the following:
  - DE German
  - FR French
  - ES Spanish
  - IT Italian
  - NL Dutch
  - PL Polish
  If parameter is `null`, the API will try to detect the language of the source.

- **tag_handling**  
  if set to "xml" , the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

- **split_sentences**  
  if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.

- **preserve_formatting**  
  if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

- **get_detect**  
  if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument `source_lang` if it was specified. If FALSE, only the translated text is returned.

- **auth_key**  
  DeepL authentication key which provides access to the API.
Details

To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation the translated text.
- source_lang detected or specified language of the input text.

References

DeepL API documentations

Examples

```r
# Simple translation
toEnglish("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("Mein Name ist Albert.", "Ich bin Physiker.", "Ich wurde 1879 in Ulm geboren.")
translator1 <- function(t) toEnglish(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Me llamo Fred.", "Je suis médecin.", "Ich komme aus der Schweiz.")
translator2 <- function(t) toEnglish(text = t, get_detect = T, auth_key = "x")
purrr::map_df(txt2, translator2)
```

toEnglish2  

Translate texts into English using DeepL

Description

toEnglish2 translates a text from German, French, Spanish, Italian, Dutch or Polish into English using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

toEnglish2(text, source_lang = NULL, get_detect = FALSE)
Arguments

**text**  text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

**source_lang**  language of the text to be translated. Can be one of the following:

- DE German
- FR French
- ES Spanish
- IT Italian
- NL Dutch
- PL Polish

If parameter `is.null`, the API will try to detect the language of the text.

**get_detect**  if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument `source_lang` if it was specified. If FALSE, only the translated text is returned.

Value

If `get_detect` is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- **translation**  the translated text(s).
- **source_lang**  detected or specified language of the input text.

Examples

```r
# Simple translation
toEnglish2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("Mein Name ist Albert.", "Ich bin Physiker.", "Ich wurde 1879 in Ulm geboren.")
purrr::map_chr(txt1, toEnglish2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Me llamo Fred.", "Je suis médecin.", "Ich komme aus der Schweiz.")
translator2 <- function(t) toEnglish2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
```
toFrench

Translate texts into French using the official DeepL Translator API

Description

toFrench translates a text from English, German, Spanish, Italian, Dutch or Polish into French using the official DeepL Translator API. To use this service, an authentication key is required.

Usage

toFrench(text, source_lang = NULL, tag_handling = NULL, split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE, auth_key = "your_key")

Arguments

text text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

source_lang language of the text to be translated. Can be one of the following:

• EN English
• DE German
• ES Spanish
• IT Italian
• NL Dutch
• PL Polish

If parameter is null, the API will try to detect the language of the source text.

tag_handling if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

split_sentences if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.

preserve_formatting if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

get_detect if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

auth_key DeepL authentication key which provides access to the API.
Details

To get an authentication key, you need to register for a DeepL Pro account ([https://www.deepl.com/pro.html](https://www.deepl.com/pro.html)). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see [https://www.deepl.com/pro-pricing.html](https://www.deepl.com/pro-pricing.html)).

Value

If `get_detect` is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation: the translated text.
- source_lang: detected or specified language of the input text.

References

DeepL API documentations

Examples

```r
# Simple translation
toFrench("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
translator1 <- function(t) toFrench(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Me llamo Fred.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toFrench(text = t, get_detect = T, auth_key = "x")
purrr::map_df(txt2, translator2)
```

---

**toFrench2**

*Translate texts into French using DeepL*

Description

`toFrench2` translates a text from English, German, Spanish, Italian, Dutch or Polish into French using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

`toFrench2(text, source_lang = NULL, get_detect = FALSE)`
Arguments

text  text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

source_lang  language of the text to be translated. Can be one of the following:
  • EN English
  • DE German
  • ES Spanish
  • IT Italian
  • NL Dutch
  • PL Polish

If parameter is_null, the API will try to detect the language of the text.

get_detect  if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

  • translation the translated text(s).
  • source_lang detected or specified language of the input text.

Examples

# Simple translation
toFrench2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
purrr::map_chr(txt1, toFrench2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Me llamo Fred.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toFrench2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
**Description**

toGerman translates a text from English, French, Spanish, Italian, Dutch or Polish into German using the official DeepL Translator API. To use this service, an authentication key is required.

**Usage**

toGerman(text, source_lang = NULL, tag_handling = NULL, split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE, auth_key = "your_key")

**Arguments**

text | text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

source_lang | language of the text to be translated. Can be one of the following:

- EN English
- FR French
- ES Spanish
- IT Italian
- NL Dutch
- PL Polish

If parameter is.null, the API will try to detect the language of the source

tag_handling | if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

split_sentences | if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.

preserve_formatting | if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

get_detect | if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

auth_key | DeepL authentication key which provides access to the API.
Details
To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

Value
If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:
- translation the translated text.
- source_lang detected or specified language of the input text.

References
DeepL API documentations

Examples

```r
# Simple translation
toGerman("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
translator1 <- function(t) toGerman(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Me llamo Fred.", "Je suis médecin.", "I'm from Wales")
translator2 <- function(t) toGerman(text = t, get_detect = T, auth_key = "x")
purrr::map_df(txt2, translator2)
```

toGerman2

Translate texts into German using DeepL

Description
toGerman2 translates a text from English, French, Spanish, Italian, Dutch or Polish into German using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage
toGerman2(text, source_lang = NULL, get_detect = FALSE)
toGerman2

Arguments

text text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

source_lang language of the text to be translated. Can be one of the following:

- EN English
- FR French
- ES Spanish
- IT Italian
- NL Dutch
- PL Polish

If parameter is.null, the API will try to detect the language of the text.

get_detect if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation the translated text(s).
- source_lang detected or specified language of the input text.

Examples

# Simple translation
toGerman2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
purrr::map_chr(txt1, toGerman2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Me llamo Fred.", "Je suis médecin.", "I'm from Wales")
translator2 <- function(t) toGerman2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
**toItalian**

Translate texts into Italian using the official DeepL Translator API

**Description**

toItalian translates a text from English, German, French, Spanish, Dutch or Polish into Italian using the official DeepL Translator API. To use this service, an authentication key is required.

**Usage**

toItalian(text, source_lang = NULL, tag_handling = NULL,
split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE,
auth_key = "your_key")

**Arguments**

text  
text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

source_lang  
language of the text to be translated. Can be one of the following:
- EN English
- DE German
- FR French
- ES Spanish
- NL Dutch
- PL Polish

If parameter is NULL, the API will try to detect the language of the source

tag_handling  
if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

split_sentences  
if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.

preserve_formatting  
if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

get_detect  
if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

auth_key  
DeepL authentication key which provides access to the API.
Details

To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation the translated text.
- source_lang detected or specified language of the input text.

References

DeepL API documentations

Examples

# Simple translation
toItalian("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
translator1 <- function(t) toItalian(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toItalian(text = t, get_detect = T, auth_key = "x")
purrr::map_df(txt2, translator2)
Arguments

- **text**: text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.
- **source_lang**: language of the text to be translated. Can be one of the following:
  - EN English
  - DE German
  - FR French
  - ES Spanish
  - NL Dutch
  - PL Polish

If parameter `is.null`, the API will try to detect the language of the text.

- **get_detect**: if `TRUE`, the language detected for the source text is also included in the response. It corresponds to the value of the argument `source_lang` if it was specified. If `FALSE`, only the translated text is returned.

Value

If `get_detect` is set to `FALSE` a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- **translation**: the translated text(s).
- **source_lang**: detected or specified language of the input text.

Examples

```r
# Simple translation
toItalian2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
purrr::map_chr(txt1, toItalian2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toItalian2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
```
toPolish

Translate texts into Polish using the official DeepL Translator API

Description
toPolish translates a text from English, German, French, Spanish, Italian or Dutch into Polish using the official DeepL Translator API. To use this service, an authentication key is required.

Usage
toPolish(text, source_lang = NULL, tag_handling = NULL,
        split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE,
        auth_key = "your_key")

Arguments

- **text**: text to be translated. Only UTF-8 encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.
- **source_lang**: language of the text to be translated. Can be one of the following:
  - EN English
  - DE German
  - FR French
  - ES Spanish
  - IT Italian
  - NL Dutch
  If parameter is null, the API will try to detect the language of the source
- **tag_handling**: if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.
- **split_sentences**: if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.
- **preserve_formatting**: if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.
- **get_detect**: if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.
- **auth_key**: DeepL authentication key which provides access to the API.
Details

To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation the translated text.
- source_lang detected or specified language of the input text.

References

DeepL API documentations

Examples

# Simple translation
toPolish("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
translator1 <- function(t) toPolish(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toPolish(text = t, get_detect = T, auth_key = "x")
purrr::map_df(txt2, translator2)

---

toPolish2  
Translate texts into Polish using DeepL

Description

toPolish2 translates a text from English, German, French, Spanish, Italian or Dutch into Polish using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

toPolish2(text, source_lang = NULL, get_detect = FALSE)
Arguments

text text to be translated. Must not exceed 5000 characters. Only UTF-8-encoded plain text is supported. May contain multiple sentences.

source_lang language of the text to be translated. Can be one of the following:

• EN English
• DE German
• FR French
• ES Spanish
• IT Italian
• NL Dutch

If parameter is null, the API will try to detect the language of the text.

get_detect if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

• translation the translated text(s).
• source_lang detected or specified language of the input text.

Examples

# Simple translation
toPolish2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
purrr::map_chr(txt1, toPolish2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toPolish2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
toSpanish

Translate texts into Spanish using the official DeepL Translator API

Description

toSpanish translates a text from English, German, French, Italian, Dutch or Polish into Spanish using the official DeepL Translator API. To use this service, an authentication key is required.

Usage

toSpanish(text, source_lang = NULL, tag_handling = NULL, split_sentences = TRUE, preserve_formatting = FALSE, get_detect = FALSE, auth_key = "your_key")

Arguments

text text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

source_lang language of the text to be translated. Can be one of the following:

• EN English
• DE German
• FR French
• IT Italian
• NL Dutch
• PL Polish

If parameter is null, the API will try to detect the language of the source

tag_handling if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

split_sentences if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.

preserve_formatting if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

get_detect if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

auth_key DeepL authentication key which provides access to the API.
Details

To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation the translated text.
- source_lang detected or specified language of the input text.

References

DeepL API documentations

Examples

```r
# Simple translation
toSpanish("Hallo Welt!", auth_key = "my_key")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
translator1 <- function(t) tospanish(text = t, auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toSpanish(text = t, get_detect = TRUE, auth_key = "x")
purrr::map_df(txt2, translator2)
```

toSpan2

Translate texts into Spanish using DeepL

Description

toSpan2 translates a text from English, German, French, Italian, Dutch or Polish into Spanish using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

toSpan2(text, source_lang = NULL, get_detect = FALSE)
Arguments

- **text**: text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.
- **source_lang**: language of the text to be translated. Can be one of the following:
  - EN English
  - DE German
  - FR French
  - IT Italian
  - NL Dutch
  - PL Polish

If parameter `is_null`, the API will try to detect the language of the text.

- **get_detect**: if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument `source_lang` if it was specified. If FALSE, only the translated text is returned.

Value

If `get_detect` is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- **translation**: the translated text(s).
- **source_lang**: detected or specified language of the input text.

Examples

```r
# Simple translation
toSpanish2("Hallo Welt!")

# Customized translator applied to multiple strings
txt1 <- c("My name is Albert.", "I'm a physicist.", "I was born in 1879 in Ulm.")
purrr::map_chr(txt1, toSpanish2)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("Je m'appelle Jean.", "Ich bin Arzt.", "I'm from Wales")
translator2 <- function(t) toSpanish2(text = t, get_detect = T)
purrr::map_df(txt2, translator2)
```
**translate**

*Translate texts using the official DeepL Translator API*

**Description**

translate2 translates texts between English, German, French, Spanish, Italian, Dutch and Polish using the official DeepL Translator API. To use this service, an authentication key is required.

**Usage**

```java
translate(text, source_lang = NULLL, target_lang = "en",
         tag_handling = NULL, split_sentences = TRUE,
         preserve_formatting = FALSE, get_detect = FALSE, auth_key = "your_key")
```

**Arguments**

text  

text to be translated. Only UTF8-encoded plain text is supported. May contain multiple sentences. The request size should not exceed 30kbytes.

source_lang  

language of the text to be translated (see below). If parameter is null, the API will try to detect the language of the source.

target_lang  

language into which to translate. Can be one of the following:

- EN English
- DE German
- FR French
- ES Spanish
- IT Italian
- NL Dutch
- PL Polish

tag_handling  

if set to "xml", the translation engine tries to find matches for XML enclosed words in the translated sentence and enclose them with the same tags. If no matching words are found, the tags are removed.

split_sentences  

if TRUE, the translation engine splits the input into sentences. If only one sentence is translated, it is recommended to set to FALSE to prevent the engine from unintentionally splitting the sentence.

preserve_formatting  

if TRUE, the translation engine tries to preserve some aspects (e.g. punctuation at the beginning and end of the sentence, upper/lower case at the beginning of the sentence) of the formatting.

get_detect  

if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

auth_key  

DeepL authentication key which provides access to the API.
Details
To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

Value
If `get_detect` is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- `translation` the translated text.
- `source_lang` detected or specified language of the input text.

References
DeepL API documentations

Examples

```r
# Simple translation
text <- "Hallo Welt!"
target_lang <- "en"
auth_key <- "my_key"
translated_text <- translate(text, target_lang = target_lang, auth_key = auth_key)

# Customized translator applied to multiple strings
txt1 <- c("Mein Name ist Albert.", "Ich bin Physiker.", "Ich wurde 1879 in Ulm geboren.")
translator1 <- function(t) translate(text = t, target_lang = "fr", auth_key = "x")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("My name is Fred.", "Je suis médecin.", "Ich komme aus der Schweiz.")
translator2 <- function(t) translate(text = t, target_lang = "es", get_detect = T, auth_key = "x")
purrr::map_df(txt2, translator2)
```

translate2

Translation texts using DeepL

Description
translate2 translates texts between English, German, French, Spanish, Italian, Dutch and Polish using the undocumented JSON-RPC DeepL API. No authentication key is required to use this service.

Usage

```r
translate2(text, source_lang = NULL, target_lang = "EN",
get_detect = FALSE)
```
Arguments

text  text to be translated. Must not exceed 5000 characters. Only UTF8-encoded plain text is supported. May contain multiple sentences.

source_lang  language of the text to be translated (see below). If parameter is NULL, the API will try to detect the language of the source.

target_lang  language into which to translate. Can be one of the following:
- EN English
- DE German
- FR French
- ES Spanish
- IT Italian
- NL Dutch
- PL Polish

get_detect  if TRUE, the language detected for the source text is also included in the response. It corresponds to the value of the argument source_lang if it was specified. If FALSE, only the translated text is returned.

Value

If get_detect is set to FALSE a character vector containing the translation is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- translation the translated text(s).
- source_lang detected or specified language of the input text.

Examples

# Simple translation
translate2("Hallo Welt!", target_lang = "EN")

# Customized translator applied to multiple strings
txt1 <- c("Mein Name ist Albert.", "Ich bin Physiker.", "Ich wurde 1879 in Ulm geboren.")
translator1 <- function(t) translate2(text = t, target_lang = "FR")
purrr::map_chr(txt1, translator1)

# Customized translator applied to multiple strings (with language detection response)
txt2 <- c("My name is Fred.", "Je suis médecin.", "Ich komme aus der Schweiz.")
translator2 <- function(t) translate2(text = t, target_lang = "ES", get_detect = T)
purrr::map_df(txt2, translator2)
Retrieve current usage data of a DeepL Pro account

Description
usage returns the character usage and the configured limit for the current period of a DeepL Pro Account.

Usage
usage(auth_key = "your_key")

Arguments
auth_key authentication key of the corresponding DeepL Pro account.

Details
To get an authentication key, you need to register for a DeepL Pro account (https://www.deepl.com/pro.html). This currently costs 20 euros per month and allows the translation of 1,000,000 characters per month (see https://www.deepl.com/pro-pricing.html).

References
DeepL API documentations

Examples

# Simple translation
usage(auth_key = "my_key")
Index

detect, 2
detect2, 3

pimp, 3
pimp2, 4

toDutch, 5
toDutch2, 6
toEnglish, 8
toEnglish2, 9
toFrench, 11
toFrench2, 12
toGerman, 14
toGerman2, 15
toItalian, 17
toItalian2, 18
toPolish, 20
toPolish2, 21
toSpanish, 23
toSpanish2, 24
translate, 26
translate2, 27

usage, 29