API Specification Doc

(*Backend Developer test app*)

Version	Date	Author	Email
1.0	15-Oct-2012	Georgios Lymperopoulos	glympe77@gmai.com

Index

Methods

1. most_words

Using the Google Cloud API *(https://cloud.google.com/translate/docs/reference/rest)* detect the input language and translate it into the language defined in the *output_language* parameter.

Once this has been done take the translated sentence and re-arrange it to have as many words in it as possible in any order as long as the total amount characters in the final string is less than or equal to the *max_characters* parameter. This result will be the *final_string* response field.

Once this is done use Google Cloud API to re-translate the *final_string* back into the original language and pass it as *final_string_translated*.

Request

Method	URL
POST	/most_words

Type	Params	Values
POST	sentence	string
POST	output_language	string
POST	max_characters	integer

sentence
`sentence` must be sent with all client requests. The original sentence to translate.

output_language
`output_language` must be sent with all client requests. The language to translate the original sentence.

max_characters
`max_characters` must be sent with all client requests. The total amount of characters for the translated sentence (in any order).

Response

Status	Response

200	**Response will be an object which has the following structure.**
	{
	"original_sentence": <string>
	"final_string": <string>
	"final_string_translated": <string>
	"duration_ms": <float>
	}
	
	original_sentence (**string**) - The original sentence to translate.
	final_string (**string**) - The original sentence translated and to the output_language and rearranged.
	final_string_translated (**string**) - The final_string translated back to the original language.
	duration_ms (**float**) - The amount of time taken for process to complete
	
	An example response is:-
	{
	"original_sentence": "Mir geht's gut",
	"final_string": "I'm good",
	"final_string_translated": "Ich bin",
	"duration_ms": 1.252474069595337
	}

400	{"error":"Invalid JSON body!"}
400	{"error":"Cannot translate to same language."}
400	{"error":"Mac chars number is too low."}
404	{"error":"Page not found."}
422	{"error":"Unprocessable Entity."}

2. best_strategy

The game is played between two people in alternating turns. Both players are always able to 'see' the values of the array and you start first. At each turn the person playing needs to pick one number from the *game_state* array by choosing from either the element at *game_state*[0] or *game_state*[n-1]. Whichever option is chosen, the player playing the round gets to remove the element from the array and add its value to their score. Then the next player makes a move and makes their choice. The game progresses until the array is empty. The player with the highest score wins.
This endpoint determines the best strategy to win this game.

Request

Method	URL
POST	/best_strategy

Type	Params	Values
POST	game_state	array

game_state
The `auth_key` is an array containing an even number of (int, float) values representing the initial game state.

Response

Status	Response

200	**Response will be an object which has the following structure.**
	
	{
	"strategy": <array>,
	"duration_ms": <float>
	}
	
	strategy (**array**) - The strategy to win the game which includes the proper direction_choice (**array**), the value (**array**) of each choice and array_index (**array**) of those choices.

	It also shows the total score gathered for p1 and p2 respectively as `p1_score` (**int**) and `p2_score` (**int**)
	
	An example response is:-
	{
	"strategy": [
	{
	"direction_choice": [
	"P1: left",
	"P1: right",
	"P1: Remaining Item"
	],
	"value": {
	"P1: 3",
	"P2: 1",
	"P1: 9",
	},
	"array_index": [
	"P1: 0",
	"P2: 2",
	"P1: 1",
	],
	"p1_score": 14,
	"p2_score": 4
	}
	],
	"duration_ms": 0.00033593177795410156
	}

400	{"error":"Invalid JSON body!"}
404	{"error":"Page not found."}
422	{"error":"Unprocessable Entity."}

3. Improve - Extend

Most Words

Feature - Create a new translation

The user will be able to create new translation and later view them. When i'll request "POST /most_words" and add the payload :

```
{
        "sentence": <string>,
        "output_language":<string>,
        "max_characters":<integer>
}
```

, then the response code should be 201 and the translation will be stored in the database.

Feature - View a translation

When a translation has been created, when i request "GET /most_words/{id}", the response status code should be 200 and it will display the translation created with the passed {id} or throw 404 if {id} does not exist.

Feature - View all translations

When i request "GET /most_words", the response status code should be 200 and it will display the all the created translations.

Best Strategy

Feature - Create a new game

The user will be able to create new games and later view them. When i'll request "POST /best_strategy" and add the payload :

```
{
        "game_state" : <array>
}
```

, then the response code should be 201 and the game will be stored in the database.

Feature - View a game

When a game has been created, when i request "GET /best_strategy/{id}", the response status code should be 200 and it will display the game created with the passed {id} or throw 404 if {id} does not exist.

Feature - View all games

When i request “GET /best_strategy”, the response status code should be 200 and it will display the all the created games.

Features for both endpoints

Feature - Pagination

The list of created games or translations could become really long, hence pagination could be added for the GET /most_words and GET /best_strategy endpoints.

Feature - Form Validation

Creating a model and storing translations and battles to database will enable the ability to create validation checks with forms.

Feature - Add security

Add authorization via a token (JSON Web token for example) and secure endpoints.

Glossary

Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
- All response are in JSON format.
- All request parameters are mandatory unless explicitly marked as `[optional]`
- The type of values accepted for a *request* parameter are shown the the values column like this `[10|<any number>]` .The `|` symbol means *OR*. If the parameter is `[optional]`, the default value is shown in blue bold text, as **`10`** is written in `[10|<any number>]`.

Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

`200` - Success of some kind
`4XX` - Error occurred in client’s part
`500` - Error occurred in server’s part

Status Code	Description
200	OK
400	Bad request
404	Resource not found
422	Unprocessable Entity
500	Internal Server Error