**MidTerm API Specification Doc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 4-23-2019 | Abraham Jimenez | Initial draft |
| 2.0 | 5-19-2019 | Grace Kohn | Update |
| 3.0 | 5-19-2019 | Sriram Kashyap | Update |

## 

## 

## 

Index

Table of Contents

[1. main 4](#_Toc9188319)

[Request 4](#_Toc9188320)

[Response 4](#_Toc9188321)

[2. Descriptors 5](#_Toc9188322)

[Request 5](#_Toc9188323)

[Response 5](#_Toc9188324)

[3. Identify 6](#_Toc9188325)

[Request 6](#_Toc9188326)

[Response 7](#_Toc9188327)

[4. Translate 7](#_Toc9188328)

[Request 7](#_Toc9188329)

[Response 8](#_Toc9188330)

[5. Sentiment 8](#_Toc9188331)

[Request 9](#_Toc9188332)

[Response 9](#_Toc9188333)

[6. Main Topic 10](#_Toc9188334)

[Request 10](#_Toc9188335)

[Response 10](#_Toc9188336)

[7. Spell Check 11](#_Toc9188337)

[Request 11](#_Toc9188338)

[Response 12](#_Toc9188339)

[8. Parts of Speech 12](#_Toc9188340)

[Get Request 12](#_Toc9188341)

[Get Response 13](#_Toc9188342)

[Post Request 13](#_Toc9188343)

[Post Response 13](#_Toc9188344)

[9. Subjectivity 14](#_Toc9188345)

[Get Request 14](#_Toc9188346)

[Get Response 14](#_Toc9188347)

[Post Request 15](#_Toc9188348)

[Post Response 15](#_Toc9188349)

[10. Lemmatize 16](#_Toc9188350)

[Get Request 16](#_Toc9188351)

[Get Response 16](#_Toc9188352)

[Post Request 16](#_Toc9188353)

[Post Response 16](#_Toc9188354)

[Conventions 18](#_Toc9188355)

[Status Codes 18](#_Toc9188356)

## 

## 

## 

Methods

PRODUCTION ENVIRONMENT: http://3.19.76.216:8080

## main

Retrieve all the features and the parameters for each one of the features

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/main/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
|  |  |  |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the list of features. Each item array has the following structure.**  {  "Description": "Identify all the different descriptors and quantify the number of times they appear. The List will be sorted Descending. Available only in English",  "End Point": "/midterm/Descriptor",  "Parameters": {  "1. type": "method used for submitting the file: 'url' or 'str' - Mandatory",  "2. url": "URL from where to get the file - Optional",  "3. text": "Text to be processed if URL is not submited - Optional"  }  } |
| 50X | Sever Errors |

## Descriptors

Look in the text all the different descriptors and quantify the number of times they appear. The List will be sorted Descending. Available only in English.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/Descriptor/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  GET  GET | type  url  text | string  string  string |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the list of descriptors and the number of times it appears on the text. Each item in the response array has the following structure.**  {"columns":["Descriptor","Occurences"],"index":[0,1],"data":[["blue",1],["beautiful",1]]} |
| 400 | {"error": "For other than English the only options available are Translate/Identify language"  } |
| 400 | {"error":"Invalid URL"} |
| 400 | {"error":"URL not informed"} |
| 400 | {"error":"Time Out reaching for the file - please try again in a few minutes"} |
| 400 | {"error":"Text too big. The total number of admitted characters is "} – Max number of characters is a system parameter |
| 400 | {"error":"Input empty"} |
| 400 | {"error":"Possible Gibberish"} – Amount of Gibberish (word not in dictionary ) defined by system parameter |
| 400 | {"error":"Language not supperted/not possible to identify""} |
| 50X | Sever Errors |

## Identify

Identify the language of the text submitted

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/Identify/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  GET  GET | type  url  text | string  string  string |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the name of the language for the input text.**  {  "Result": "The language of the input is Spanish; Castilian"  } |
| 400 | {"error":"Invalid URL"} |
| 400 | {"error":"URL not informed"} |
| 400 | {"error":"Time Out reaching for the file - please try again in a few minutes"} |
| 400 | {"error":"Text too big. The total number of admitted characters is "} – Max number of characters is a system parameter |
| 400 | {"error":"Input empty"} |
| 400 | {"error":"Possible Gibberish"} – Amount of Gibberish (word not in dictionary ) defined by system parameter |
| 400 | {"error":"Language not supperted/not possible to identify""} |
| 50X | Sever Errors |

## Translate

Translate the input language to English

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/Translate/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  GET  GET | type  url  text | string  string  string |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the name of the language for the input text.**  [  {  "Original Text": "Hola Mundo"  },  {  "Translation Text": "Hello World"  }  ] |
| 400 | {"error":"Invalid URL"} |
| 400 | {"error":"URL not informed"} |
| 400 | {"error":"Time Out reaching for the file - please try again in a few minutes"} |
| 400 | {"error":"Text too big. The total number of admitted characters is "} – Max number of characters is a system parameter |
| 400 | {"error":"Input empty"} |
| 400 | {"error":"Possible Gibberish"} – Amount of Gibberish (word not in dictionary ) defined by system parameter |
| 400 | {"error":"Language not supperted/not possible to identify"} |
| 400 | {"error":"Translation only open for other than English"} |
| 50X | Sever Errors |

## Sentiment

Polarize the text submitted (whole text, sentences or words – based on the request-) with a value between -1 (negative) to 1 (positive). Available only in English

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/Sentiment/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  GET  GET  GET | Type  url  text  level | string  string  string  string |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the sentiment analysis:**   * **Total positive components and average positive value** * **Total negative components and average negative value** * **Overall**   {"columns":["Value","Sentiment Average","Total Elements"],"index":[0,1,2],"data":[["Overall",1.0,1],["Possitive",0.7,1],["Negative",0.0,0]]} |
| 400 | {"error":"Invalid URL"} |
| 400 | {"error":"URL not informed"} |
| 400 | {"error":"Time Out reaching for the file - please try again in a few minutes"} |
| 400 | {"error":"Text too big. The total number of admitted characters is "} – Max number of characters is a system parameter |
| 400 | {"error":"Input empty"} |
| 400 | {"error":"Possible Gibberish"} – Amount of Gibberish (word not in dictionary ) defined by system parameter |
| 400 | {"error":"Language not supperted/not possible to identify""} |
| 50X | Sever Errors |

## Main Topic

Identify the main topic of the text by finding the most common words.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/MainTopic/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  GET  GET | Type  url  text | string  string  string |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the main topic of the input text.**  [  {  "Original Text": "This is an NLP tool."  },  {  "Main Topic Text": "The main topic is nlp tool"  }  ] |
| 400 | {"error":"Invalid URL"} |
| 400 | {"error":"URL not informed"} |
| 400 | {"error":"Time Out reaching for the file - please try again in a few minutes"} |
| 400 | {"error":"Text too big. The total number of admitted characters is "} – Max number of characters is a system parameter |
| 400 | {"error":"Input empty"} |
| 400 | {"error":"Possible Gibberish"} – Amount of Gibberish (word not in dictionary ) defined by system parameter |
| 400 | {"error":"Language not supperted/not possible to identify""} |
| 50X | Sever Errors |

## Spell Check

Check spelling and correct any misspelled words.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | midterm/SpellCheck/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  GET  GET | Type  url  text | string  string  string |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the main topic of the input text.**  [  {  "Original Text": " Speelling eror."  },  {  "Corrected Text": "Spelling error"  }  ] |
| 400 | {"error":"Invalid URL"} |
| 400 | {"error":"URL not informed"} |
| 400 | {"error":"Time Out reaching for the file - please try again in a few minutes"} |
| 400 | {"error":"Text too big. The total number of admitted characters is "} – Max number of characters is a system parameter |
| 400 | {"error":"Input empty"} |
| 400 | {"error":"Possible Gibberish"} – Amount of Gibberish (word not in dictionary ) defined by system parameter |
| 400 | {"error":"Language not supperted/not possible to identify""} |
| 50X | Sever Errors |

## Parts of Speech

#### Pass a string to recognize the parts of speech

**GET/PartsOfSpeech/pos**

Swagger URL : <http://localhost:8080/>

## Get Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | **/PartsOfSpeech/pos** |

## Get Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the instructions for the POST call.**  {  "instructions": "The sample call to the POST method will take in as input a JSON with an element called input"  } |

**POST/PartsOfSpeech/pos**

## Post Request

|  |  |
| --- | --- |
| **Method** | **CURL** |
| **POST** | **curl -X POST "http://localhost:8080/PartsOfSpeech/pos" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"input\": \"Here is the String\"}"** |

## Post Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the Parts of speech tagged appropriately**  {  "pos": [  [  "Here",  "RB"  ],  [  "is",  "VBZ"  ],  [  "the",  "DT"  ],  [  "String",  "NN"  ]  ]  } |
| |  |  | | --- | --- | | 400 | ***Invalid Argument*** | | ***Invalid Argument*** |
| |  |  | | --- | --- | | 500 | ***Mapping Key Error*** | | ***Mapping Key Error*** |

## 9. Subjectivity

#### Pass a string to recognize the subjectivity or objectivity of the text

**GET/Subjectivity/sub**

Swagger URL : <http://localhost:8080/>

## Get Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | **/Subjectivity/sub** |

## Get Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the instructions for the POST call.**  {  "instructions": "The sample call to the POST method will take in as input a JSON with an element called input"  } |

**POST/Subjectivity/sub**

## Post Request

|  |  |
| --- | --- |
| **Method** | **CURL** |
| **POST** | **curl -X POST "http://localhost:8080/Subjectivity/sub" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"input\": \"The avengers end game as created a very good culmination for the series\"}"** |

## Post Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the Subjectivity Score for the** **input**  {  "subjectivityScore": 0.5900000000000001  } |
| |  |  | | --- | --- | | 400 | ***Invalid Argument*** | | ***Invalid Argument*** |
| |  |  | | --- | --- | | 500 | ***Mapping Key Error*** | | ***Mapping Key Error*** |

## 10. Lemmatize

#### Pass a string and its words will be lemmatized

**GET/Lemmatize/lemmatize**

Swagger URL : <http://localhost:8080/>

## Get Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | **/Lemmatize/lemmatize** |

## Get Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the instructions for the POST call.**  {  "instructions": "The sample call to the POST method will take in as input a JSON with an element called input"  } |

**POST/Lemmatize/lemmatize**

## Post Request

|  |  |
| --- | --- |
| **Method** | **CURL** |
| **POST** | **curl -X POST "http://localhost:8080/Lemmatize/lemmatize" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"input\": \"There are a number of strange running concerns of non involvement of local governments in the decision making processes lately\"}"** |

## Post Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object containing the Lemmatized words from the** **input**  {  "lemmatized\_word": "LEMMATIZED = There are a number of strange running concern of non involvement of local government in the decision making process lately"  } |
| |  |  | | --- | --- | | 400 | ***Invalid Argument*** | | ***Invalid Argument*** |
| |  |  | | --- | --- | | 500 | ***Mapping Key Error*** | | ***Mapping Key Error*** |

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.

## Status Codes

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

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |