

Introduction

This document describes the current programmatic API that is made available to some users of the Text Inspector product created by WebLingua.

The document is intended for a technical audience, and assumes an understanding of RESTful APIs, HTTPS, JSON and other technical details.

Further information

For more information, please contact your Web Lingua or Text Inspector account manager.

Changes for API v1 compared to beta

In moving out of beta there have been a few changes to the API that will require users changing their code in order to use the new live server:

- 1) The hostname for API calls has been changed to: <https://textinspector.com> ****required****
- 2) The API version number has been added to all API calls - e.g. /api/v1/createsession
- 3) The API route to analyse a document has been given a specific route name rather than just posting to / - e.g. /api/v1/newanalysis
- 4) The session cookie name has had it's spelling corrected - e.g. "textinpsector.session" -> "textinspector.session"

The changing of the hostname (1) is required to use the live server.

Code used on the Beta server will continue to work without the other changes listed, however we would recommend updating as soon as possible as support for the old Beta API will be withdrawn in the very near future.

Please note that if you were using the Beta server you will need to have your access enabled on the live server. To do that please follow the instructions in the next section.

Using the API

In order to use the API, if you haven't already you will need to register on the main TextInspector site and then request API access via the "API for Developers" page on the site.

Please do this by going to the following two links.

1. TextInspector Registration: <https://textinspector.com/account/register>

2. Request API Access: https://textinspector.com/help/?page_id=974

API URI

The base URI for all API requests is:

<https://textinspector.com>

e.g.

GET `https://textinspector.com/api/v1/createsession`

Text Analysis API flow

The API expects a user to submit a document once for analysis and then request one or more results for that document from the various parsers.

Therefore an example flow of API requests might be:

1. Create session
2. Submit new text for analysis
3. Get statistics parser results
4. Get tagger parser results
5. Submit new text for analysis
6. Get statistics parser results
7. Get tagger parser results
8. etc.

API methods

GET /api/v1/createsession

For each session that you use the API you need to first authenticate with the API to get a session ID, and that session ID needs to be remembered and passed as a cookie header to all subsequent requests in the session.

To authenticate and get the Session ID you have to make a **createsession** call, passing your username and password.

Note that both username and password should be URL encoded - e.g. @ -> %40 etc.

The response from this call will give you your "sessionid".

Curl example

```
curl -X GET
"https://user_email_address:password@textinspector.com/api/v1/createsession" -H "accept: application/json" -L
```

Example response

```
{
  "sessionid" : "731999172284994221689694364605242328"
}
```

Once you have your session ID all subsequent requests should have that value set as a "textinspector.session" cookie header.

E.g.

```
Cookie: textinspector.session={sessionid};
```

POST /api/v1/newanalysis

Submit a document for processing. Returns a **ctxId** and the count of the number of documents processed. Use the **ctxId** and appropriate document number in subsequent requests to get results from different parsers.

Curl example

```
curl -X POST "https://textinspector.com/api/v1/newanalysis" -H
"Content-Type: application/json" -H "accept: application/json" -H
"Cookie: textinspector.session={sessionid};" -L --data
'{"text":"Try+the+tool+out+with+this+paragraph+of+text+or+you+can+r
eplace+this+whole+text+with+some+text+of+your+choosing.%23This+is+a
+second+document","delimiter":"%23","split":"1","textmode":"Writing"}
'
```

Example response

```
{
  "response" : {
    "ctxId" : "AFD52E08-4DD5-11E8-9E8C-26C6F64F64F04",
    "doc_count" : 2,
    "resultType" : "api_start",
    "errors" : []
  },
  "name" : "api_start",
  "type" : "api_start"
}
```

JSON input parameters - submitted in the request body

name	use	example
------	-----	---------

text (required)	The document text to process.	"text": "This is a test doc#And this is another"
split	Set to one if submitting more than one text - multiple texts should be separated with the delimiter below.	"split": "1"
delimiter	The delimiter to use if submitting multiple texts. Only required if also using "split". Default=#	"delimiter": "#"
textmode	The text mode for the analysis - Writing, Reading, Listening. Default=Writing	"textmode": "Writing"

JSON response values		
name	use	example
ctxId	The Context to use for future requests. A context can contain >1 document	"ctxId": "AFD52E08-5DD5-11E8-9E8C-26C6F64F64F64F64"
doc_count	The number of documents found in the input text using the delimiters specified, if any	"doc_count": "2"
resultType	The type of response being sent	"resultType": "api_start"

Having submitted a document for processing, you can request results from the different parsers.

To do this use the context ID returned from the newanalysis POST and specify which document you'd like to parse.

GET /api/v1/{ctxID}/doc{docNum}/statistics

Get the results for the "Statistics" parser

Curl example
curl -X GET "https://textinspector.com/api/v1/{ctxID}/doc{docNum}/statistics"

```
-H "accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - statistics

```
{
  "exists" : 1,
  "type" : "statistics",
  "name" : "Statistics",
  "response" : {
    "documentId" : "doc1",
    "summary" : {
      "uniqueWordCount" : "264",
      "uniqueWordRatio" : "0.52",
      "avgWordsPerSentence" : "8.96",
      "numberCount" : "10",
      "wordCount" : "502",
      "averageSyllablesPerWord" : "2.69",
      "excludeDigits" : "false",
      "wordsMoreThanTwoSyllables" : "244",
      "sentenceCount" : "56",
      "averageSyllablesPerSentence" : "24.07",
      "syllableCount" : "1348",
      "digitCount" : "15",
      "syllablesPerHundredWords" : "268.53",
      "wordsMoreThanTwoSyllablesPercentage" : "48.61"
    },
    "readability_scores" : {
```

```
    "fleschKincaidGrade" : "19.59",
    "gunningFogIndex" : "23.03",
    "fleschReadingEase" : "-29.44"
  },
  "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
}
```

GET /api/v1/{ctxID}/doc{docNum}/errors

Get the results for the "Errors" parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxId}/doc{docNum}/errors" -H
"accept: application/json" -H "Cookie:
textinspector.session={sessionId};" -L
```

Example response - errors

```
{
  "name" : "Errors",
  "response" : {
    "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819",
    "spelling_errors" : [
      "repurpose",
      "functionalized",
      "envisioneer",
      "intermandated",
      "functionalities",
```

```
    "reintermediate",
    "Intrinsicly",
    "Monotonectally",
    "tailers",
    "strategize",
    "recaptiualize",
    "Holisticly",
    "centric",
    "vis",
    "reconceptualize",
    "Rapidiously",
    "Energistically",
    "deliverables",
    "infomediaries",
    "incentivize",
    "schemas",
    "productize",
    "mindshare",
    "scalable",
    "evisculate",
    "Synergistically"
  ],
  "summary" : {
    "perHundredWords" : "5.18",
    "total" : "26",
    "perSentence" : "0.46"
  },
```

```
    "documentId" : "doc1"
  },
  "exists" : 1,
  "type" : "errors"
}
```

GET /api/v1/{ctxID}/doc{docNum}/diversity

Get the results for the “Lexical Diversity” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxId}/doc{docNum}/diversity" -H
"accept: application/json" -H "Cookie:
textinspector.session={sessionId};" -L
```

Example response - diversity

```
{
  "response" : {
    "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819",
    "lexical_diversity" : {
      "vocd" : "200.00",
      "mltd" : "206.57"
    },
  },
}
```

```
    "documentId" : "doc1"
  },
  "name" : "Lexical Diversity",
  "type" : "diversity",
  "exists" : 1
}
```

GET /api/v1/{ctxID}/doc{docNum}/tagger

Get the results for the “Tagger” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctdId}/doc{docNum}/tagger" -H
"accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - tagger

```
{ "type" : "tagger",
  "exists" : 1,
  "response" : {
    "documentId" : "doc1",
    "parts_of_speech" : {
      "VVG" : {
        "tokens" : "6",
```

```
    "description" : "verb, gerund/participle",
    "unique_tokens" : "4"
  },
  "VVZ" : {
    "unique_tokens" : "1",
    "description" : "verb, present 3d p. sing.",
    "tokens" : "1"
  },
  "RB" : {
    "tokens" : "66",
    "description" : "adverb",
    "unique_tokens" : "34"
  },
  "NN" : {
    "tokens" : "106",
    "unique_tokens" : "59",
    "description" : "noun, singular or mass"
  },
  "VVD" : {
    "unique_tokens" : "4",
    "description" : "verb, past tense",
    "tokens" : "7"
  },
  "IN" : {
    "unique_tokens" : "10",
    "description" : "preposition/subord. conj.",
    "tokens" : "61"
```

```
},
"DAT" : {
  "description" : "determiner, article",
  "unique_tokens" : "2",
  "tokens" : "5"
},
"VV" : {
  "tokens" : "43",
  "description" : "verb, base form",
  "unique_tokens" : "36"
},
"CO" : {
  "unique_tokens" : "2",
  "description" : "coordinating conjunction",
  "tokens" : "5"
},
"VVP" : {
  "tokens" : "1",
  "description" : "verb, present, non-3rd p.",
  "unique_tokens" : "1"
},
"VVN" : {
  "tokens" : "6",
  "description" : "verb, past participle",
  "unique_tokens" : "6"
},
"JJ" : {
```

```
    "unique_tokens" : "57",
    "description" : "adjective",
    "tokens" : "93"
  },
  "JJS" : {
    "tokens" : "1",
    "description" : "adjective, superlative",
    "unique_tokens" : "1"
  },
  "FW" : {
    "unique_tokens" : "1",
    "description" : "foreign word",
    "tokens" : "2"
  },
  "NNS" : {
    "tokens" : "86",
    "description" : "noun plural",
    "unique_tokens" : "48"
  },
  "NP" : {
    "description" : "proper noun, singular",
    "unique_tokens" : "2",
    "tokens" : "2"
  },
  "CD" : {
    "tokens" : "10",
    "unique_tokens" : "5",
```

```
        "description" : "cardinal number"
    },
    "JJR" : {
        "tokens" : "1",
        "unique_tokens" : "1",
        "description" : "adjective, comparative"
    }
},
"summary" : {
    "verbalElementsPerSentence" : "0.93",
    "tokenRatio" : "0.53",
    "totalNounElements" : "109",
    "elements" : "274",
    "coverage" : "100.00",
    "coverageCounted" : "502",
    "nounElementsPerSentence" : "1.95",
    "types" : "264",
    "coverageNotCounted" : "0",
    "tokensPerType" : "1.90",
    "tokens" : "502",
    "totalVerbalElements" : "52"
},
"ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
},
"name" : "Tagger"
}
```

GET /api/v1/{ctxID}/doc{docNum}/metadiscourse

Get the results for the “Metadiscourse” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxID}/doc{docNum}/metadiscourse
" -H "accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - metadiscourse

```
{ "exists" : 1,
  "type" : "metadiscourse",
  "response" : {
    "documentId" : "doc1",
    "summary" : {
      "Total" : {
        "TypesPercent" : "1.52",
        "TokensPercent" : "1.52",
        "Tokens" : "4",
        "Types" : "4"
      },
      "Unlisted" : {
        "Tokens" : "488",
        "TokensPercent" : "97.21",
        "TypesPercent" : "98.48",
        "Types" : "259"
      }
    }
  }
}
```

```
    "Attitude marker" : {
      "TokensPercent" : "0.20",
      "Tokens" : "1",
      "TypesPercent" : "0.38",
      "Types" : "1"
    },
    "Logical connective" : {
      "Types" : "3",
      "TokensPercent" : "2.59",
      "Tokens" : "13",
      "TypesPercent" : "1.14"
    }
  },
  "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
},
"name" : "Metadiscourse"
}
```

GET /api/v1/{ctxID}/doc{docNum}/taggedlexical-bnc

Get the results for the “Lexical: BNC” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxId}/doc{docNum}/taggedlexical
-bnc" -H "accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - taggedlexical-bnc

```
{
  "name" : "Lexis: BNC",
  "response" : {
    "summary" : {
      "80thPercentileTypes" : "20010",
      "elementsIncludingHomonyms" : "272",
      "totalCountedTokensItems" : "451",
      "totalTokenItems" : "502",
      "meanLFCPerType" : "12657.05",
      "meanLFCPerToken" : "11908.49",
      "meanLFCPer100Token" : "1190849.00",
      "60thPercentileTypes" : "5690",
      "totalLexicalTokensFrequencyCount" : "5370729",
      "totalLexicalTypesFrequencyCount" : "3341462",
      "uniqueCountedItemTypes" : "238",
      "coverage" : "89.84",
      "coverageTokensNotCounted" : "51",
      "50thPercentileTypes" : "3770",
      "meanLFCPer100Types" : "1265705.30",
      "coverageTokensCounted" : "451",
      "uniqueItemTypes" : "264",
      "70thPercentileTypes" : "10270"
    },
    "documentId" : "doc1",
    "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
  },
  "exists" : 1,
}
```

```
"type" : "taggedlexical-bnc"
}
```

GET

/api/v1/{ctxID}/doc{docNum}/taggedlexical-coca

Get the results for the “Lexical: COCA” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctdId}/doc{docNum}/taggedlexical
-coca" -H "accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - taggedlexical-coca

```
{
  "type" : "taggedlexical-coca",
  "exists" : 1,
  "name" : "Lexis: COCA",
  "response" : {
    "documentId" : "doc1",
    "summary" : {
      "uniqueItemTypes" : "264",
      "meanLFCPer100Types" : "1117732.58",
      "coverageTokensCounted" : "451",
      "70thPercentileTypes" : "9430",
      "totalLexicalTokensFrequencyCount" : "4705681",
      "60thPercentileTypes" : "5190",
```

```
    "totalLexicalTypesFrequencyCount" : "2950814",
    "uniqueCountedItemTypes" : "238",
    "coverage" : "89.84",
    "50thPercentileTypes" : "3260",
    "coverageTokensNotCounted" : "51",
    "totalTokenItems" : "502",
    "meanLFCPerType" : "11177.33",
    "meanLFCPerToken" : "10433.88",
    "meanLFCPer100Token" : "1043388.25",
    "80thPercentileTypes" : "14320",
    "elementsIncludingHomonyms" : "272",
    "totalCountedTokensItems" : "451"
  },
  "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
}
```

GET /api/v1/{ctxID}/doc{docNum}/academicwordlist

Get the results for the “Lexical: AWL” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxId}/doc{docNum}/academicwordlist" -H "accept: application/json" -H "Cookie:
textinspector.session={sessionId};" -L
```

Example response - academicwordlist

```
{
  "response" : {
    "summary" : {
      "AWL 2" : {
        "Types" : "12",
        "Tokens" : "16",
        "TokensPercent" : "3.25",
        "TypesPercent" : "4.58"
      },
      "AWL 1" : {
        "Tokens" : "24",
        "TokensPercent" : "4.87",
        "TypesPercent" : "4.58",
        "Types" : "12"
      },
      "AWL Total" : {
        "TokenPercent" : "25.56",
        "TypePercent" : "27.48",
        "Types" : "72",
        "Tokens" : "126"
      },
      "AWL 9" : {
        "TypesPercent" : "1.53",
        "TokensPercent" : "1.22",
        "Tokens" : "6",
        "Types" : "4"
      },
    },
  },
}
```

```
"Unlisted" : {
  "Types" : "188",
  "Tokens" : "358",
  "TokensPercent" : "72.62",
  "TypesPercent" : "71.76"
},
"AWL 7" : {
  "Types" : "6",
  "TypesPercent" : "2.29",
  "Tokens" : "9",
  "TokensPercent" : "1.83"
},
"K1" : {
  "Types" : "1",
  "Tokens" : "7",
  "TokensPercent" : "1.42",
  "TypesPercent" : "0.38"
},
"AWL 4" : {
  "Types" : "8",
  "TypesPercent" : "3.05",
  "Tokens" : "15",
  "TokensPercent" : "3.04"
},
"AWL 8" : {
  "Types" : "5",
  "TypesPercent" : "1.91",
```

```
    "TokensPercent" : "2.03",
    "Tokens" : "10"
  },
  "AWL 3" : {
    "Tokens" : "12",
    "TokensPercent" : "2.43",
    "TypesPercent" : "3.05",
    "Types" : "8"
  },
  "AWL 5" : {
    "Types" : "11",
    "TokensPercent" : "4.67",
    "Tokens" : "23",
    "TypesPercent" : "4.20"
  },
  "K3" : {
    "Types" : "1",
    "Tokens" : "2",
    "TokensPercent" : "0.41",
    "TypesPercent" : "0.38"
  },
  "AWL 6" : {
    "Types" : "6",
    "Tokens" : "11",
    "TokensPercent" : "2.23",
    "TypesPercent" : "2.29"
  }
}
```

```
    },
    "documentId" : "doc1",
    "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
  },
  "name" : "Lexis: AWL",
  "type" : "academicwordlist",
  "exists" : 1
}
```

GET /api/v1/{ctxID}/doc{docNum}/lexical

Get the results for the “Lexis: EVP” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxId}/doc{docNum}/lexical" -H
"accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - lexical

```
{
  "name" : "Lexis: EVP",
  "response" : {
```

```
"documentId" : "doc1",
"summary" : {
  "A2" : {
    "tokens_percentage" : "11.80",
    "tokens" : "59",
    "types_percentage" : "10.65",
    "types" : "28"
  },
  "C2" : {
    "types_percentage" : "2.66",
    "types" : "7",
    "tokens_percentage" : "2.20",
    "tokens" : "11"
  },
  "C1" : {
    "types_percentage" : "7.98",
    "types" : "21",
    "tokens_percentage" : "6.60",
    "tokens" : "33"
  },
  "Unlisted" : {
    "types_percentage" : "38.02",
    "types" : "100",
    "tokens_percentage" : "35.00",
    "tokens" : "175"
  },
  "A1" : {
```

```
    "types_percentage" : "6.84",
    "types" : "18",
    "tokens" : "55",
    "tokens_percentage" : "11.00"
  },
  "B1" : {
    "tokens_percentage" : "13.20",
    "tokens" : "66",
    "types_percentage" : "14.07",
    "types" : "37"
  },
  "B2" : {
    "types_percentage" : "19.77",
    "types" : "52",
    "tokens_percentage" : "20.20",
    "tokens" : "101"
  }
},
"ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
},
"type" : "lexical",
"exists" : 1
}
```

GET /api/v1/{ctxID}/doc{docNum}/tiprofile

Get the results for the “Scorecard” parser.

Curl example

```
curl -X GET
"https://textinspector.com/api/v1/{ctxID}/doc{docNum}/tiprofile" -H
"accept: application/json" -H "Cookie:
textinspector.session={sessionid};" -L
```

Example response - tiprofile

```
{
  "response" : {
    "overall_rating" : {
      "Percentage" : "84.71",
      "CEFR level" : "D1",
      "Number of metrics used" : "17"
    },
    "documentId" : "doc1",
    "scorecard" : {
      "Propositional Density" : {
        "Verbal elements per sentence CEFR level" : "A1",
        "Verbal elements per sentence Score" : "0.93"
      },
      "Lexical Sophistication: Corpus of Contemporary American
English" : {
        "COCA: % of words (types) at 10-20K level Score" :
"11.76",
```

```
    "COCA 80th percentile (Types) CEFR level" : "D2",
    "COCA: % of words (types) at 10-20K level CEFR level" :
"D2",
    "COCA 50th percentile (Types) Score" : "3510.00",
    "COCA 70th percentile (Types) CEFR level" : "D2",
    "COCA 60th percentile (Types) Score" : "5830.00",
    "COCA 50th percentile (Types) CEFR level" : "D2",
    "COCA 80th percentile (Types) Score" : "16860.00",
    "COCA 60th percentile (Types) CEFR level" : "D2",
    "COCA 70th percentile (Types) Score" : "9870.00"
},
  "Lexical Sophistication: British National Corpus" : {
    "BNC: % of words (types) at 0-1K level CEFR level" :
"D2",
    "BNC 60th percentile (Types) CEFR level" : "D2",
    "BNC: % of words (types) at 0-1K level Score" :
"16.91",
    "BNC 70th percentile (Types) Score" : "10430.00",
    "BNC 80th percentile (Types) Score" : "20010.00",
    "BNC 50th percentile (Types) Score" : "3930.00",
    "BNC 50th percentile (Types) CEFR level" : "D2",
    "BNC: % of words (types) at 10-20K level Score" :
"12.13",
    "BNC 80th percentile (Types) CEFR level" : "D2",
    "BNC 70th percentile (Types) CEFR level" : "D2",
    "BNC: % of words (types) at 10-20K level CEFR level" :
"D2",
    "BNC 60th percentile (Types) Score" : "5690.00"
  },
```

```

    "Lexical Sophistication: English Vocabulary Profile" : {
      "EVP: % of words (types) at C1 level CEFR level" :
"D2",
      "EVP: % of words (types) at C1 level Score" : "7.98",
      "EVP: % of words (types) at A1 level CEFR level" :
"D2",
      "EVP: % of words (types) at B1 level CEFR level" :
"C1+",
      "EVP: % of words (types) at A1 level Score" : "6.84",
      "EVP: % of words (types) at B1 level Score" : "14.07"
    },
    "Statistics" : {
      "Average words per sentence Score" : "8.96",
      "Average words per sentence CEFR level" : "A2",
      "Average Syllables per sentence CEFR level" : "B2",
      "Average Syllables per sentence Score" : "24.07"
    }
  },
  "ctxId" : "FA8F842C-8B3B-11E8-8F83-C2659BF81819"
},
"name" : "Scorecard",
"exists" : 1,
"type" : "tiprofile"
}

```

Longer example document

This shows an example requests with >1000 words.

```
curl -X POST "https://textinspector.com/api/v1/" -H "accept: application/json" -H "Content-Type: application/json" -d '{"text": "5 May. I must have been asleep, for certainly if I had been fully awake I must have noticed the approach of such a remarkable place. In the gloom the courtyard looked of considerable size and as several dark ways led from it under great round arches, it perhaps seemed bigger than it really is. I have not yet been able to see it by daylight. When the calèche stopped, the driver jumped down and held out his hand to assist me to alight. Again I could not but notice his prodigious strength. His hand actually seemed like a steel vice that could have crushed mine if he had chosen. Then he took out my traps, and placed them on the ground beside me as I stood close to a great door, old and studded with large iron nails, and set in a projecting doorway of massive stone. I could see even in the dim light that the stone was massively carved, but that the carving had been much worn by time and weather. As I stood, the driver jumped again into his seat and shook the reins; the horses started forward, and trap and all disappeared down one of the dark openings. I stood in silence where I was, for I did not know what to do. Of bell or knocker there was no sign; through these frowning walls and dark window openings it was not likely that my voice could penetrate. The time I waited seemed endless, and I felt doubts and fears crowding upon me. What sort of place had I come to, and among what kind of people? What sort of grim adventure was it on which I had embarked? Was this a customary incident in the life of a solicitor's clerk sent out to explain the purchase of a London estate to a foreigner? Solicitor's clerk. Mina would not like that. Solicitor 'for just before leaving London I got word that my examination was successful; and I am now a full-blown solicitor. I began to rub my eyes and pinch myself to see if I were awake. It all seemed like a horrible nightmare to me, and I expected that I should suddenly awake, and find myself at home, with the dawn struggling in through the windows, as I had now and again felt in the morning after a day of overwork. But my flesh answered the pinching test, and my eyes were not to be deceived. I was indeed awake and among the Carpathians. All I could do now was to be patient, and to wait the coming of the morning. Just as I had come to this conclusion I heard a heavy step approaching behind the great door, and saw through the chinks the gleam of a coming light. Then there was the sound of rattling chains and the clanking of massive bolts drawn back. A key was turned with the loud grating noise of long disuse, and the great door swung back. Within, stood a tall old man, clean shaven save for a long white moustache, and clad in black from head to foot, without a single speck of colour about him anywhere. He held in his hand an antique silver lamp, in which the flame burned without chimney or globe of any kind, throwing long quivering shadows as it flickered in the draught of
```

the open door. The old man motioned me in with his right hand with a courtly gesture, saying in excellent English, but with a strange intonation: 'Welcome to my house. Enter freely and of your own will' He made no motion of stepping to meet me, but stood like a statue, as though his gesture of welcome had fixed him into stone. The instant, however, that I had stepped over the threshold, he moved impulsively forward, and holding out his hand grasped mine with a strength which made me wince, an effect which was not lessened by the fact that it seemed as cold as ice 'more like the hand of a dead than a living man. Again he said: 'Welcome to my house. Come freely. Go safely; and leave something of the happiness you bring' The strength of the handshake was so much akin to that which I had noticed in the driver, whose face I had not seen, that for a moment I doubted if it were not the same person to whom I was speaking; so to make sure, I said interrogatively: 'Count Dracula?' He bowed in a courtly way as he replied: 'I am Dracula; and I bid you welcome, Mr. Harker, to my house. Come in; the night air is chill, and you must need to eat and rest.' As he was speaking, he put the lamp on a bracket on the wall, and stepping out, took my luggage; he had carried it in before I could forestall him. I protested but he insisted: 'Nay, sir, you are my guest. It is late, and my people are not available. Let me see to your comfort myself.' He insisted on carrying my traps along the passage, and then up a great winding stair, and along another great passage, on whose stone floor our steps rang heavily. At the end of this he threw open a heavy door, and I rejoiced to see within a well-lit room in which a table was spread for supper, and on whose mighty hearth a great fire of logs, freshly replenished, flamed and flared. The Count halted, putting down my bags, closed the door, and crossing the room, opened another door, which led into a small octagonal room lit by a single lamp, and seemingly without a window of any sort. Passing through this, he opened another door, and motioned me to enter. It was a welcome sight; for here was a great bedroom well lighted and warmed with another log fire, 'also added to but lately, for the top logs were fresh 'which sent a hollow roar up the wide chimney. The Count himself left my luggage inside and withdrew, saying, before he closed the door: '\"}"

Response

```
{
  "response" : {
    "ctxId" : "89A7D820-6AEO-11E8-8B4F-8D1A62DF0F26",
    "resultType" : "api_start",
```

```
    "doc_count" : 1,
    "errors" : []
  },
  "name" : "api_start",
  "type" : "api_start"
}
```

Statistics request

```
curl -X GET
"https://textinspector.com/api/v1/89A7D820-6AE0-11E8-8B4F-8D1A62DF0
F26/doc{docNum}/statistics" -H "accept: application/json"
```

Response

```
{
  "response" : {
    "resultType" : "statistics",
    "documentId" : "doc1",
    "ctxId" : "89A7D820-6AE0-11E8-8B4F-8D1A62DF0F26",
    "summary" : {
      "syllablesPerHundredWords" : "130.69",
      "wordsMoreThanTwoSyllablesPercentage" : "5.08",
      "averageSyllablesPerSentence" : "33.42",
      "numberCount" : "1",
      "wordCount" : "1023",
      "avgWordsPerSentence" : "25.57",
      "sentenceCount" : "40",
      "averageSyllablesPerWord" : "1.31",
      "uniqueWordRatio" : "0.42",
      "syllableCount" : "1337",
      "digitCount" : "1",
      "wordsMoreThanTwoSyllables" : "52",
      "uniqueWordCount" : "430",
      "excludeDigits" : "false"
    }
  },
  "name" : "Statistics",
  "exists" : 1,
  "type" : "statistics"
}
```

Example PHP code for using the API

```
<?php

/**
 * Example PHP code for using the Text Inspector API from Weblingua
 */

$api_base_url = 'https://textinspector.com/api/v1';

$api_username = 'YOUR_USERNAME';
$api_password = 'YOUR_PASSWORD';

// Get session ID
$ch = curl_init($api_base_url.'/createsession');
curl_setopt($ch, CURLOPT_HTTPHEADER, array('accept:
application/json'));
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_USERPWD, $api_username . ":" .
$api_password);
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
$return = curl_exec($ch);
curl_close($ch);
$response = json_decode($return);
$sessionid = $response->sessionid;

if (!$sessionid) {
    print "Failed to get session ID\n";
    print_r($return);
    Exit;
}
print "Got Session ID:$sessionid\n";

// Submit a new text for analysis
$fields = array(
    'text' => urlencode('Try the tool out with this paragraph of text
here, by pressing Analyse below. Or you can replace this whole text
with some text of your choosing. Simply paste any text into this box,
or else type it in yourself! Text Inspector gives different scores for
```

Writing texts (student writing) and Reading and Listening texts (e.g. texts designed for classroom reading or listening). The default is Writing, so if your text is for Reading or Listening, please go to Advanced Options to change the mode and get accurate calculations. Look at our Subscriptions section for special offers and discounts.')

```
'delimiter' => urlencode('#'),
'split' => urlencode(1),
'textmode' => 'Listening'
);
$ch = curl_init($api_base_url.'/newanalysis');
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json','accept: application/json', 'Cookie:
textinpsector.session='.$sessionid));
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($ch,CURLOPT_POST, count($fields));
curl_setopt($ch,CURLOPT_POSTFIELDS, json_encode($fields));
$return = curl_exec($ch);
curl_close($ch);
$response = json_decode($return);
$ctxid = $response->response->ctxId;

if (!$ctxid) {
    print "Failed to get context ID\n";
    print_r($return);
    Exit;
}
print "Got Context ID: $ctx\n";

// Get statistics parser results
$ch = curl_init($api_base_url.'/'.$ctxid.'/doc1/statistics');
curl_setopt($ch, CURLOPT_HTTPHEADER, array('accept:
application/json'));
curl_setopt($ch, CURLOPT_HTTPHEADER, array("Cookie:
textinpsector.session=".$sessionid));
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
$return = curl_exec($ch);
curl_close($ch);
$parser_results = json_decode($return);

print_r($parser_results);
```

Document version history

Version	Date	Status	Reviewed by
1	April 2018	Draft	Nik
2	May 2018	V1.0 published at https://docs.google.com/document/d/1iLm7ND4yKd12iYmR-itnS3thCh1yLgRSBljGzLUJV6M/edit?usp=sharing	Nik
3	June 8 2018	V1.1 - remove 250 word limit for API users. Limit now 15,000	Nik
4	June 14 2018	V1.2 - added missing statistics entries, added taggedlexical-bnc, taggedlexical-coca, tagger, metadiscourse	Nik
5	July 16 2018	V1.3 - added missing academicwordlist, lexical and tiprofile . Added sessionId as part of authentication header	Mary
6	Sept 25 2018	V1.4 - updated details about accessing API and Session ID usage	Nick
7	April 30,2019	V1.5 - fixed typo in cookie session name	Mary
8	May 21 2019	V.1.5.1 - updated for changes to the API for v1 of the live API - first non-beta version	Nick