|  |
| --- |
|  |
| SENTIMO |
| Python SDK Documentation |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | May 2, 2018 | Initial Draft |
| 1.1 | June 8, 2018 | Add Section 2.4 Knowledge Management Module |

Contents

[Section 1 SENTIMO Python SDK Introduction 2](#_Toc516651691)

[Basics 2](#_Toc516651692)

[Evaluating responses 2](#_Toc516651693)

[Authentication 2](#_Toc516651694)

[Section 2 API References 3](#_Toc516651695)

[Module #1 class SecuredUpload 3](#_Toc516651696)

[Secured Upload Data 4](#_Toc516651697)

[Module #2 class GetSentiments 6](#_Toc516651698)

[General Sentiment Analysis 7](#_Toc516651699)

[Fine-grained Emotion Analysis 10](#_Toc516651700)

[Module #3 class AnalyzeOnTheFly 13](#_Toc516651701)

[General Sentiments Analysis 14](#_Toc516651702)

[Fine-grained Emotion Analysis 18](#_Toc516651703)

[Module #4 class KnowledgeManagement 22](#_Toc516651704)

[Add Dictionary 23](#_Toc516651705)

[Delete Dictionary 25](#_Toc516651706)

[Get Dictionary Item 26](#_Toc516651707)

[Get Dictionary List 27](#_Toc516651708)

[Update Dictionary 28](#_Toc516651709)

[Section 3 Status/Error Codes 30](#_Toc516651710)

[Section 4 Numeric Indicators in Sentiment Output 31](#_Toc516651711)

[Section 5 Numeric Indicators in Emotion Output 32](#_Toc516651712)

[Section 6 Other Outputs 33](#_Toc516651713)

[Section 7 Quick Start Examples 34](#_Toc516651714)

[Extracting the SDK folder 34](#_Toc516651715)

[Updating the configurations 34](#_Toc516651716)

[Running the examples 34](#_Toc516651717)

# Section 1 SENTIMO Python SDK Introduction

Welcome to the SENTIMO Python SDK Documentation.

The SENTIMO Python SDK lets you upload textual data and receive a list of associated sentiments and emotions.

This document provides the complete list of all the classes, methods, functions and parameters available as part of the SENTIMO Python SDK.

## Basics

The Python SDK is a collection of call methods to the SENTIMO platform. All class object should be constructed with appropriate attribute information before calling the included methods.

Arguments are:

* + **url**: the target server for SENTIMO
  + **token**: a valid token (PID) to authenticate the end user.
  + **Timeout (optional)** : waiting time for response

The detailed examples on various parameters are provided in the following chapter.

## Evaluating responses

All Python SDK responses contain a ‘dict’ object which is similar to the JSON format, and they will always include a top-level Boolean property *status*, indicating *success* or *failure*.

For a *failure* result, a 5 digit error code will be provided to help refer to the error information. The detailed information can be referred to in Chapter [*Status/Error codes*](#_5_Status/Error_Codes).

All other properties are defined in the respective documentation for each relevant method.

## Authentication

Authenticate your Web API requests by using the provided Partner ID token, which idetntifies a valid user account. The PID token should be sent using the HTTP header parameter of your outbound requests.

pid

# Section 2 API References

## Module #1 class SecuredUpload

This module helps with uploading target data to server.

>>> #Initial Python class  
>>> from sentimo\_developer\_cloud.secured\_upload import SecuredUpload  
>>> service = SecuredUpload(url = **TARGET\_URL**, token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| url | REQUIRED, class parameter |
| **String** | Used for service address communication |
| token | REQUIRED, class parameter |
| **String** | a valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used to specify wait time for response |

### Secured Upload Data

#### Function upload\_data (content):

###### Parameters

|  |  |
| --- | --- |
| content |  |
| **String** | Main content of data to be uploaded |

###### Example:

>>> response = service.upload\_data(‘Trains are very crowded today.’)

#### Function upload\_full\_dataset (data):

###### Parameters

|  |  |  |
| --- | --- | --- |
| data | *An array containing elements below:* | |
|  | post\_id | The domain name under which this set of results was produced |
| **String** |  |
| source | Source of data |
| **String** |  |
| user\_id | User ID of poster of data |
| **String** |  |
| post\_time | Time when post was uploaded to source |
|  | **String** |  |
|  | content | Main content of data |
|  | **String** |  |

###### Example:

input = {

"data":[{

"post\_id":"001",

"source":"DummySource1",

"user\_id":"user1",

"post\_time":"2016-02-08-19-03-01",

"content":"sample comment one"

},{

"post\_id":"002",

"source":"DummySource2",

"user\_id":"user2",

"post\_time":"2016-02-08-19-03-02",

"content":"sample comment two"

}]

}

>>> response = service.upload\_data(input)

#### Response

This call returns a list of Data IDs for uploaded data

###### Parameters

|  |  |
| --- | --- |
| status | Returns status of call |
| **String** | Possible values: 1, 1000, 1001, 1002, 1003, 1004 |
| data\_ids | Data ID returned by database |
| **integer** | -1 indicates that the particular data was not uploaded |

###### Sample Response

A sample result of a successful request is as follows:

{

"status": 1

"data\_ids":[621292,621293]

}

## Module #2 class GetSentiments

This module returns the results of general sentiment analysis of a particular Data ID if the *sentiment* module is called, and fine-grained emotion analysis if the *sentimo* module is called. To use these methods, the Data ID of the data must be known. There are two methods by which this call can be implemented. They both return the same data format.

>>> # Start Python class  
>>> from sentimo\_developer\_cloud.get\_sentiments import GetSentiments   
>>> service = GetSentiments(url = **TARGET\_URL**, token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| url | REQUIRED, class parameter |
| **String** | Used for service address communication |
| token | REQUIRED, class parameter |
| **String** | a valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used for how long the service response from servers. |

### General Sentiment Analysis

#### Function retrieve\_sentiment (dataIds, domains):

In this method, users can indicate the Data IDs of the data to retrieve the analysis results. Data IDs need not be in sequence. Users can also indicate their domain of interest in the request.

###### Parameters

|  |  |
| --- | --- |
| dataIds | REQUIRED |
| **String** | Multiple Data IDs to be separated by comma |
| domains | OPTIONAL |
| **String** | Multiple domains to be separated by comma |

###### Examples:

Without domain specification:

>>> response = service.retrieve\_sentiments(dataIds=’621236,621237,621238’)

With domain specification:

>>> response = service.retrieve\_sentiments(dataIds=’621236,621237,621238’, domain = ‘general,education’)

#### Function retrieve\_sentiment\_set (dataId, domain, num):

This method returns the analysis results of a series of Data IDs in sequence. Users can specify the starting Data ID and the number of data records they want analysed. Users can also indicate their domain of interest in the request and the maximum number of records to return. The default size of records returned is 20 if the *max* parameter is omited.The maximum size of records that can be returned in a single call is 100.

###### Parameters

|  |  |
| --- | --- |
| dataId | REQUIRED, header parameter |
| **String** | Data ID previously returned by database |
| domains | OPTIONAL |
| **String** | Multiple domains to be separated by comma |
| num | OPTIONAL |
| **Integer/**  **String** | Range of data set from calling dataId |

###### Examples:

Without domain specification:

>>> response = service.retrieve\_sentiments(dataIds=’621236’, num = 50)

With domain specification:

>>> response = service.retrieve\_sentiments(dataId=’621236’, domain = ‘general,education’, num = ‘50’)

#### Response

This response returns the result of sentiment analysis on a particular Data ID. It returns the sentiment attributed to a particular target object in JSON format.

###### Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| status | Returns status of call | | | | |
| **String** | 1 if successful. If not, one of the error codes in (1000,1001,1002,1003 or 1004) | | | | |
| results | *An array containing elements below:* | | | | |
|  | domain\_name | The domain name under which this set of results were produced | | | |
| **String** |  | | | |
| data | *An array containing elements below:* | | | |
|  | data\_id | ID of data, as stored in database | | |
|  |  | **long** |  |  | |
|  |  | objects | *An array containing elements below:* | | |
|  |  |  | target\_object | | The object towards which the sentiment is attributed |
|  |  |  | **String** | |  |
|  |  |  | sentiment | | The sentiment attributed to the particular data given the above domain name |
|  |  |  | **float** | | Possible Values:  0, 1, 2, 3, 4, 5 |
|  |  |  |  | |  |

###### Sample Response

{

"status":1,

"results": [{

"domain\_name":"transport",

"data": [{

"data\_id":621236,

"objects":[{

"target\_object":"defaultobject",

"sentiment":0.0

}]

}, {

"data\_id":621237,

"objects":[{

"target\_object":"defaultobject",

"sentiment":0.0

}]

}, {

"data\_id":621238,

"objects":[{

"target\_object":"defaultobject",

"sentiment":0.0

}]

}]

}, {

"domain\_name":"company-eval",

"data":[{

"data\_id":621236,

"objects":[{

"target\_object":"defaultobject",

"sentiment":0.0

}]

}, {

"data\_id":621237,

"objects":[{

"target\_object":"defaultobject",

"sentiment":0.0

}]

}, {

"data\_id":621238,

"objects":[{

"target\_object":"defaultobject",

"sentiment":0.0

}]

}]

}]

}

### Fine-grained Emotion Analysis

#### Function retrieve\_sentiment (dataIds, domain):

In this method, users can indicate the Data IDs of the data to retrieve the analysis results. Data IDs need not be in sequence. Users can also indicate their domain of interest in the request.

###### Parameters

|  |  |
| --- | --- |
| dataIds | REQUIRED |
| **String** | Multiple Data IDs to be separated by comma |
| domains | OPTIONAL |
| **String** | Multiple domains to be separated by comma |

###### Examples:

Without domain specification:

>>> response = service.retrieve\_sentimo(dataIds=’621236,621237,621238’)

With domain specification:

>>> response = service.retrieve\_sentimo\_set(dataId=’621236’, domain = ‘general,education’, num = 5)

#### Function retrieve\_sentiment\_set (dataId, domain, num):

This methods returns the analysis results of a series of Data IDs in sequence. Users can specify the starting Data ID and specify the number of data records they wish to retrieve, and the maximum number of records to return. The default size for records returned is 20, if the *max* parameter is omitted. The maximum size of records cannot exceed 100.

###### Parameters

|  |  |
| --- | --- |
| dataId | REQUIRED |
| **String** | Data ID previously returned by database |
| domains | OPTIONAL |
| **String** | Multiple domains to be separated by comma |
| num | OPTIONAL |
| **Integer /String** | Range of data set from calling dataId |

###### Examples:

Without domain specification:

>>> response = service.retrieve\_sentimo\_set(dataIds=’635913’, num = 2)

With domain specification:

>>> response = service.retrieve\_sentimo\_set(dataId=’635913’, domain = ‘transport’, num = 2)

#### Response

This call returns the result of sentiment analysis on data. It returns the sentiment attributed to a particular target object in JSON format.

###### Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| status | Returns status of call | | | | |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) | | | | |
| results | *An array containing elements below:* | | | | |
|  | domain\_name | The domain name under which this set of results were produced | | | |
| **String** |  | | | |
| data | *An array containing elements below:* | | | |
|  | data\_id | ID of data, as stored in database | | |
|  |  | **long** |  |  | |
|  |  | objects | *An array containing elements below:* | | |
|  |  |  | target\_object | | The object towards which the sentiment is attributed |
|  |  |  | **String** | |  |
|  |  |  | sentiment | | The sentiment attributed to the particular data given the above domain name |
|  |  |  | **float** | | Possible Values:  0, 1, 2, 3, 4, 5 |
|  |  |  |  | |  |

###### Sample Response

{

"results": [

{

"domain\_name": "transport",

"data": [

{

"data\_id": 635913,

"objects": [

{

"sentiment": 4,

"scores": [

{

"score": 1,

"category": "Positive"

},

{

"score": 0,

"category": "Negative"

},

{

"score": 0,

"category": "Anger"

},

{

"score": 0,

"category": "Anxiety"

},

{

"score": 0,

"category": "Sadness"

},

{

"score": 0,

"category": "Satisfaction"

},

{

"score": 0,

"category": "Happiness"

},

{

"score": 0,

"category": "Excitement"

}

],

"target\_object": "defaultobject"

}

]

}

]

}

],

"status": "1"

}

## Module #3 class AnalyzeOnTheFly

This module allows for immediate analysis of data sent in the request body, returning results of the general sentiment analysis if the sentiment module is called, and the fine-grained emotion analysis if the sentimo module is called.

>>> #Initial Python class  
>>> from sentimo\_developer\_cloud.analyze\_on\_the\_fly import AnalyzeOnTheFly  
>>> service = AnalyzeOnTheFly (url = **TARGET\_URL**, token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| url | REQUIRED, class parameter |
| **String** | Used for service address communication |
| token | REQUIRED, class parameter |
| **String** | a valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used for how long the service response from servers. |

General Sentiments Analysis  
  
Function analyze\_sentiment\_on\_the\_fly (content, target, domain):

In this method, users can send request with only the target analysis content (string or array of string) to retrieve the general sentiment results on the fly. Users can also indicate their targets or domain of interest in the request.

Parameters

|  |  |
| --- | --- |
| content | REQUIRED |
| **[String]/ String** | main content to be analyzed |
| domain | OPTIONAL |
| **String** | Multiple domains to be separated by comma |
| targets | OPTIONAL |
| **String** | Target objetcs, up to 5 targets can be defined |

###### Examples:

Without target object specification:

>>> service.analyze\_sentiment\_on\_the\_fly(content = ‘Love my new phone! Only that the battery runs out very fast’)

With target object specification:

>>> service.analyze\_sentimo\_on\_the\_fly(content = ‘Love my new phone! Only that the battery runs out very fast’, target = ‘sir’)

#### Function analyze\_sentiment\_full\_dataset\_on\_the\_fly (dataset, target, domain):

In this method, users can send request with full array dataset to retrieve the general sentiments results on the fly. Users can also indicate their targets or domain of interest in the request.

###### Parameters

|  |  |
| --- | --- |
| domain | OPTIONAL |
| **String** | Single domain |
| targets | OPTIONAL |
| **String** | Target objects, up to 5 targets can be defined |

dataset

|  |  |  |
| --- | --- | --- |
| **JSON** | An array containing the elements below | |
|  | title | REQUIRED |
|  | **String** |  |
|  | type | REQUIRED |
|  | **String** |  |
|  | user\_id | REQUIRED |
|  | **String** |  |
|  | screen\_name | REQUIRED |
|  | **String** |  |
|  | userjoindate | REQUIRED |
|  | **int** |  |
|  | postcount | REQUIRED |
|  | **String** |  |
|  | post\_time | REQUIRED |
|  | **Time** | In the format "YYYY-MM-DDThh:mm:ss.sTZD" |
|  | post\_id | REQUIRED |
|  | **String** |  |
|  | content | REQUIRED |
|  | **String** |  |
|  | reply\_to\_post\_id | REQUIRED |
|  | **String** |  |
|  | reply\_to\_uid | REQUIRED |
|  | **String** |  |
|  | msg\_from | REQUIRED |
|  | **String** |  |
|  | likes | REQUIRED |
|  | **int** |  |
|  | url | REQUIRED |
|  | **String** |  |
|  | source | REQUIRED |
|  | **String** |  |

###### Examples:

Input = [

{

"title": "-",

"type": "string",

"user\_id": "anon\_user",

"screen\_name": "anon\_user",

"userjoindate": 0,

"postcount": "string",

"post\_time": "2016-10-17T03:35:13.830Z",

"post\_id": "466382190189470\_606986016129086",

"content": "You served the nation with great dignity and commanded great love and respect from the people. Rest in peace, Sir",

"reply\_to\_post\_id": "-",

"reply\_to\_uid": "-",

"msg\_from": "anon\_user\_2",

"likes": 0,

"url": "https://www.facebook.com/masagos/posts/606986016129086",

"source": "facebook"

}

]

Without target object specification:

>>> service.analyze\_sentiment\_full\_dataset\_on\_the\_fly(dataset = input)

With target object specification:

>>> service.analyze\_sentiment\_full\_dataset\_on\_the\_fly(dataset = input, target = ‘sir’)

#### Response

This call returns the result of sentiment analysis on data. It returns the general sentiment attributed to a particular target object in JSON format.

###### Parameters

|  |  |
| --- | --- |
| results | Returns the domain name, target object and sentiment associated with it |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) |

###### Sample Response

{

"result": "[{"domain\_name":"general","objects":[{"sentiment":4, "target\_object":"defaultobject"}],"content":"You served the nation with great dignity and commanded great love and respect from the people. Rest in peace, Sir"}]",

"message": "",

"status": "1"

}

### Fine-grained Emotion Analysis

#### Function analyze\_sentimo\_on\_the\_fly (content, target, domain):

In this method, users can send request with only the target analysis content (string or array of string) to retrieve the fine-grained emotions results on the fly. Users can also indicate their targets or domain of interest in the request.

###### Parameters

|  |  |
| --- | --- |
| content | REQUIRED |
| **[String]/**  **String** | main content to be analyzed |
| domain | OPTIONAL |
| **String** | Multiple domains to be separated by comma |
| targets | OPTIONAL |
| **String** | Target objects, up to 5 targets can be defined |

###### Examples:

Without target object specification:

>>> service.analyze\_sentimo\_on\_the\_fly(content = ‘Love my new phone! Only that the battery runs out very fast’)

With target object specification:

>>> service.analyze\_sentimo\_on\_the\_fly(content = ‘Love my new phone! Only that the battery runs out very fast’, target = ‘sir’)

#### Function analyze\_sentimo\_full\_dataset\_on\_the\_fly (dataset, domain):

In this method, users can send request with full array dataset to retrieve the fine-grained emotions results on the fly. Users can also indicate their targets or domain of interest in the request.

###### Parameters

|  |  |
| --- | --- |
| domain | OPTIONAL |
| **String** | Multiple domains to be separated by comma |
| targets | OPTIONAL |
| **String** | Target objects, up to 5 targets can be defined |

dataset

|  |  |  |
| --- | --- | --- |
| **String** | An array containing the elements below | |
|  | title | REQUIRED |
|  | **String** |  |
|  | type | REQUIRED |
|  | **String** |  |
|  | user\_id | REQUIRED |
|  | **String** |  |
|  | screen\_name | REQUIRED |
|  | **String** |  |
|  | userjoindate | REQUIRED |
|  | **int** |  |
|  | postcount | REQUIRED |
|  | **String** |  |
|  | post\_time | REQUIRED |
|  | **Time** | In the format "YYYY-MM-DDThh:mm:ss.sTZD" |
|  | post\_id | REQUIRED |
|  | **String** |  |
|  | content | REQUIRED |
|  | **String** |  |
|  | reply\_to\_post\_id | REQUIRED |
|  | **String** |  |
|  | reply\_to\_uid | REQUIRED |
|  | **String** |  |
|  | msg\_from | REQUIRED |
|  | **String** |  |
|  | likes | REQUIRED |
|  | **int** |  |
|  | url | REQUIRED |
|  | **String** |  |
|  | source | REQUIRED |
|  | **String** |  |

###### Example:

input = ‘[

{

"title": "-",

"type": "string",

"user\_id": "anon\_user",

"screen\_name": "anon\_user",

"userjoindate": 0,

"postcount": "string",

"post\_time": "2016-10-17T03:35:13.830Z",

"post\_id": "466382190189470\_606986016129086",

"content": "You served the nation with great dignity and commanded great love and respect from the people. Rest in peace, Sir",

"reply\_to\_post\_id": "-",

"reply\_to\_uid": "-",

"msg\_from": "anon\_user\_2",

"likes": 0,

"url": "https://www.facebook.com/masagos/posts/606986016129086",

"source": "facebook"

}

]’

Without target object specification:

>>> service.analyze\_sentimo\_full\_dataset\_on\_the\_fly(dataset = input)

With target object specification:

>>> service.analyze\_sentimo\_full\_dataset\_on\_the\_fly(dataset = input, target = ‘sir’)

#### Response

This call returns the result of sentiment analysis on data. It returns the fine-grained emotions attributed to a particular target object in JSON format.

###### Parameters

|  |  |
| --- | --- |
| results | Returns the domain name, target object and sentiment associated with it |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful. If not, one of the error codes (1000,1001,1002,1003 or 1004) |

###### Sample Response

{

"result": "[{"domain\_name":"general","objects":[{"sentiment":4,"scores":[{"score":0,"category":"Negate"},{"score":1,"category":"Positive"},{"score":0,"category":"Negative"},{"score":0,"category":"Anxiety"},{"score":0,"category":"Anger"},{"score":0,"category":"Sadness"},{"score":1,"category":"Satisfaction"},{"score":0,"category":"Happiness"},{"score":0,"category":"Excitement"}],"target\_object":"sir"},{"sentiment":4,"scores":[{"score":0,"category":"Negate"},{"score":6,"category":"Positive"},{"score":0,"category":"Negative"},{"score":0,"category":"Anxiety"},{"score":0,"category":"Anger"},{"score":0,"category":"Sadness"},{"score":2,"category":"Satisfaction"},{"score":2,"category":"Happiness"},{"score":0,"category":"Excitement"}],"target\_object":"defaultobject"}],"content":"You served the nation with great dignity and commanded great love and respect from the people. Rest in peace, Sir"}]",

"message": "",

"status": "1"

}

## Module #4 class KnowledgeManagement

**This module is only avalible to the SDK licensees.**

This module allows for immediate analysis of data sent in the request body, returning results of the general sentiment analysis if the sentiment module is called, and the fine-grained emotion analysis if the sentimo module is called.

>>> #Initial Python class  
>>> from sentimo\_developer\_cloud.knowldge\_management import KnowldgeManagement  
>>> service = KnowledgeManagement (url = **TARGET\_URL**, token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| url | REQUIRED, class parameter |
| **String** | Used for service address communication |
| token | REQUIRED, class parameter |
| **String** | a valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used for how long the service response from servers. |

Add Dictionary  
Function add\_dictionary (dataset, domain):

##### Parameters

|  |  |
| --- | --- |
| domain | OPTIONAL |
| **String** | Function domain |

dataset

|  |  |  |
| --- | --- | --- |
| **JSON** | An array containing the elements below | |
|  | category\_code | REQUIRED |
|  | **String** | Possible values: Negate, Positive, Negative, Anxiety, Anger, Sadness, Satisfaction, Happiness, Excitement |
|  | sentimo\_word | REQUIRED |
|  | **String** |  |

###### Example:

input = [

{

"category\_code": "Negate",

"sentimo\_word": "string"

}

]

Without domain object specification:

>>> service.add\_dictionary(dataset = input)

With target object specification:

>>> service.add\_dictionary(dataset = input, domain = ‘education’)

#### Response

##### Parameters

|  |  |
| --- | --- |
| result | Returns the domain name, PID, dictionary category, dictionary category code and entry ID |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) |

Sample Response

{

"result":"[{\"sentimo\_word\":\"string\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170595}]",

"message": "",

"status": "1"

}

### Delete Dictionary

Function delete\_dictionary (dataIds, domain):

Deletes entries in dictionary by IDs.

##### Parameters

|  |  |
| --- | --- |
| Ids | REQUIRED |
| **Array[long]** | IDs of dictionary terms |

Example:

>>> service.delete\_dictionary(dataIds = [172345,174348])

Ids=170595

#### Response

##### Parameters

|  |  |
| --- | --- |
| result | Returns the ID of entry deleted |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) |

Sample Response

{

"result":"{\"sentimo\_word\":\"string\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170595}",

"message": "deleted dictionary for ids",

"status": "1"

}

### Get Dictionary Item

#### Function get\_dictionary\_item(dataId):

Retrieves information about a specific entry from its ID.

##### Parameters

|  |  |
| --- | --- |
| dataId | REQUIRED |
| **Long/String** | ID of dictionary item |

Example:

>>> service.get\_dictionary\_item(dataIds = 172345)

Ids=170595

#### Response

##### Parameters

|  |  |
| --- | --- |
| result | Returns domain name, partner ID, dictionary category, dictionary category code, and entry ID |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) |

Sample Response

{

"result":"{\"sentimo\_word\":\"string\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170595}"

"message": "",

"status": "1"

}

### Get Dictionary List

#### Function get\_dictionary\_list(domain):

Retrieves list of entries under specific domains.

##### Parameters

|  |  |
| --- | --- |
| domain | REQUIRED |
| **String** | Function domain |

Example:

>>> service.get\_dictionary\_item(domain = ‘education’)

Ids=170595

#### Response

##### Parameters

|  |  |
| --- | --- |
| result | Returns all entries in domain, with domain name, partner ID, dictionary category, dictionary category code, and entry ID for each entry |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) |

Sample Response

{

"result":"[

{\"sentimo\_word\":\"string1\",\"domain\_name\":\"general\",\"partner\_id\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170593},

{\"sentimo\_word\":\"string\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170595},

{\"sentimo\_word\":\"string2\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170591},

{\"sentimo\_word\":\"string3\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170589},

{\"sentimo\_word\":\"string4\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170588},

{\"sentimo\_word\":\"string5\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170592},

{\"sentimo\_word\":\"string6\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170590}

]",

"message": "",

"status": "1"

}

### Update Dictionary

#### Function update\_dictionary(dataset, domain):

Allows users to make changes to existing entry in dictionary.

##### Parameters

|  |  |
| --- | --- |
| domain | OPTIONAL |
| **String** | Function domain |

dataset

|  |  |  |
| --- | --- | --- |
| **JSON** | An array containing the elements below | |
|  | id | REQUIRED |
|  | **String**  category\_code |  |
|  |  | REQUIRED |
|  | **String** | Possible values: Negate, Positive, Negative, Anxiety, Anger, Sadness, Satisfaction, Happiness, Excitement |
|  | sentimo\_word | REQUIRED |
|  | **String** |  |

Example:

Input = [

{

"id": 170595,

"category\_code": "Negate",

"sentimo\_word": "string"

}

]

Without domain object specification:

>>> service.update\_dictionary(dataset = input)

With target object specification:

>>> service.update\_dictionary(dataset = input, domain = ‘education’)

#### Response

##### Parameters

|  |  |
| --- | --- |
| result | Returns the domain name, PID, dictionary category, dictionary category code and entry ID |
| **String** |  |
| message |  |
| **String** |  |
| status | Returns status of call |
| **String** | 1 if successful, if not one of the error codes (1000,1001,1002,1003 or 1004) |

Sample Response

{

"result":"[{\"sentimo\_word\":\"string\",\"domain\_name\":\"general\",\"partner\_id\":\"<PID>\",\"category\_name\":\"Negate\",\"category\_code\":18,\"id\":170595}]"

"message": "",

"status": "1"

}

# Section 3 Status/Error Codes

|  |  |
| --- | --- |
| Status | Description |
| 1 | Successful |
| 1000 | One or more input parameters not in valid format |
| 1001 | No data has been found for the given input parameters |
| 1002 | JSON error |
| 1003 | Database error |
| 1004 | Datetime parsing error |

# Section 4 Numeric Indicators in Sentiment Output

The main part of SentiMo's output is the "Sentiment". It has the following six values:

|  |  |  |
| --- | --- | --- |
| Status | Description | Meaning |
| 0 | Neutral | Neither positive nor negative |
| 1 | Mixed-Neutral | Contains both positive and negative sentiments with equal weightage of each |
| 2 | Negative | Contains only negative sentiments |
| 3 | Mixed-Negative | Contains both positive and negative sentiments, but with a stronger weightage of negative sentiments |
| 4 | Positive | Contains only positive sentiments |
| 5 | Mixed-Positive | Contains both positive and negative sentiments, but with a stronger weightage of positive sentiments |

For certain use cases, some users may want to convert SentiMo’s 6-class outputs to 4-class (quaternary) scheme. This convertion is straightforward:

Convert 0-Neutral as “Neutral”; 1-Mixed-Neutral, 3-Mixed-Negative, 5-Mixed-Positive as “Mixed”; 2-Negative as “Negative”; 4-Positive as “Positive”

Converting SentiMo’s 6-class outputs to 3-class (trinary) is also straightforward:

Convert 0-Neutral as “Neutral”; 1-Mixed-Neutral, 2-Negative, 3-Mixed-Negative as “Negative”; 4-Positive, 5-Mixed-Positive as “Positive”

Converting SentiMo’s 6-class outputs to 2-class (binary) is complex and not recommended. The main reason for this is that a 2-class sentiment classification (without any *neutral* utterance) does not commonly exist in any real-world context. One possible approximation of the conversion may look like:

Convert 0-Neutral, 1-Mixed-Neutral, 2-Negative, 3-Mixed-Negative as “Negative”; 4-Positive, 5-Mixed-Positive as “Positive”

# Section 5 Numeric Indicators in Emotion Output

The emotion output highlight whether the text message expresses or implies one or more of the six common emotion types or more generally, affective states: Satisfaction, Happiness, Excitement, Sadness, Anxiety and Anger. Note that a text message may convey two or more types of emotions, such as excitement and sadness, simultaneously.

Each of the six emotion outputs (Satisfaction, Happiness, Excitement, Sadness, Anxiety and Anger) has one of these two values:

|  |  |  |
| --- | --- | --- |
| Status | Description | Meaning |
| 0 | No | Does not express or imply this emotion |
| 1 or more | Yes | Expresses or implies this emotion, and the higher value indicates higher strength |

# Section 6 Other Outputs

SentiMo also produces three additional outputs. "Negate" indicates if there is a negation in the text message. "Positive" indicates if the text message expresses or implies a positive meaning, with higher values indicating higher strength. "Negative" indicates if the text message expresses or implies a negative meaning, with higher values indicating higher strength.

These outputs are meant to provide extra information about SentiMo's analysis logic. You may choose to ignore them and focus on the main outputs.

# Section 7 Quick Start Examples

If you have Python installed, you can extract the project folder and run the examples case on your Python terminal.

## Extracting the SDK folder

[sudo] unzip semtimo\_python\_sdk.zip

Before to extract the SDK folder, please create your own project folder. Depending on your system, you may need to use sudo at the beginning of command for Linux/Unix-based systems, if permission is denied.

## Updating the configurations

{

"url": "Target Service Address” ,

"token" : "Registered Token“,

"timeout": 5

}

After extracting the SDK folder, you may need to update **config.json** file under the SDK root folder before try to run the example scripts. Please update your target RESTful platform address under *url* key, and update your PID value under *token*. You can also update the *timeout* key based on your preference.

## Running the examples

python AnalyzeOnTheFly.py

Depends on your system and development environment, you can use terminal or other tools to run the example scripts in Python.

You should see the result in your terminal window or in the console of your specific Integrated Development Enviornment.