|  |
| --- |
|  |
| SENTIMO |
| Node.JS SDK Documentation |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | May 22, 2018 | Initial Draft |

Contents

[Section 1 SENTIMO Node.JS SDK Introduction 2](#_Toc515028508)

[Basics 2](#_Toc515028509)

[Evaluating responses 2](#_Toc515028510)

[Authentication 2](#_Toc515028511)

[Section 2 API References 3](#_Toc515028512)

[Module #1 class SecuredUpload 3](#_Toc515028513)

[Module #2 class GetSentiments 6](#_Toc515028514)

[*General Sentiment Analysis* 6](#_Toc515028515)

[Fine-grained Emotion Analysis 10](#_Toc515028516)

[Session #3 class AnalyzeOnTheFly 14](#_Toc515028517)

[General Sentiment Analysis 14](#_Toc515028518)

[Fine-grained Emotion Analysis 18](#_Toc515028519)

[Section 3 Status/Error Codes 22](#_Toc515028520)

[Section 4 Numeric Indicators in Sentiment Output 23](#_Toc515028521)

[Section 5 Numeric Indicators in Emotion Output 24](#_Toc515028522)

[Section 6 Other Outputs 25](#_Toc515028523)

[Section 7 Quick Start Examples 26](#_Toc515028524)

[Extracting the SDK folder 26](#_Toc515028525)

[Updating the configurations 26](#_Toc515028526)

[Running the examples 26](#_Toc515028527)

# Section 1 SENTIMO Node.JS SDK Introduction

Welcome to the SENTIMO Node.JS SDK Documentation.

The SENTIMO Node.JS 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 Node.JS SDK.

## Basics

The Node.JS 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.

Compulsory arguments are:

* + **hostname** : the host name of target server
  + **path** : the service path for of host server
  + **token** : a valid token (PID) to authenticate the end user.

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

## Evaluating responses

All Node.JS 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 class helps with uploading data to server.

#Initial Node.JS  
const SecuredUpload = require('../sentimo\_developer\_cloud/secured\_upload');  
const service = new SecuredUpload(hostname=**TARGET\_HOST**, path = **PATH,** token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| hostname | REQUIRED, class parameter |
| **String** | Host name of the API server |
| path | REQUIRED, class parameter |
| **String** | The target path of provided service |
| token | OPTIONAL, class parameter |
| **String** | A valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used to specify wait time for response |

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

###### Parameters

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

Example:

service.upload\_data(content = 'The MRT is very croweded today.', **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response)

}

});

#### 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"

}]

}’

service.upload\_data\_set(content = input, **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response)

}

});

#### 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 |

Success Response

A sample result of a successful request is as follows:

{

"status": 1

"data\_ids":[621292,621293]

}

## Module #2 class GetSentiments

This call 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.

#Initial Node.JS  
const GetSentiments = require('../sentimo\_developer\_cloud/get\_sentiments);  
const service = new Getsentiments(hostname=**TARGET\_HOST**, path = **PATH,** token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| hostname | REQUIRED, class parameter |
| **String** | Host name of the API server |
| path | REQUIRED, class parameter |
| **String** | The target path of provided service |
| token | OPTIONAL, class parameter |
| **String** | A valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used to specify wait time for response |

#### 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 |

###### Sample calls:

Without domain specification:

service.retrieve\_sentiment(dataIds = '621236,621237,621238', domain = '', **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

With domain specification:

service.retrieve\_sentiment(dataIds = '621236,621237,621238', domains = 'general,education' , **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

##### 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 |

##### Sample calls

Without domain specification:

service.retrieve\_sentiment\_set(dataId = '621236', domains = '', num = 10, **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

With domain specification:

service.retrieve\_sentiment\_set(dataId = '621236', domains = 'general,education', num = 10, **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

#### Response

This call 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:

service.retrieve\_sentimo(dataIds = '621236,621237,621238', domain = '', **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

With domain specification:

service.retrieve\_sentiment(dataIds = '621236,621237,621238', domains = 'general,education' , **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

#### 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:

service.retrieve\_sentimo\_set(dataId = '621236', domains = '', num = 10, **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

With domain specification:

service.retrieve\_sentimo\_set(dataId = '621236', domains = 'general,education', num = 10, **function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response);

}

});

#### 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"

}

## Session #3 class AnalyzeOnTheFly

This call 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 Node.JS  
const AnalyzeOnTheFly = require('../sentimo\_developer\_cloud/analyze\_on\_the\_fly);  
const service = new Getsentiments(hostname=**TARGET\_HOST**, path = **PATH,** token = **PID**, timeout = **30**)

###### Parameters

|  |  |
| --- | --- |
| hostname | REQUIRED, class parameter |
| **String** | Host name of the API server |
| path | REQUIRED, class parameter |
| **String** | The target path of provided service |
| token | OPTIONAL, class parameter |
| **String** | A valid token to authenticate the end user |
| timeout | OPTIONAL, class parameter |
| **Integer** | Used to specify wait time for response |

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

|  |  |
| --- | --- |
| content | REQUIRED |
| **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 targets and domain object specification:

service.analyze\_sentiment\_on\_the\_fly(content = 'Love my new phone! Only that the battery runs out very fast.','' ,'' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response)

}

});

Without target object specification:

With target object specification:

service.analyze\_sentiment\_on\_the\_fly(content = 'Love my new phone! Only that the battery runs out very fast.',targets = 'sir' ,'' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response)

}

});

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

###### Parameters

|  |  |
| --- | --- |
| pid | REQUIRED, header parameter |
| **String** | A valid token to authenticate the end user |
| domain | OPTIONAL |
| **String** | Multiple domains to be separated by comma |
| 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, targets = '',domain = '' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(res)

}

});

With target object specification:

service.analyze\_sentiment\_full\_dataset\_on\_the\_fly(dataset = input, targets = 'sir,phone',domain = 'transport' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(res)

}

});

#### Response

###### 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):

###### Parameters

|  |  |
| --- | --- |
| content | REQUIRED |
| **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 targets and domain object specification:

service.analyze\_sentimo\_on\_the\_fly(content = 'Love my new phone! Only that the battery runs out very fast.','' ,'' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response)

}

});

Without target object specification:

With target object specification:

service.analyze\_sentimo\_on\_the\_fly(content = 'Love my new phone! Only that the battery runs out very fast.',targets = 'sir' ,'' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(response)

}

});

#### 

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

###### 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** |  |

###### Sample input

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"

}

]’

###### Sample calls

Without target object specification:

service.analyze\_sentimo\_full\_dataset\_on\_the\_fly(dataset = input, targets = '',domain = '' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(res)

}

});

With target object specification:

service.analyze\_sentimo\_full\_dataset\_on\_the\_fly(dataset = input, targets = 'sir,phone',domain = 'transport' ,**function**(errors, response){

**if**(errors){

console.log(errors)

}**else**{

console.log(res)

}

});

#### Response

###### 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"

}

# 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 Node.JS plalform installed or your explorer support Node.JS and Javascript, you can extract the project folder and run the examples case on your Node.JS environment.

## Extracting the SDK folder

[sudo] unzip semtimo\_nodejs\_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

{

"hostname": "sentimoplus.net" ,

"path" : "/sentimo-webservice/rest"

"token" : "11111111-2222-3333-aaaa-ffffffff" ,

"timeout": 30

}

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

**Reminders**: please make sure the hostname should not contain **http** or **https** header.

## Running the examples

Depends on your system and development environment, you can use IDE tools or different explorers (ex. Google Chrome, FireFox) to run the example scripts in Node.JS.

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