

# **Technical Description of the Opinion Analysis web service**

## **Version 1.1**

## Content

1. Modification .....	3
Version 1.1 .....	3
2. Overview .....	3
3. Web-service access.....	3
3.1. SOAP access .....	3
3.2. REST access.....	4
4. Function description.....	4
Synapse_Analysis_Output GetOpinions(Synapse_Analysis_Input input); .....	4
5. Opinion XML output.....	5
6. Javascript example .....	6

---

## 1. Modification

### **Version 1.1**

- Unified credentials system.
- English integration.

### **Version 1.2**

- Unified apikey system.

## 2. Overview

The aim of this document is to present and explain the Opinion and Sentiment Analysis Web-service API. It provides a description of the web-service function, and a comprehensive example of consumption.

## 3. Web-service access

This web-service is available both through a SOAP and a REST access point. This section describes how to connect to one of these two access points.

### **3.1. SOAP access**

The web-service expose a WSDL at this address:

<http://api-synapse-dev.azurewebsites.net/sentiment/SentimentAnalysisWCF.svc?singleWsd1>

This WSDL can be used to automatically generate a proxy for your application.

#### **Manual configuration**

If you cannot use the WSDL, or if you prefer to build manually the request, here are the connection parameters for the service:

Service address:

<http://api-synapse-dev.azurewebsites.net/sentiment/SentimentAnalysisWCF.svc>

Service host:

<http://api-synapse-dev.azurewebsites.net/sentiment>

SOAPAction:

<http://api-synapse-dev.azurewebsites.net/sentiment/SentimentAnalysisWCF.svc/ISentimentAnalysisWCF/GetOpinions>

The web-service is configured to receive and send XML.

### 3.2. REST access

In order to provide a convenient way to call the web service from a JQuery, the function is also available at:

<http://api-synapse-dev.azurewebsites.net/sentiment/SentimentAnalysisWCF.svc/rest/GetOpinions>

The section 6 provides an example of how to call the web-service using JQuery and the REST access point.

## 4. Function description

***Synapse\_Analysis\_Output GetOpinions(Synapse\_Analysis\_Input input);***

### **Description:**

The function takes as input a structure containing the credentials identifying the client willing to consume the web-service, the language of the text, and the text to analyse.

The function returns another structure with the text augmented by the detected opinions (as XML annotations), and the possible exception and error codes raised by the service.

### **Argument:**

*input* is a structure containing the following elements:

- *apikey*: the apikey (as given by Synapse)
- *lang*: language of the text. Can take the values "fr" for French, and "en" for English.
- *text*: the text to be processed

XML example for input:

```
<Synapse_Analysis_Input>
  <apikey>apikey</apikey>
  <lang>fr</lang>
  <text>Le restaurant est bon.</text>
</Synapse_Analysis_Input>
```

### **Returns:**

*output* is a structure containing the following elements:

- *opinions*: input text enriched by XML annotations (format provided in section 5). Null/empty if an error occurred.
- *exceptions*: structure containing
  - o *errcode*: the error code. 0 if no error were raised.
  - o *message*: comprehensive message describing the problem if an error was raised (i.e. if the errcode is different from 0).

Typical XML output corresponding to the input example:

```
<Synapse_Analysis_Output>
  <opinions>
    <comment id="0" commentPolarity="pos" commentIntensity="2"><phr
id="phr_0_0_m"><Opinion id="opi_0_p2_0_m" type="jugement"
category="evaluation" pol="pos" intensity="2" length="22" lengthAscii="22"/>Le
restaurant est bon.</phr></comment>
  </opinions>
  <exceptions>
    <errcode>0</errcode>
    <message/>
  </exceptions>
</Synapse_Analysis_Output>
```

Error code values (errcode):

- 0: no error.
- 1: wrong API key.
- 2: expired API key, user does no longer have the right to consume web-service. Contact Synapse to renew your subscription.
- 3: unrecognized language
- 4: no text specified, or unrecognized text format.
- 5: internal error. If problem persist, contact support.

## 5. Opinion XML output

A typical XML output (field opinions in the output structure) is the following:

```
<comment id="0" commentPolarity="pos" commentIntensity="2">
  <phr id="phr_0_0_m">
    <Opinion id="opi_0_p2_0_m" type="jugement"
category="evaluation" pol="pos" intensity="2" length="22"
lengthAscii="22"/>Le restaurant est bon.
  </phr>
</comment>
```

The XML wrap the input text inside a <comment/> node. This node may have the following attributes:

- *id*: a generated id, usually "0".
- *commentPolarity*: overall polarity detected. Can have the following values:
  - o *pos*: positive polarity
  - o *neg*: negative polarity
  - o *neutral*: mixed polarity, or no opinion detected
- *commentIntensity* (optional): if a polarity is detected, gives on a 1-3 scale the intensity of the overall opinion

Opinions are also returned in text with the tags <Opinion/>. These tags may have the following attributes:

- *id*: a generated id. The id is unique inside the document. Several runs with same text will generate the same ids for the same opinion.
- *type*: opinion type
- *category*: opinion sub-type
- *pol*: polarity of the opinion
- *intensity*: intensity of the opinion
- *length*: length of the opinion in the text, in number of characters, starting from the end of the <Opinion/> tag.

An optional tag <phr> segments the text into sentences (French only).

## 6. Javascript example

This example shows how to make a simple javascript call to the web service using JQuery. JQuery library import is mandatory and not included in this sample.

```
var xmlInput = '<Synapse_Analysis_Input> <apikey>' + apikey +
'<\apikey><lang>' + lang + '<\lang> <text>' + text +
'<\text> </Synapse_Input_Analysis>';
var opinions = "";

$.ajax({
  type : "POST",
  url : "http://api-synapse-
dev.azurewebsites.net/sentiment/SentimentAnalysisWCF.svc/rest/
getopinions",
  data : xmlInput,
  dataType : 'xml',
  contentType : "text/xml; charset=utf-8",
  success : parseResponse,
  error : errorHandler
});

function parseResponse(output_analysis) {
  opinions = new XMLSerializer()
    .serializeToString( output_analysis );
}

function errorHandler(XHR, textStatus, errorThrown) {
  alert(errorThrown);
}
```