

# **Technical Description of the Keywords and Concepts Extraction 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.....	3
4. Function description.....	4
Synapse_Analysis_Output_KeyWordsConcepts (Synapse_Analysis_Input input); .....	4
5. KeyWordsAndConcepts XML output.....	6
6. Javascript example .....	7

---

## 1. Modification

### *Version 1.1*

- Unified apikey system.

## 2. Overview

The aim of this document is to present and explain the Keywords and Concepts Extraction 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/concepts/MotsClesWCF.svc?singleWSDL>

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/concepts/MotsClesWCF.svc>

Service host:

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

SOAPAction:

<http://api-synapse-dev.azurewebsites.net/concepts/IMotsClesWCF/GetKeyWordsAndConcepts>

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/concepts/MotsClesWCF.svc/rest/GetKeyWordsAndConcepts>

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\_KeyWordsConcepts**      **GetKeyWordsAndConcepts**  
(**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)
- *format*: (optional) desired output format. Default is xml (see description below).
- *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>
  <format>xml</format>
  <lang>fr</lang>
  <text>Le mot lapin est un terme très général qui désigne en français certains animaux lagomorphes à longues oreilles, que l'on différencie des lièvres par une silhouette moins élancée et par les petits qui naissent aveugles et nus, cachés dans un nid creusé au sol. Ces animaux ne correspondent donc pas à un niveau précis de classification scientifique. "Lapin" est en fait un nom vernaculaire ambigu, désignant une partie seulement des différentes espèces de mammifères classées dans la famille des Léporidés, famille qui regroupe à la fois les lièvres et les lapins. Longtemps classés dans l'ordre des rongeurs, ils le sont maintenant dans un ordre à part entière : les Lagomorphes.</text>
</Synapse_Analysis_Input>
```

### **Returns:**

*output* is a structure containing the following elements:

- *keyWordsAndConcepts*: XML description of keywords and concepts (format provided in section 5). Can be null/empty if an error occurred.
- *exceptions*: structure containing
  - o *errcode*: the error code. 0 if no error were raised.

- *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 (some extra concept and keyword nodes pruned for lisibility):

```

<Synapse_Analysis_Output_KeyWordsConcepts>
  <keyWordsAndConcepts><kw_c>
    <concepts3>
      <concept3 code="243" freq="1.00">mammifères</concept3>
      <concept3 code="242" freq="0.43">zoologie</concept3>
      <concept3 code="10" freq="0.30">classification</concept3>
      [...]
    </concepts3>
    <concepts4>
      <concept4 code="243.2" freq="1.00">rongeurs</concept4>
      <concept4 code="243.1" freq="0.72">anatomie des
mammifères</concept4>
      <concept4 code="34.5" freq="0.40">narcissisme</concept4>
      <concept4 code="242.1" freq="0.36">zoologie</concept4>
      [...]
    </concepts4>
    <keywords>
      <keyword gram="N" freq="0.33">lapin</keyword>
      <keyword gram="N" freq="0.28">rongeur</keyword>
      <keyword gram="N" freq="0.22">nom vernaculaire</keyword>
      [...]
    </keywords></kw_c>
  </keyWordsAndConcepts>
  <exceptions>
    <errcode>0</errcode>
    <message/>
  </exceptions>
</Synapse_Analysis_Output>

```

Error code values (errcode):

- 0: no error.
- 1: wrong credentials.
- 2: expired credentials, user does no longer have the right to consume web-service. Contact Synapse to renew your subscription.
- 3: unrecognized language – analysis returned for the default language.
- 4: no text specified, or unrecognized text format.
- 5: internal error. If problem persist, contact support.
- 6 unrecognized output format – analysis returned in default format.

## 5. KeyWordsAndConcepts XML output

This section details the XML output. The concepts and Synapse taxonomy are further detailed in another document which can be provided on demand.

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

```
<kw_c>
  <concepts3>
    <concept3 code="243" freq="1.00">mammifères</concept3>
    <concept3 code="242" freq="0.43">zoologie</concept3>
    <concept3 code="10" freq="0.30">classification</concept3>
    [...]
  </concepts3>
  <concepts4>
    <concept4 code="243.2" freq="1.00">rongeurs</concept4>
    <concept4 code="243.1" freq="0.72">anatomie des
mammifères</concept4>
    <concept4 code="34.5" freq="0.40">narcissisme</concept4>
    <concept4 code="242.1" freq="0.36">zoologie</concept4>
    [...]
  </concepts4>
  <keywords>
    <keyword gram="N" freq="0.33">lapin</keyword>
    <keyword gram="N" freq="0.28">rongeur</keyword>
    <keyword gram="N" freq="0.22">nom vernaculaire</keyword>
    [...]
  </keywords>
</kw_c>
```

Under the <kw\_c/> root node, the output contains three sections:

- <concepts3/> : contains the high level concepts extracted from the text. Each concept is presented in a <concept3/> tag containing:
  - o An attribute *code* identifying the concept (an integer, for instance 243 in the first line of example above)
  - o An attribute *freq* with the relative frequency score of the concept, compared to the other concepts (float value between 0 and 1, 1.0)
  - o The label of the concept category.
- <concepts4/> contains the finer level concepts extracted. As for the high level concepts, each <concept4/> tag contains :
  - o An identifier code (two integer separated by a dot, e.g. 243.2 for the first concept4 of the example above)
  - o An attribute *freq* with the relative frequency score of the concept
  - o The label of the category.
- <keywords> contains the extracted keywords form the text. Each keyword, wrapped in a <keyword/> tag, has the following attributes:
  - o *gram* indicated the grammatical category of the keyword (N for noun, A for adjective, V for verb).
  - o *freq* is the frequency score of the keyword.

## 6. Javascript example

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

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

var keywords = "";

$.ajax({
    type : "POST",
    url : "http://api-synapse-
dev.azurewebsites.net/keywords_and_concepts/MotsClesWCF.svc/re
st/GetKeyWordsAndConcepts",
    data : xmlInput,
    dataType : 'xml',
    contentType : "text/xml",
    success : parseResponse,
    error : errorHandler
});

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

    $("#outputTxt").setValue(keywords);
}

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