# INT LEXICONSERVICE - MANUAL

Author: Mathieu Fannee, INT – April 2021

## CONTENT

## INTRODUCTION

The INT LexiconService is a webservice that gives any piece of software quick online access to a lexicon by means of http requests. The LexiconService is designed to access a computational lexicon with an Impact Lexicon Database structure.[1] The service is now deployed at INT and gives access to INT's GiGaNT lexicon of 13th to 20th century Dutch.

This webservice offers various possibilities. One can obtain the word forms belonging to a given lemma, or the other way round, one can get the lemma corresponding to a given word form. It is also possible to expand any word with its complete paradigm. And one can limit the results to a given period of history, or to a given part-of-speech. Finally, the lexical information provided by the webservice can be given in both XML or JSON format.

In the following, we'll be describing how requests need to be formulated to get the information you need.

## QUERY BASICS

To be able to get lexical information from the LexiconService, you need to provide it with at least three things:

- A word
- A lexicon to look up this word in
- And what you need to get in return (a lemma, a paradigm, …)

Let's start with the lexicon in which the word should be looked up. Let's say we want to access some Dutch lexicon, which is named '*lexicon_service_db*'.[2] Your request will have to contains this part:

```
...database=lexicon_service_db...
```

Then you have to tell the LexiconService what you need to get. Several operations are possible, like:

- Get the lemma of a word form
- Get the word forms of a lemma (=its paradigm)
- Expand a word to its complete paradigm (=lemma and all word forms)
- Get related lemmata (eg. synonyms)

Telling the LexiconService which word your search is about can be done in different ways, depending on the operation that needs to be performed. So we're going to describe the possible operations now.

[1] The current Dutch lexicon consists of a lemmata table with grammatical categories (part of speech), a wordforms table, an attestations table, a documents table with metadata (source information and date).

[2] The list of available lexica depends on the specific LexiconService instance you're using and also on the period of time (as new lexica might be added and old ones removed). Please contact the administrator of the LexiconService instance you are using to get an up-to-date list of the available lexica.

## GET A LEMMA

Telling the LexiconService to get the lemma of a word form is done by simply telling ‘get_lemma’. The first part of your http requests will therefore look like this:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?...
```

> BEWARE*: lexiconservice.inl.nl* (mentioned here above) is just a sample hostname, so this host might not be available anymore as you get to read this. Check the document about the currently available lexica (or contact the INT) to get the URL at which the service can currently be reached*.*

You then need to set a few parameters: the word form you want the lemma from, and the name of the lexicon where to look it up. Let’s say your word form is *liep* (the Dutch for *walked*). Your complete request will look like this:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep
```

Provided that you want JSON output, the LexiconService will send a response like this:

```
{
"message":"OK",
"lemmata_list":
    [{"query_word":"liep",
     "found_lemmata":
         [{"lemma":"lijp","pos":"ADJ ADV", …},
          {"lemma":"lopen","pos":"VRB", …}]
    }]
}
```

> *NB: the LexiconService can perform the same search with the more explicit verb ‘get_lemma_from_wordform’. But since ‘get_lemma’ and ‘get_lemma_from_wordform’ are full synonyms of each other, using the one or the other will make no difference to the results. Which one to use is up to you.*

## GET WORDFORMS

Telling the LexiconService to get the word forms of a lemma is done by simply telling ‘get_wordforms’. The first part of your http requests will therefore look like this:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?...
```

You then need to set a few parameters: the lemma you want the word forms from, and the name of the lexicon where to look it up. Let’s say your lemma is *lopen* (the Dutch for *to walk*). Your complete request will look like this:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?
database=lexicon_service_db&lemma=lopen
```

Provided that you want JSON output, the LexiconService will send a response like this:

```
{
"message":"OK",
"wordforms_list":
    [{"query_word":"lopen",
     "found_wordforms":["loop","loope","lopen", …]
    }]
}
```

*NB: just like 'get_lemma', the verb 'get wordforms' has a full synonym called 'get_wordforms_from_lemma' (which is not as short, but of course more explicit). Once again, which verb to use is up to you. Both mean the same and will yield the same results*

## EXPAND A WORDFORM OR LEMMA

Telling the LexiconService to expand a given word form or lemma to its complete paradigm is done by using the verb 'expand'. The first part of your http requests will therefore look like this:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/expand?...
```

You then need to set a few parameters: the word form you want to expand, and the name of the lexicon where to look it up. Let's say your word form is *loopt* (the Dutch for *(he) walks*). Your complete request will look like this:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/expand?
database=lexicon_service_db&wordform=loopt
```

Provided that you want JSON output, the LexiconService will send a response like this:

```
{
"message":"OK",
"wordforms_list":
    [{"query_word":"loopt",
      "found_wordforms":["eloopen","geloopen","gelopen","gheloopen",
        "ghelooppen","laupe","liep","liepen","liept","loepen","loop","loope",
        "loopen","loopend","loopende","loopenden","loopens","looppen","looppene",
        "loopt","lopen","lopende"]
    }]
}
```

BEWARE*:* the LexiconService does not only expand a word with its complete paradigm but also with its lemma. In some cases, this behaviour might be inappropriate, for example because the paradigm consists of historical words whereas the lemma is a modern form. In such cases, you might want to exclude the lemma from the expansion results. To do so, just add the 'exclude_lemma' parameter to the request:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/expand?
database=lexicon_service_db&wordform=loopt& exclude_lemma=true
```

## FIND RELATED LEMMATA

If a lexicon describes (semantic) relationships between lemmata, the LexiconService can retrieve those too. However, unlike the previous query's, searching for relationships between lemmata can't be done in one single query. You will need two query's instead.

The first query will be about getting the ID of the lemma your search is about. So, let's say you'd like to find lemmata having relationships with the lemma *lopen* (Dutch for *to walk*), you will first need to use this query:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma_id_from_lemma?
database=lexicon_service_db&lemma=lopen
```

Provided that you want JSON output, the LexiconService will send a response like this:

```
{
"message":"OK",
"lemmata_list":
    [{"query_word":"lopen",
     "found_lemmata":
         [{"lemma":"lopen","lemma_id":"14162", …}]
    }]
}
```

As requested, the response contains the ID of the lemma *lopen*. This ID can now be used to retrieve relationships with other lemmata. To achieve that, use the verb 'get_related_lemmata' and add the lemma ID as a parameter:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_related_lemmata?
database=lexicon_service_db&lemma_id=14162
```

The LexiconService will send a response like the following. The response will contain the lemmata having some (semantic) relationship with our query lemma, their IDs, and also the type of relationship:

```
{
"message":"OK",
"query_lemma_id":"551"
"found_lemmata_and_relations":
    [{"relation":”synonym”,
      "lemma_id":"23456",
      "lemma":"gaan",
      …},
     {"relation":”synonym”,
      "lemma_id":"34567",
      "lemma":"functioneren",
      …}
    ]
}
```

At last: if you want to, it is possible to limit your search to a specific type of relationships. For example, if you just need to know if a given lemma has synonyms (and you're not interested in other relationships), set the 'relation' parameter accordingly:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/ get_related_lemmata?
database=lexicon_service_db&lemma_id=12345&relation=synonym
```

In that case, the response will contain only synonyms and no other types of relations.

# ADVANCED QUERIES

## LIMIT THE SEARCH TO SOME DATASET

Some lexica consist of distinct parts, which you might want to query separately. For example, a lexicon of Dutch might mostly consist of every day Dutch, but might also have a set of scientific (medical, etc.) Dutch lemmata as well. If you want to query the 'medical Dutch' part only, you can specify that by setting the 'dataset' parameter:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep&dataset=medical
```

*NB: when available, the datasets of a lexicon are named in the document about the currently available lexica (provided by the INT) .*

## LIMIT THE SEARCH TO SOME PROVENANCE

The lemmata and word forms of a lexicon might have a provenance specification. This feature describes the origin of the data.

For example: some lemmata of a lexicon might have been gathered by an automatic script without any quality control, whereas other lemmata of the same lexicon might have been added manually by very conscientious lexicographers. As a consequence, the first set of lemmata will have the provenance feature set to 'unchecked', and the second to 'validated'.

If you want to limit your search to lemmata of a given provenance, that can be specified by adding the 'lemma_provenance' parameter:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/… …&lemma_provenance=validated
```

The same can be done for the paradigm (word forms) of a lemma, with the 'provenance' parameter:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/… …&paradigm_provenance=validated
```

*NB: when available, the provenances specified in a lexicon are named in the document about the currently available lexica (provided by the INT) .*

## LIMIT THE SEARCH TO SOME PART-OF-SPEECH

We've seen before how to get the lemma of a given word form. The needed query for the Dutch word form *liep* was:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep
```

The result consisted of two very different lemmata, a verb and an adjective/adverb:

```
{
"message":"OK",
"lemmata_list":
    [{"query_word":"liep",
     "found_lemmata":
         [{"lemma":"lijp","pos":"ADJ ADV"},
          {"lemma":"lopen","pos":"VRB"}]
    }]
}
```

Now imagine you're not interested in adjectives (ADJ), but only in verbs (VRB). You can set an extra parameter 'pos' (part-of-speech) to say just that:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep&pos=VRB
```

*NB: you'll find out about the correct **parts-of-speech notations** for your queries in the section 'Parts-of-speech notation'.*

Now the LexiconService output will be:

```
{
"message":"OK",
"lemmata_list":
    [{"query_word":"liep",
      "query_pos":"VRB",
      "found_lemmata":
          [{"lemma":"lopen","pos":"VRB"}]
    }]
}
```

Exactly the same can be achieved for the other operation 'get_wordforms'.

The request for word forms of lemma *lopen* limited to a part-of-speech *VRB* will look like:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?
database=lexicon_service_db&lemma=lopen&pos=VRB
```

Finally, if you want to filter on multiple parts-of-speech, add them separated by a pipe ('|'):

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?
database=lexicon_service_db&lemma=lopen&pos=VRB|ADJ|NOU
```

## PARTS-OF-SPEECH NOTATION

Using parts-of-speech in your queries implies that you know what the parts-of-speech look like in the lexicon you're querying.

There is an easy way to get an overview of the parts-of-speech in use. Use this request:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_parts_of_speech?
database=lexicon_service_db
```

The LexiconService output will show what parts-of-speech notation is like in the lexicon:

```
{
"message":"OK",
" part_of_speech":
    ["ADJ", "ADV", "ART", "CON", "NOU", "NUM", "PART", …]
}
```

These are the parts-of-speech you can use in the 'pos' parameter of your 'get_wordforms' or 'get_lemma' queries (see '*Limit the search to some part-of-speech*' section).

## LIMIT THE SEARCH TO A PERIOD OF TIME

The LexiconService offers the possibility to limit a search to a given period of time. This can be achieved by adding two parameters, 'year_from' and 'year_to'. It is possible to use only one of them, of both at the same time.

Let's say you'd like to get word forms of the word *lopen* before the year 1700, your request will look like:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?
database=lexicon_service_db&lemma=lopen&year_to=1700
```

This will give quite some oldish word forms:

```
… "found_wordforms":[ "gheloopen","ghelooppen",… ]} …
```

To get the modern word forms (after 1900) of *lopen* instead, we can use 'year_from':

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?
database=lexicon_service_db&lemma=lopen&year_from=1900
```

With more modern forms as a result:

```
… "found_wordforms":[ "geloopen","liep","loopen","loopend","loopende","loopt" ]} …
```

Of course, the 'year_from' and 'year_to' parameters can be used together so as to isolate a given period of time. Say we want the paradigm of *lopen* in the period 1600-1700, our request will be:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms?
database=lexicon_service_db&lemma=lopen&year_from=1600&year_to=1700
```

## QUERYING MORE WORDS AT ONCE

In the sections hereabove we explored the possibilities for getting paradigm information and such for one single word only. But the LexiconService can process lists of words as well.

Sending a query for list of words is easy, just separate the different words by comma's:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep,werk,dacht
```

The result will be expectably:

```
{
"message":"OK",
"lemmata_list":
    [{"query_word":"liep",
      "query_pos":"",
      "found_lemmata":
          [{"lemma":"lopen","pos":"VRB"}]
     },
     {"query_word":"werk",
      "query_pos":"",
      "found_lemmata":
          [{"lemma":"werk","pos":"NOU"}]
     },
     {"query_word":"dacht",
      "query_pos":"",
      "found_lemmata":
          [{"lemma":"denken","pos":"VRB"}]
     }]
}
```

As we saw before, when writing a query about a word, it is possible to specify its part of speech. When dealing with lists of words, the parts of speech need to be comma separated as well.

So, querying about the words *liep* ('walked'), *man* ('husband'), *aardig* ('nice', 'sweet'), the parts of speech information being respectively VRB (verb), NOU (noun) and ADJ (adjective), the resulting query will be:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep,man,aardig&pos=VRB,NOU,ADJ
```

Make sure, when adding parts of speech information, that each word form or lemma is provided an own part of speech. That is: the number of word forms or lemmata must equal the number of parts of speech. Otherwise the LexiconService won't be able to match the lemmata with the parts of speech information, and you will get an error.

Remember the comma is used to separate the parts-of-speech of each lemma. If you need to filter on more parts-of-speech for one of the lemma, use the pipe ('|') separator. In the following query, we filter lemma 'man' on parts-of-speech NOU and VRB:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep,man,aardig&pos=VRB,NOU|VRB,ADJ
```

## CASE SENSITIVITY

By default, the LexiconService is case sensitive. So querying the word *liep* or the word *Liep* (same word, but front letter in uppercase) might give different results.

If you want the LexiconService to be case *insensitive* instead, we can require just that by adding one parameter: 'case_sensitive=true/false'. Like that:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep&case_sensitive=false
```

## QUERYING WITH IDS INSTEAD OF LEMMATA

The ‘*Find related lemmata*’ section introduced a particular type of query, in which a lemma ID had to be provided as a parameter, instead of the more obvious and also more common type of parameter: a word. Although lemma IDs are mostly not needed to formulate query’s, they are mostly given as part of the output, and for sake of disambiguation (think of homonyms!), it is possible to formulate some query’s with IDs Here are some examples.

The first example is the query that was introduced in the ‘*Find related lemmata*’ section: getting the ID of a lemma.

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma_id_from_lemma?
database=lexicon_service_db&lemma=lopen
```

The opposite query (t.i. getting a lemma given its ID) is possible too:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma_from_lemma_id?
database=lexicon_service_db&lemma_id=14162
```

As expected, the LexiconService will send a response like this:

```
{
"message":"OK",
"lemmata_list":
    [{"query_lemma_id":"14162",
     "found_lemmata":[{"lemma":"lopen","lemma_id":"14162","pos":"VRB", …}]
    }]
}
```

At last, one can get the paradigm of a lemma, given its ID:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_wordforms_from_lemma_id?
database=lexicon_service_db&lemma_id=14162
```

Provided that you want JSON output, the LexiconService will send a response like this:

```
{
"message":"OK",
"wordforms_list":
    [{"query_lemma_id":"14162",
     "found_wordforms":["loop","loope","lopen", …]
    }]
}
```

For the moment, that’s all the LexiconService can do with lemma IDs. But once again: most of the time, IDs are not needed.

## ERROR INFORMATION

When the LexiconService needs to send out an error message, it puts this message into the JSON of XML response.

For example:

```
{
 "message":"ERROR: here comes your error message.",
 "lemmata_list":[]
}
```

The error messages of the LexiconService are designed to be as self-explanatory as possible. Error messages clearly tell which part of the input is missing or illegal, so finding the right way to solve a problem is mostly straightforward.
If not, don't hesitate to contact the administrator of the LexiconService instance you are using.

## PREVENT CACHING

As part of their optimization strategies, some servers might cache requests and responses, in such a way that they can reply faster and with less CPU use when receiving a request they had to process before. A bad thing about this is that if you're working with a growing lexicon, you won't be able to get newly added information for a word you already send a request about.

This can be solved easily, just by adding a 'dummy' parameter with some random number to the http request, in such a way that the request will always look different from requests sent before, even if those were about the same word. For example:

```
https://lexiconservice.inl.nl/LexiconService/lexicon/get_lemma?
database=lexicon_service_db&wordform=liep&dummy=1384187319550
```

## OUTPUT TYPE

The LexiconService can give both XML and JSON output. The output type cannot be set by an explicit parameter of the http request: you have to set it in the AJAX call of the application you're using to connect to the LexiconService.

## UPGRADE FROM PREVIOUS VERSION

If you’ve been using an old version of the LexiconService (that is: **installed before august 2014**), you’ll notice the output of the service has changed a bit. So, if you’ve implemented the LexiconService within your own software, you’ll need to slightly adapt the part of your code which parses the LexiconService output.

In the old version, the output consisted of a *lemma* object or a *wordform* object. Those objects could only contain one set of results, about only one query word. Since the new versions of the LexiconService can process multiple query words at once (all sent within one single request, see section ‘*Querying more words at once*’), the output now contains a list of results instead of a single result: one set per query word.
So, the *lemma* and *wordform* objects are now embedded within a *lemmata_list* or a *wordforms_list* object respectively. Those lists contain the original *lemma* and *wordform* objects, which are now renamed *found_lemmata* and *found_wordforms*. They were renamed that way to distinguish them from the queried wordforms, which are stored in *query word*.

In the following table, the output differences between the old version (till august 2014) and the current version of the LexiconService are summed up, so you can see in a glance what has changed.

| Task | Output of old version<br>(before august 2014) | Output of current version |
|---|---|---|
| **Get lemma** | {<br>"lemmata":[<br>{"lemma":"lopen","pos":"VRB"},<br>{"lemma":"lijp","pos":"ADJ"}<br>]<br>} | {<br>"message":"OK",<br>"lemmata_list":[<br>{<br>"query_word":"liep",<br>"query_pos":"",<br>"found_lemmata":[<br>{"lemma":"lopen","pos":"VRB", …},<br>{"lemma":"lijp","pos":"ADJ", …}<br>]<br>},<br>{<br>"query_word":"hoorde",<br>…<br>}<br>]} |
| **Get wordforms** | {<br>"wordform":[<br>"loop","loope","lopen", …<br>]<br>} | {<br>"message":"OK",<br>"wordforms_list":[<br>{<br>"query_lemma_id":"",<br>"query_word":"lopen",<br>"query_pos":"",<br>"found_wordforms":[<br>"loop","loope","lopen", …<br>]<br>},<br>{<br>"query_word":"horen",<br>…<br>}<br>]} |
| **Expand a word or lemma** | {<br>"wordform":[<br>"geloopen","gelopen", …<br>]<br>} | {<br>"message":"OK",<br>"wordforms_list":[<br>{<br>"query_lemma_id":"",<br>"query_word":"loopt",<br>"query_pos":"",<br>"found_wordforms":[<br>"geloopen","gelopen", …<br>]<br>},<br>{<br>"query_word":"hoorde",<br>…<br>}<br>]} |