

Advanced Intouch Search Service

Introduction

This document describes various ways to perform machine-to-machine queries to Link Mobility's database. No single protocol would satisfy all users, so there is a choice to be made on the part of the user between different transport protocols, authentication methods and result formats. This way, we can accommodate anything from a quick-and-dirty visual basic subroutine to a dedicated application for directory assistance operators. These are the available options, and will be discussed in the following sections:

TIPS AND TRICKS

Before getting started it could be smart to read thru some tips and tricks that highlights the most common issues done by those not reading the documentation properly...

- **I want to use the SOAP/WSDL transport protocol**
 - For .NET remember to set "style=document"
 - Using username/password instead of IP whitelisting, add username and password when downloading the WSDL
`http://live.intouch.no/tk/webservice.php?wsdl&style=document&username=XXX&password=YYY`
- **I want to perform a search and only get exact matches**
 - Add the parameter 'maxpass=1'
`http://live.intouch.no/tk/search.php?qry=22994400&maxpass=1&format=json` (Exact hit)
`http://live.intouch.no/tk/search.php?qry=229944001&maxpass=1&format=json` (Not exact, no hit)
- **I want to perform a number search for NO, SE and DK.**

If you only want to perform a number search, we have implemented some logic so you can search for NO, SE and DK for private and business in one and same query.

 - For Norway you don't need any area code, for SE '0046' and DK '0045'
`http://live.intouch.no/tk/search.php?qry=004533272727&maxpass=1&format=json`
- **I want to search for persons and businesses in SE/DK**

Read info under "Foreign search"
- **Other...**
 - **Read the documentation properly!**
 - **Don't find any help in the documentation?**
Ask: support@linkmobility.com for help

Transport protocols

- HTTP
 - HTTP GET
 - HTTP POST
- HTTPS
 - HTTPS GET
 - HTTPS POST
- SOAP/WSDL
 - HTTP/RPC/literal
 - ISO 8859-1
 - UTF-8
 - HTTP/document/literal
 - ISO 8859-1
 - UTF-8

Authentication methods

- IP white list
- Username/password

Result formats

- Export formats
 - vcard
 - bigvcard
 - xml_vcard
 - xml_rdf
- Internal structure formats
 - xml
 - xml_topdown
 - xmlsms
 - json / json2

Getting started

Before introducing all the options, let us look at two examples:

In a browser, open the following url:

<http://live.intouch.no/tk/search.php?qry=ola+nordmann+fagernes&format=vcard>

If your computer's IP address is on the white list, this should output:

```
BEGIN:VCARD
VERSION:2.1
N:Nordmann;Ola;;;
ADR;HOME;;;Fagernes;;2900;
TEL;CELL:954 93 312
END:VCARD
```

PHP is a freely available scripting language which supports SOAP. This complete PHP program uses SOAP/RPC/literal, IP white list authentication and vcard format:

```
<?
$Telefonkatalogen = new SoapClient('http://live.intouch.no/tk/webservice.php?wsdl');
echo $Telefonkatalogen->search('qry=ola+nordmann+fagernes&format=vcard');
?>
```

Output:

```
BEGIN:VCARD
VERSION:2.1
N:Nordmann;Ola;;;
ADR;HOME;;;Fagernes;;2900;
TEL;CELL:954 93 312
END:VCARD
```

Choosing transport protocol

All available protocols use HTTP, and the most efficient method is to use HTTP directly. Most programming environments support means of downloading data via HTTP. This is a widely used protocol which does not change frequently and which is consistently implemented across platforms and programming environments. This is the recommended transport protocol.

There are programming environments where it is more convenient to use SOAP to encapsulate the requests in XML, and this is also an option. However, it introduces no new functionality over HTTP and in many cases causes trouble due to the wide range of options, subsets, profiles, styles, methods and flavors of SOAP in different programming environments. The main use of SOAP in this context is for integration with Visual Basic for Microsoft Office applications.

Authentication

If the service is accessed without or with improper authentication, the service returns a web page requesting temporary registration instead of a proper SOAP response. There are several alternative authentication schemes, of which two are recommended for new projects:

- IP white list
- Username/password

IP white list

For server-to-server applications where your server has a fixed IP address, you may register that address permanently with your Link Mobility account using the admin portal at <https://admin.intouch.no> IP addresses can also be registered temporarily at the “access denied” page which is returned upon failed authentication. However, temporary registration must be accompanied by cookies, which is usually impractical in a SOAP context. Cookies are not necessary for permanently registered IP addresses.

Username / password

If the client IP address is unknown or prone to change, you may supply your Link Mobility username and password with each request. The function call does not have username/password parameters. Instead, these parameters must be piggy-backed on the http request which encapsulates the soap request. This is done for you if you supply a username and password when downloading the wsdl file. For example, if you download the wsdl file through this address:

`http://live.intouch.no/tk/websvcsendsms.php?wsdl&username=per&password=xo`

Then each soap request will automatically be piggy-backed with the username “per” and password “xo”. If you don’t know the username and password at the time the wsdl file is downloaded, you may add them to the url which the web service uses at run-time.

Choosing a result format

The result formats are grouped in three categories, with different applications. If you plan to import data in a structured way, you should use an *export format*. These are based on the vcard standard. The **vcard** format produces a minimal vCard suitable for mobile applications. The **bigvcard** format also conforms to the vCard standard, and contains more information. For the convenience of those who prefer XML, the **xml_rdf** format wraps the **bigvcard** format in XML according to a proposed open standard. This format is somewhat bloated, though, and a simpler version is available as **xml_vcard**.

If you need access to Link Mobility's internal data structure, use one of the *internal structure formats*. While these formats contain all available information, they also expose unnecessary complications and large data sets to most users, related to the way listings are organized internally in Link Mobility's systems.

Number series

Companies with switchboards typically reserve a series of telephone numbers. Only a few of these are exposed to users searching for the company name. However, Link Mobility collects information about such series, which is used in number-to-name lookups. This information is represented by the fields **tfnr_start**, **tfnr_stopp** and **prioritet**. Such listings may also contain a **tfnr** field, which should contain the company's main telephone number. The numerical **prioritet** field indicates the relative importance of this main telephone number, where a lower number indicates higher importance. Note that Link Mobility's search application automatically looks up any public directory listings containing the switchboard number indicated in the number series, in order to present more reliable and exhaustive information about the subscriber. Such matches are listed before the actual number series match.

Sublisting matches

A search may match a main listing (for example "Oslo Kommune Vann- og Avløpsetaten ") or a sublisting ("Oslo Kommune Vann- og Avløpsetaten Døgnvakt"). There are two ways to represent this in the internal structure formats:

- The **xml** format will root the result at the matched sublisting. The user may specify whether to populate the **parent** and **children** elements. The **parent** elements will then recursively contain listings out to the main listing, while the **children** elements contain sublistings in a similar, recursive manner inwards. Only the matched sublisting may have *both* **parent** and **children** populated.
- The **xml_topdown** format will root the result at the main listing, provided that the user chooses to load parents of matched listings. Only **children** elements are populated in this format, and no **parent** elements. This may be slightly easier to parse. The disadvantage is that there may be no way to tell which level of sublistings was actually matched.

Transferring child and parent listings to the client is optional (see **loadparents** and **loadchildren** options in the search section). The internal structure formats contain **id** numbers which can be used to retrieve these listings in subsequent requests. If parents or children are not loaded, the **children** and **parent** elements are empty, while **childrenid** and **parentid** elements are populated.

Transport protocols

Web service

The web service is accessed through the wsdl file, which contains the schemas and pointers needed for further requests. The format of the requests and replies are described elsewhere in this document. The wsdl url can be appended by the following parameters:

Parameter	Use	Values	Default
style	Specify the SOAP style. The document style is required for current versions of .NET	document or rpc	rpc
charset	Specify character set in queries and replies	ISO-8859-1 or UTF-8	UTF-8

Example:

<http://live.intouch.no/tk/webservice.php?wsdl&style=document>

The web service contains the following functions:

Function	Use	Parameters	Result format
search	Perform search queries	Text string qry , which contains a variable number of url-encoded parameters, see the search section.	Any result format, including internal structure formats, encapsulated in a text string
lookup	Perform lookup queries	Text string qry , which contains a variable number of url-encoded parameters, see the lookup section	Any result format, including internal structure formats, encapsulated in a text string
searchxml	Perform search queries	Text string qry , which contains a variable number of url-encoded parameters, see the search section.	Any internal structure format, exposed as XML
lookupxml	Perform lookup queries	Text string qry , which contains a variable number of url-encoded parameters, see the search section.	Any internal structure format, exposed as XML
searchxmlVCard	Perform search queries	Text string qry , which contains a variable number of url-encoded parameters, see the search section.	The format parameter must be set to <code>xml_vcard</code>
lookupxmlVCard	Perform lookup queries	Text string qry , which contains a variable number of url-encoded	The format parameter must be set to <code>xml_vcard</code>

	parameters, see the lookup section	
--	---------------------------------------	--

All functions accept one parameter, which is a text string. This string may contain a variable number of parameters, which are described later. The contents of this string follows the url-encoding standard². Note that character set conversion is performed before url decoding, which means that non-ASCII characters should *not* be url-encoded.

Example:

A search query may be specified by the following parameters:

- qry="Steen & Strøm"
- from=1
- to=10

This would be url-encoded like this:

```
qry=Steen+%26+Strøm&from=1&to=10
```

When using the UTF-8 character set, the binary stream to be sent over the wire as the SOAP parameter *qry* would be:

0x0000	71 72 79 3d 53 74 65 65	qry=Stee
0x0008	6e 2b 25 32 36 2b 53 74	n+%26+St
0x0010	72 c3 b8 6d 26 66 72 6f	r..m&fro
0x0018	6d 3d 31 26 74 6f 3d 31	m=1&to=1
0x0020	30	0

Note that **qry** is the name of the single SOAP parameter, and also the name of a commonly used parameter which is encoded *within* the SOAP parameter.

HTTP

There service can be accessed through two urls, one for searches and one for lookups. The parameters and uses of these are described in the sections search.php and lookup.php, respectively.

Parameters can be passed using either GET or POST (see 2).

Example:

```
http://live.intouch.no/tk/search.php?qry=ola+nordmann+fagernes&format=vcard
```

Queries are expected to be encoded in the ISO-8859-1 (Latin 1) character set. Results are in the same character set.

Make sure to properly url-encode parameters when using the GET method:

Example:

Steen & Strøm could be encoded as

Steen+%26+Strøm or

Steen%20%26%20Strøm or

Steen+%26+Str%f8m or

Steen+%26+Str%F8m

Search

This is the only method for performing searches in the directory, and it functions in the same manner whether it is accessed through HTTP or any of the search methods in the web service.

The parameters specific to this method are the search string, the result domain (private/public/business listings), the error tolerance to use and the result size. In addition are parameters for result format, loading of children and parent listings, and level of detail, which are in common with the lookup method.

Searches can either be specified in a single string which may contain both name, address, telephone number, business category etc., or in separate fields. It is usually recommended to use the single string even though your query source may contain structured information, among other things to avoid confusion about the status of middle names. The order of search terms within the single string is not critical, but slightly better results may be obtained when terms are listed in “mail address” order: First name, last name, street address, postal code, and location. Of course, it is not necessary to supply all terms in every search.

It is possible to narrow the search into any combination of the following categories:

- Private listings, corresponds to white pages + unlisted, unreserved mobile phones
- Public listings, belonging to public-service entities (libraries, municipalities etc.)
- Business listings, including yellow pages.

These categories are ignored when searching for a telephone number, in which case it makes little sense to narrow down the search.

Search-specific parameters

Field name	Content	Value	Default
qry	Query string as it appears in the full-text input field of TK bedrift or TK sms. The word "nær" has the special meaning "near", i.e. "svein nær asylet". Use <i>either</i> this field <i>or</i> the following 10 fields.	String. Examples: "thomas nylænder oslo" or "91887750"	
Qaqry	Used for splitting the query string left and right of "nær". The query qry="svein nær asylet" can also be specified as Qaqry="svein" and Qbqry="asylet". Qbqry can be used independently of Qaqry in conjunction with one or several of the fields below.	String	
Qbqry		String	
Qavdeling	Department name (only for internal directory)	String	
Qetternavn	Last name / company name	String	
Qfornavn	First and middle names	String	
Qveinavn	Street name	String	
Qhusnr	House number	String	
Qpostnr	Postal code (zip code)	String	
Qpoststed	Postal location (city name)	String	
Qbransje	Business category	String	
Qnumber	Telephone number (reverse lookup)	String	
Qpriv	Specify which parts of the directory to search: Private, Public, Business, Foretaksregisteret or Locdir (user-specific internal directory), respectively.	0 or 1	1
Qoff		0 or 1	1
Qnliv		0 or 1	1
Qbrreg		0 or 1	1
Qlocdir		0 or 1	1
Qsubscriber		If the user's telephone number is known, fill it into this field. The search engine will try to locate the subscriber's home address and bias the search result slightly towards listings that are close by. Also, if the user's telephone number is registered with one of Link Mobility's customers with a local directory, that directory is searched before the public directory. This is reflected in the result by setting the userID attribute to that customer's ID number instead of the search provider's ID number	String

Qmobilepeople	Set to 1 if search pertains to the Mobilepeople service, in which case the userID change related to Qsubscriber is only performed for those customers registered to use the Mobilepeople service	0 or 1	0
Qfodselsdato	Filter search result by birth date. Note: Birth date is not indexed, so this operation is performed after the search proper. Combine with parameters to=400 and maxpass=0.	Yyyy-mm-dd	
Qlat	Provide a latitude and longitude to find listings close to this position. Ranking will be strongly biased in favor of listings close to this reference point.	59.869	
Qlon		10.921	
from	From and to are used to limit the search results. From=1, to=5, will give the first five hits. From=6, to=10 will give the next five.	1-...	1
to		1-...	5
minpass	Searches are done in "passes", where the error tolerance is increased for each pass. Minpass specifies the minimum number of passes to use. Minpass=0 will use all available passes. Minpass=2 usually gives the best result. If no hits are found by pass 2, the error tolerance will automatically be increased.	0-10	2
maxpass	Searches are done in "passes", where the error tolerance is increased for each pass. Maxpass specifies the maximum number of passes to use. Maxpass=0 will use all available passes, and usually gives the best result.	0-10	0
filterduplicates	Some searches match at different levels of the same listing tree. If filterduplicates is set, all but the first match are removed.	0 or 1	1
extractdetails	If this is set, the search retrieves information about relevance score and details about which fields matched the query.	0 or 1	0
Qsortfields	Contains a list of fields to sort by, separated by space (which is url-encoded as + or %20). Leave field blank to use relevance ranking.	String. Example: "etternavn fornavn"	

If no searchfield are set, *and* only the Qlocdir field is set (i.e. Qpriv=0&Qoff=0&Qnliv=0&Qbrreg=0&Qloc=1), the entire user-specific internal directory is returned (limited by the from and to fields)

Parameters common to search and lookup

Field name	Content	Value	Default
format	The search result can be returned in different formats. This field specifies which of the to use	online html2 xml xmlsms xml_topdown html vcard bigvcard infoside minikart smskart sms xml_vcard xml_rdf	xml
loadparents	Specifies the number of generations of parent listings to load. -1 means all generations.	-1 or 0-...	-1
loadchildren	Specifies the number of generations of child listings to load. -1 means all generations.	-1 or 0-...	0
fuzzyexpand	Load those child listings that are most relevant to present to a user. Large listing trees may not have a telephone number in the root listing. Fuzzyexpand will then load the first sublistings which has a telephone number. If the listing tree is very small (i.e. only mobile and fax numbers), fuzzyexpand will load all sublistings.	0 or 1	1
loadnear	Only applicable to proximity searches. If this is set, the system will load those listings that matched the b-part of the search. Example: "svein nær asylet". If loadnear is set, the system will load not only the listings matching "svein" being close to "asylet", but also the listings that matched "asylet" and were close to "svein".	0 or 1	0

killduplicates	Remove all information about duplicates, keep only the first duplicate in a group.	0 or 1	0
splitduplicates	Split the duplicates into separate listings. See the example later in this document for an explanation of killduplicates and splitduplicates.	0 or 1	1
expand	Expand the listings by joining postal location and category names (using postal code and category id)	0 or 1	1
expand2	Expand the listings by joining telco name	0 or 1	0
loadcoords	Load coordinates (latitude, longitude and uncertainty) of each listing	0 or 1	0
ctrycode	Add country code +47 to all phone numbers	0 or 1	0

Lookup

After performing a search, it is possible to request additional information about the matched listings or their children or parents. Listings are referenced by their **id** number and the **table** value, both of which must come from a previous call to search or lookup. The id number is not persistent and changes after each index build, which happens daily. Lookups should therefore be done shortly after a search. Listings are numbered sequentially, and in order to avoid data sifting, it is necessary to supply some additional information about the requested listing. This can either be the **id** number of the listing's parent, the **id** number of one of its children or the contents of the listing's **idlinje** field.

The most common uses for lookup is to request multiple formats for the same result set or for loading children listings after a user requests it in an interactive user interface.

Lookup-specific parameters

Field name	Content	Value	Default
table	Name of the table where the listing resides	"listing", "locations", "forvalt" etc	
id	The id number of the listing	Integer	
ref	The id number of the parent or one of the child listings. This is necessary to prevent harvesting (extracting all id numbers)	Integer	
content	The contents of the field idlinje of the listing. Used to look up a known listing in a second format.	String	

In addition to these parameters, the lookup function accepts some parameters common to both the lookup and search functions, see the search section.

Example:

Search for “Norsk tipping boligtelefoner” returned listing ID=”159892”, table=”listings” and idlinje=”X58KVG0N”. Lookup can then be done with this string:

```
table=listings&id=159892&content=X53KVG0N&format=xml&loadchildren=-1
```

Note that id numbers change daily, so this example will not work unless you first search for the current ID numbers.

Format reference

vcard

This format is described in detail in [5]. This is a text-based format. Each hit is represented as one vcard, and a result set may contain several vcards.

Each vcard begins with the header “BEGIN:VCARD” and ends with the footer “END:VCARD”. Each vcard contains a number of attribute/value pairs, one per line. The line separator is NL+CR (char 13 + char 10). Attributes are separated from values by a colon (:). If a line contains more than one colon, all but the first one are part of the value. Each attribute may be of zero, one or more *types*. For instance, a telephone number may be for a normal phone, cellular phone or fax. The corresponding attribute, “TEL” may therefore have the types “CELL”, “FAX” or none, which signifies a regular telephone number. Type specifications follow immediately after the attribute name. Attributes and types are separated by semicolon (;). Some values consist of several parts. For example, the “ADR” attribute contains the fields “Pobox”, “Extended Address”, “Street”, “Locality”, “Region”, “Postal Code” and “Country”. These are separated by semicolon (;). The order of these fields is defined in the standard.

Example:

```
BEGIN:VCARD
VERSION:2.1
N:Dahl;Arne;;adv.;jr
ADR;HOME;CHARSET=ISO-8859-1;;;Ropernveien 5B;Snarøya;;1367;
TEL;PREF;HOME:67 53 86 58
END:VCARD
```

Any attribute which contains non-ASCII characters (æ, ø or å) have a type which specifies the character set. This type is “CHARSET=ISO-8859-1”. However, if the transport protocol uses another character set (UTF-8), this takes precedence and the charset type should be ignored.

Any attribute which contains control characters (newline and linefeed) have a type which specifies the encoding. This type is "ENCODING=QUOTED-PRINTABLE". This only applies to the "LABEL" attribute.

A vcard may contain more than one TEL attribute corresponding to different telephone numbers belonging to the same subscriber.

Attribute	Values	Examples	Types	Meaning
N	Family Name	"Dahl"	No types	
	Given Name	"Arne"		
	Additional Names	Always empty		
	Honorifix Prefixes	"adv."		
	Honorific Suffixes	"jr"		

Attribute	Values	Examples	Types	Meaning
ADR	Post Office Box	"Pb. 464 Økern"	WORK	Business address
	Extended Address	"0512 Oslo"	HOME	Home address
	Street Address	"Hagegata 273"		
	Locality	"Oslo"		
	Region	Always empty		
	Postal Code	"0589"		
	Country	Always empty		

Attribute	Values	Examples	Types	Meaning
ORG	Organisation Name	"Norsk Tipping"	No types	
	Organisation Unit	Always empty		

Attribute	Values	Examples	Types	Meaning
TEL	Telephone number	"991 69 127"	WORK	Business phone
			HOME	Home phone
			PREF	Preferred phone
			CELL	Cellular phone
			FAX	Telefax

Attribute	Values	Examples	Types	Meaning
EMAIL	Email address	"sveinb@pvv.org"	INTERNET	Always set

Attribute	Values	Examples	Types	Meaning
URL	Internet address	"http://www.db.no"	No types	

bigvcard

This format is similar to the **vcard** format, but contains more attributes:

Example:

```
BEGIN:VCARD
VERSION:2.1
FN:Arne Dahl jr adv.
N:Dahl;Arne;;adv.;jr
ADR;HOME;CHARSET=ISO-8859-1;;;Ropernveien 5B;Snarøya;;1367;
LABEL;PARCEL;ENCODING=QUOTED-PRINTABLE;CHARSET=ISO-8859-1:Arne Dahl jr
adv.=0D=0ARopernveien 5B=0D=0A1367 Snarøya
LABEL;INTL;ENCODING=QUOTED-PRINTABLE;CHARSET=ISO-8859-1:Arne Dahl jr
adv.=0D=0ARopernveien 5B=0D=0AN-1367 Snarøya=0D=0ANorway
TEL;PREF;HOME:67 53 86 58
END:VCARD
```

Attribute	Values	Examples	Types	Meaning
ORGNR	Organisation num.	900999999	No types	

Attribute	Values	Examples	Types	Meaning
FN	Display name	“Arne Dahl jr adv.”	No types	

Attribute	Values	Examples	Types	Meaning
LABEL	Display name	“Arne Dahl jr adv.=0D=0ARopernveien 5B=0D=0A1367 Snarøya”	PARCEL	For use on parcels
			INTL	For overseas use

xml_vcard

This format presents the same data as **bigvcard**, but in an XML format. Each vcard is represented by a **listing** element which contains one element for each vcard attribute. These elements are names identically to the vcard attribute, i.e. “N”, “ADR” etc. These elements contain “type” and “value” elements which contain the types and values. Only the **listing** element contains attributes. These are **table** and **id**, which can be used for later reference with **lookup**. The semantics of the vcard attributes and values are the same as for the vcard format (see previous section).

A search may return more than one listing. Therefore, the search result is returned in a **result** element inside a **asearch_vcard** element. The **result** element contains **hit** elements which in turn contain one **listing** element each. See also the XML schema in the WSDL file7.

xml_rdf

This format presents the same data as **bigvcard**, but in a standardized XML format⁶. Each vcard is represented by a **listing** element which contains one element for each vcard attribute. These elements are names identically to the vcard attribute, i.e. “N”, “ADR” etc. These elements contain “type” and “value” elements which contain the types and value. Those vcard attributes that contain subvalues (N, ADR, ORG), contain named elements with these values.

```
▼<vCard:ADR rdf:parseType="Resource">
  <rdf:type rdf:resource="http://www.w3.org/2001/vcard-rdf/3.0#work" />
  <vCard:Pobox/>
  <vCard:Extadd/>
  <vCard:Street>Rosenkrantz' gate 9</vCard:Street>
  <vCard:Locality>Oslo</vCard:Locality>
  <vCard:Region/>
  <vCard:Pcode>0159</vCard:Pcode>
  <vCard:Country/>
</vCard:ADR>
▼<vCard:ORGNR rdf:parseType="Resource">
  <rdf:value>992434643</rdf:value>
</vCard:ORGNR>
▼<vCard:LABEL rdf:parseType="Literal">
  <rdf:type rdf:resource="http://www.w3.org/2001/vcard-rdf/3.0#parcel" />
  ▼<rdf:value>
    Link Mobility AS
    <br/>
    Rosenkrantz' gate 9
    <br/>
    0159 Oslo
  </rdf:value>
</vCard:LABEL>
▼<vCard:LABEL rdf:parseType="Literal">
  <rdf:type rdf:resource="http://www.w3.org/2001/vcard-rdf/3.0#intl" />
  ▼<rdf:value>
    Link Mobility AS
    <br/>
    Rosenkrantz' gate 9
    <br/>
    N-0159 Oslo
    <br/>
    Norway
  </rdf:value>
</vCard:LABEL>
```

A search may return more than one listing. Therefore, the search result is returned in a **result** element inside a **search_vcard** element. The **result** element contains **hit** elements which in turn contain one **listing** element each.

```
▼<hit line="1">
  ▼<listing table="listing" id="200652">
    ▼<duplicates>
```

xml

This is the basic inner structure format. Listings currently contain the following fields:

Field name	Value
idlinje	Internal id value used for reference with lookup, example: "ZOHC1JXH"
tlfnr	(unformatted) Telephone number, example: "23327770"
etternavn	Last name
fornavn	First name
veinavn	Street name
husnr	House number
oppgang	Letter (add to house number)
postnr	Postal code
virkkode	Business type: P: Private N: Business B: Business (=N) O: Public I: Customer-specific register
apparatype	Device type: T: Telephone M: Mobile phone S: Switchboard F: Telefax X: Text telephone (for hearing impaired)
telco	Telephone operator. Two-letter abbreviation. See "telconame" below.
kilde	Origin of listing. May contain these values: G: Grensebasen (Company listings) P: privateListing (includes company mobiles) N: Nummerserier (from fixed line operators)
yrke	Profession
pbadresse	Post box number and address. Example: "273 Sentrum"
pbpostnr	Postal code associated with post box address
htlfnr	As soon as it is known that a telephone number will change, the old number appears in Htlfnr and the new one in tlfnr. The date of the switch is given in the field Gyldigfra on the format yyyy-mm-dd
gyldigfra	
foretaksnr	Company registration number
tlfnr_start	Number ranges span a series of telephone numbers starting at
tlfnr_stopp	tlfnr_start and ending at tlfnr_stopp
parentid	The id number of the parent listing, or 0 if the listing is root.
childrenid	The id number of the child listing. The number and order of these attributes corresponds to the number and order of the children.
ekstratekst	Extra text to be displayed with the name

epost	Email address
lenke	Web address
bransjekode	Yellow pages' internal business category code. See also the named category fields below. Can be 0 or blank, which indicates a listing which is not from yellow pages.
bransjer	Contain business categories from the user's own listings. The semantics is up to the user, and is usually used for department / division names for internal listing. Always empty in telephone directory listings.
daid	The id number of a display ad.
bkdata	Not in use
prioritet	Used with number series to indicate the relative importance of telephone numbers within one group of duplicates. Telephone numbers should be presented in descending order of <i>prioritet</i> .
thumbnail	Contains reference to thumbnail image. Only used with the user's own listings.
fodselsdato	Contains birth year of subscriber in YYYY format.
status	Only used with user's own listings. May contain "skjult", in which case the listing should not initially be displayed in an expand/collapse tree.
txtline	Used with company listings. May contain a few lines of text describing the business' products and services. Lines are separated by
 (five characters).
henvisning	Contains cross-references. If company is known under two names, <i>A</i> and <i>B</i> , its full listing being under the name <i>B</i> , there will be a reference on print in the form of "A see B". These two listings are joined to one, where the name fields reflect the <i>B</i> values, while <i>A</i> is in the henvisning field.
kommunenr	Official municipality number
kid	Customer identifier
lat	Latitude, signed floating point number, degrees east (only visible if loadcoords was set to 1)
lon	Longitude, signed floating point number, degrees north (only visible if loadcoords was set to 1)
dev	Incertainty in position, measured in meters (only visible if loadcoords was set to 1)
hbransjebokmaal	Contain business category names, divided in main category/subcategory (h / not h) and the two Norwegian written standards (Bokmål and Nynorsk). Only visible if expand was set to 1.
hbransjenynorsk	
bransjebokmaal	
bransjenynorsk	
kommune	Municipality name. Only visible if expand was set to 1.
fylke	County name. Only visible if expand was set to 1.
landsdel	Province name. Only visible if expand was set to 1.
poststed	Locality name referred to by postnr. Only visible if expand was set to 1.
pbpoststed	Locality name referred to by pbpostnr. Only visible if expand was set to 1.
telconavn	Name of subscription provider

bydel	Part of town, popular names
delivery	Infopage data: Provides delivery?
installment	Infopage data: Provides installments?
creditcard_list	Infopage data: List of accepted credit cards
bus_stop	Infopage data: Closest bus stop
AddressDescript	Infopage data: Popular address
PlaceNear1	Infopage data: First reference point
PlaceNear2	Infopage data: Second reference point
public_transport	Infopage data: Available means of transportation
popular_address_list	Infopage data: Popular addresses
opening_hours	Infopage data: Opening hours
inside_toll_zone	Infopage data: Inside toll road zone?
parking	Infopage data: Provides parking?
product_list	Infopage data: List of products

The section on the internal structure explains the relationship between listings, children and parents which is exposed in the xml format.

When using proximity search, and the *loadnear* search attribute is set to 1, each match will contain a *near* element which contains the listing which matched the right-hand side of the search. To calculate the (approximate) distance and bearing between the two matches, the following PHP code can be used:

```
function distancebearing($lat1,$lon1,$lat2,$lon2) {
    $Rearth=6371000;

    $dlat=($lat1-$lat2)*M_PI/180*$Rearth;
    $dlon=($lon1-
$lon2)*cos($lat1*M_PI/180)*M_PI/180*$Rearth;

    $dist=sqrt($dlat*$dlat+$dlon*$dlon);
    $dir=180/M_PI*atan2(-$dlon,$dlat);
    if ($dir<0) $dir+=360;

    return array($dist,$dir);
}
```

xml_topdown

This is the same format as *xml*, but with different relationships between listings, children and parents, as explained in the structure on the internal structure.

xmlsms

This is the same format as *xml*, but with the additional field *smsformat*, which contains the same textual representation as the *sms* format.

json

This format contains the same information as *xml*, but using the JSON^[10] format instead. JSON objects do not have an equivalent of XML attributes. Since there are no name collisions between attributes and elements in the XML schema, both attributes and elements are encoded as properties in JSON.

Example:

```
{
  "qry":"norsk tipping hamar",
  "result":{
    "hitLinesBeforeFilter":1,
    "approxHits":1,
    "userID":4,
    "1":{"listing":{
      "table":"listings",
      "id":"1734263",
      "duplicates":
      [
        {
          "table":"listings",
          "id":"1734263:0",
          "idlinje":"Z0GTGCSQ",
          "tlfnr":"62514000",
          "etternavn":"Norsk Tipping AS",
          "veinavn":"Jønsrudv.",
          "husnr":"21",
          "oppgang":"inng. Måsåbekkveien",
          "postnr":"2315",
          "virkkode":"N",
          "apparatype":"T",
          "telco":"TN",
          "kilde":"G",
          "foretaksnr":"925836613",
          "parentid":"0",
          "childrenid":"1409727",
          "childrenid":"1409728",
          "childrenid":"1409729",
          "childrenid":"1409730",
          "epost":"kundeservice@norsk-tipping.no",
          "lenke":"http://www.norsk-tipping.no",
          "bransjekode":"1272",
          "daid":"10159285",
          "prioritet":"0",
          "kommunenr":"403",
          "kid":"1086812",
          "poststed":"Hamar",
          "kommune":"Hamar",
          "fylke":"Hedmark",
          "landsdel":"Ø",
          "bransjebokmaal":"Lotteri- og spillvirksomhet",
          "bransjenynorsk":"Lotteri- og spelverksemd"
        },
        {
          "table":"listings",
          "id":"1734263:1",
          "idlinje":"Z0HBVS8X",
          "lenke":"www.norsk-tipping.no",
          "daid":"10159285",
          "kommunenr":"0"
        }
      ]
    }
  }
}
```

json2

This format is equal to the json format, except that matches are placed in an array named "hits" instead of numbered object properties. In addition, the field "startHit" contains the hit number of the first element in the "hits" array.

Foreign searches

A few foreign databases are connected to the search interface and can return results in the same formats as the domestic searches. However, the internal connection to these databases precludes the use of the lookup function and requires different search fields. For foreign searches, these fields are used in the search function:

Field name	Content	Value	Default
Qnavn	Name of person or company telephone, street address.	String. Example: "svein berge"	
Qsted	Name of geographic location	String. Example: "stockholm"	
Qland	2-letter iso code for country being searched	String. Currently accepted values: "se", "dk"	
Qpriv	Flag to indicate whether private or company listings should be searched.	"0" or "1"	

Code samples

This section contains snippets of code in different languages, using different features, to help you get started.

C# Example. Download here

<http://content.linkmobility.com/link/CExample.zip>

PHP, using SOAP and VCard.

```
<?
$Telefonkatalogen = new SoapClient("http://live.intouch.no/tk/webservice.php?wsdl&style=document");
print_r ($Telefonkatalogen->search(Array("qry"=>'qry=ola+nordmann+fagernes&format=vcard'));
?>
```

Visual Basic for applications, using SOAP and vcard format.

```
Sub testwebservice()
    Set Telefonkatalogen = New SoapClient30

    Telefonkatalogen.MSSoapInit _
        "http://live.intouch.no/tk/webservice.php?wsdl", _
        "Telefonkatalogen"

    Debug.Print Telefonkatalogen.search("qry=svein+berge+oslo&format=vcard")
End Sub
```

JavaScript/ECMAScript/Jscript, using HTTP and XML: Will not work in any web browser because of security restrictions on cross-domain scripting.

```
if (window.XMLHttpRequest) { // Mozilla, Safari, ...
    http_request = new XMLHttpRequest();
} else if (window.ActiveXObject) { // IE
    http_request = new ActiveXObject("Microsoft.XMLHTTP");
}

http_request.onreadystatechange=function(){alert(http_request.responseText)};

http_request.open('GET','http://live.intouch.no/tk/search.php?qry=svein+berge+oslo&format=xml');
http_request.send(null);
```

References

- [1] PHP language: <http://www.php.net>
- [2] HTTP protocol: <http://www.ietf.org/rfc/rfc2616.txt>
- [3] SOAP protocol: <http://www.w3.org/TR/soap/>
- [4] XML format: <http://www.w3.org/TR/REC-xml/>
- [5] VCARD format: <http://www.ietf.org/rfc/rfc2426.txt>
- [6] VCARD/XML/RDF format: <http://www.w3.org/TR/vcard-rdf>
- [7] WSDL file: <http://live.intouch.no/tk/webservice.php?wsdl> (style=document)
- [8] Search url: <http://live.intouch.no/tk/search.php>
- [9] Lookup url: <http://live.intouch.no/tk/lookup.php>
- [10] JSON: <http://www.json.org>