



livepro Search API Reference

15 July 2022

Version 2.7

Contents

Introduction	3
API Key.....	3
Function Reference	3
Test search form.....	3
A note on the API Key.....	3
Search.....	3
Announcements Search	4
Read	4
Report – Frequently Accessed	5
Virtual Agent Features	6
Feedback	6
Appendix	7
Record Content Formats by Type.....	7
Example Javascript, jQuery, Bootstrap implementation	12
Version Information	13

Introduction

The livepro Search API is used to search for records and retrieve record information from a livepro instance by a third party application.

API Key

An API key is required to interact with the server. The API key is directly related to a user in the livepro application and using the API key will expose the same information to the third party application that is available to the livepro user as if they had logged into the livepro application. Please contact livepro support to obtain a license key by emailing support@livepro.com.au

Function Reference

Test search form

A test search form is available at:
<server url>/search.html

Example:

<https://example.livepro.com.au/search.html>

A note on the API Key

You need to provide the API key with every call to the server. This can be provided as a GET parameter in the query string or as a request HEADER "x-api-key".

Search

Request

URI: <server url>/livepro-api/search

HTTP method: GET

Parameter: apikey

Parameter: query – the search terms (use * to retrieve all records)

Parameter: limit – the maximum number of records to return

Parameter: start – the record to start at in the list of results (used for pagination in conjunction with the limit parameter)

Parameter: outputformat (html/json) – how to return the results

JSON Response

success – whether the server was successfully queried or not. Always returns true currently.

parameters – an object containing the parameters provided in the request

count – the number of records returned in this response

total – the total number of records in the search

records – an array of records

errors* – (optional) an array of errors

JSON Search Response Record

A record consists of the following fields:

id – the id of the record

title – the title of the record

summary – the record summary of content

type – the type of record. Can have a value of (documentnote, workinstruction, hub, compass)

Example

<https://example.livepro.com.au/livepro-api/search?query=test&limit=25&start=0&apikey=MYAPIKEY&outputformat=json>

Announcements Search

An announcement search behaves very similarly to a normal “Search” but only returns announcements. You use the same “Read” function to view the announcement record.

Request

URI: <server url>/livepro-api/announcements

HTTP method: GET

Parameter: apikey

Parameter: query – the search terms (use * to retrieve all announcements)

Parameter: limit – the maximum number of records to return

Parameter: start – the record to start at in the list of results (used for pagination in conjunction with the limit parameter)

Parameter: outputformat (html/json) – how to return the results

JSON Response

success – whether the server was successfully queried or not. Always returns true currently.

parameters – an object containing the parameters provided in the request

count – the number of records returned in this response

total – the total number of records in the search

records – an array of announcement records

errors* – (optional) an array of errors

JSON Search Response Record

A record consists of the following fields:

id – the id of the record

title – the title of the record

summary – the record summary of content

type – the type of record – will always be ‘announcement’.

Example

To show the 5 latest announcements:

<https://example.livepro.com.au/livepro-api/announcements?5&apikey=MYAPIKEY&outputformat=json>

Read

Request

URI: <server url>/livepro-api/read

HTTP method: GET

Parameter: apikey

Parameter: id – the id of the record to read (see Parameter: name)

Parameter: name – the permalink of the record to read (see Parameter: id)

NOTE: either id or name should be provided – not both.

Parameter: hidetitle(true/false) – will hide the title section when outputformat is html

Parameter: outputformat (html/json) – how to return the results. If html is selected then the livepro

application will render the content using the styles defined in the livepro application.

Parameter: `call_function` – the callback function to call when a livepro link is clicked. This function will take a record id as a parameter. Eg:

```
function openLiveproRecord(idValue){  
    console.log(idValue);  
}
```

Response

success – whether the server was successfully queried or not. Always returns true currently.

record – the record queried

errors* – (optional) an array of errors

JSON Read Response Record

A record consists of the following fields:

id – the id of the record

type - the type of record. Can have a value of (documentnote, workinstruction, hub, compass)

title – the title of the record

permalink – the name of the record

content – this is the data of the record and it will change depending the type of record (see Appendix: Record Content Formats by Type)

related_links – links to related records (see JSON Related Link Response Record)

JSON Related Link Response Record

A related link record consists of the following fields:

title – the display text of the link

ordering – the order the link should be displayed in

to_record_id – the id of the related record

Example

https://example.livepro.com.au/livepro-api/read?apikey=MYAPIKEY&id=100&outputformat=json&call_function=myCallBack

Report – Frequently Accessed

Request

URI: <server url>/livepro-api/report/frequent_access

HTTP method: GET

Parameter: `apikey`

Parameter: `limit` – the maximum number of records to return (defaults to 10).

Parameter: `from_date` – the start date of the report in dd/MM/yyyy format (eg. 31/12/2015)*

Parameter: `to_date` – the end date of the report in dd/MM/yyyy format (eg. 31/12/2015)*

* Note: `from_date` and `to_date` are optional – if either is not set it will default to the last month

Response

- Always a json response

success – whether the server was successfully queried or not. Always returns true currently.
records – the record returned
count – the number of records returned
errors* – (optional) an array of errors

JSON Response Record

A record consists of the following fields:

id – the id of the record
title – the title of the record
summary – the record summary of content
url – the url to access the record

Example

https://example.livepro.com.au/livepro-api/report/frequent_access?apikey=MYAPIKEY&limit=5

Virtual Agent Features

If virtual agent features are enabled and virtual agent content is available, then the following properties will appear in the JSON Read Response Record. This is only available for Document Note currently. These fields mirror the fields in the “Virtual Agent” tab of the document note when enabled.

virtual_agent_content – container object for virtual agent content containing:

chat – plain text content associated with chat
voice – plain text content associated with voice
other – plain text content

Feedback

You can post a feedback entry against a knowledge object version.

Post Request

URI: <server url>/livepro-api/feedback

HTTP method: POST

Parameter: apikey

BODY:

```
{
  "feedback": {
    "content": <FEEDBACK_CONTENT>,
    "knowledge_object_version_id": <ID OR EMPTY FOR GENERAL FEEDBACK>
  }
}
```

Appendix

Record Content Formats by Type

Document Note (documentnote)

content – html content

Eg:

```
{
  "success":true,
  "record":{
    "id":"210",
    "type":"documentnote",
    "title":"Sugar Price List",
    "permalink":"sugar-price-list",
    "content":"<h1>Some Content</h1>\n<p>Content TBA</p>",
    "related_links":[
      {
        "title":"Coffee Price List",
        "ordering":"1",
        "to_record_id":"709"
      }
    ],
    "virtual_agent_content":{
      "chat":"Sugar costs the following...",
      "voice":"Sugar costs the following...",
      "other":""
    }
  }
}
```

Work Instruction (workinstruction)

steps – array of content forming work instruction steps

id – id of the step

title – display title of the step

content – html content

links – links between steps from one step to another step

title – display title of the link

ordering - the order the link should be displayed in

from_record_id – the step record id this link is from

to_record_id – the step record id this link is to

Eg:

```
{
  "success":true,
  "record":{
    "id":"210",
    "type":"workinstruction",
    "title":"How to make coffee",
    "permalink":"how-to-make-coffee",
    "content":{
      "steps": [{
        "id": "229",
        "title": "Put water in kettle",
        "content": "<p>Fill the kettle to the max line.</p>"
      },{
        "id": "230",
        "title": "Get a cup",
        "content": "<p>Find a cup in the cupboard</p>"
      },{
        "id": "231",
        "title": "Wait for the kettle to boil",
        "content": "<p> Wait for the kettle to boil</p>"
      }],
      "links": [{
        "title": "I don't have a cup",
        "ordering": "0",
        "from_record_id": "229",
        "to_record_id": "230"
      },{
        "title": "I have a cup",
        "ordering": "1",
        "from_record_id": "229",
        "to_record_id": "231"
      }]
    },
    "related_links":[]
  }
}
```

Hub

start_content – html content at the start of the record display

hub_links – array of links to records

icon_id – this link's icon id. This can be used to retrieve an icon for display by calling:

<https://example.livepro.com.au/livepro-api/read?apikey=MYAPIKEY&id=63&context=image>

title – display title of the link

ordering – the order the link should be displayed in

to_record_id – the record id this link is to

end_content - html content at the end of the record display

Eg:

```
{
  "success":true,
  "record":{
    "id":"210",
    "type":"hub",
    "title":"Types of Coffee",
    "permalink":"types-of-coffee",
    "content":{
      "start_content": "<p>Below are some types of coffee.</p>"
      "hub_links": [{
        "ordering": "0",
        "title": "Robusta",
        "to_record_id": "658",
        "icon_id": "65"

      },{
        "ordering": "0",
        "title": "Arabica",
        "to_record_id": "659",
        "icon_id": "66"
      }],
      "end_content": "<p>Thank you.</p>"
    },
    "related_links":[]
  }
}
```

Compass

start_content – html content at the start of the record display

questions – array of questions and answers

title – the question

alias – a code for the question

answers – array of possible answers

title – the answer

alias – a code for the answer

outcome_map – array of possible selections and responses in the format of

“question alias” : “response alias”

for each question. Then

outcome – the alias of the outcome for the question/answer state

outcomes – array of records which are selected from the outcome map

to_record_id – the record to go to

alias – the code for the outcome in the outcome_map

Eg:

```
{
  "success":true,
  "record":{
    "id":"210",
    "type":"compass",
    "title":"Can I get a drivers license",
    "permalink":"can-i-get-a-drivers-license",
    "content":{
      "start_content": "<p>Please answer the questions below:</p>",
      "questions": [{
        "title": "Do you own a car?",
        "alias": "own-car",
        "answers": [{
          "title": "Yes",
          "alias": "yescar",
        },{
          "title": "No",
          "alias": "nocar",
        }]
      },{
        "title": "Are you over 18?",
        "alias": "over-18",
        "answers": [{
          "title": "Yes",
          "alias": "yes18",
        },{
          "title": "No",
          "alias": "no18",
        }]
      }],
      "outcome_map": [{
        "own-car": "yescar",
```

```

        "over-18 ": "yes18",
        "outcome": "cangetlicense"
    },{
        "own-car": "nocar",
        "over-18 ": "yes18",
        "outcome": "cannotgetlicense"
    },{
        "own-car": "nocar",
        "over-18 ": "yes18",
        "outcome": "cannotgetlicense"
    },{
        "own-car": "nocar",
        "over-18 ": "no18",
        "outcome": "cannotgetlicense"
    }],
    "outcomes": [{
        "to_record_id": "450",
        "alias": "cannotgetlicense"
    },{
        "to_record_id": "451",
        "alias": "cangetlicense"
    }]
}
"related_links":[]
}
}

```

Example Javascript, jQuery, Bootstrap implementation

Two files are included as part of this documentation:

`api-include.js`

`apitest.html`

Version Information

Changed By	Date	Version Number	Notes
Lester Bloomberg	19/11/2015	1.0	Initial version
Lester Bloomberg	02/12/2015	1.1	Added: Report – Frequently Accessed Fixed: Added apikey variable to read example. Added * note to search query parameter
Lester Bloomberg	14/03/2017	2.0	Expanded API and added record types and example code.
Lester Bloomberg	23/03/2017	2.1	Changed example code from “apitest.php” to “apitest.html” Included example code for Compass record type in apitest.html
Lester Bloomberg	13/06/2017	2.2	Added announcement call documentation.
Lester Bloomberg	15/09/2017	2.3	Added hidetitle option to read request NOTE: CHANGE: compass opens option in new window.
Lester Bloomberg	22/12/2017	2.4	Added HEADER option for apikey
Lester Bloomberg	22/01/2018	2.5	Added livepro branding
Lester Bloomberg	24/04/2019	2.6	Added Virtual Agent Features section Added virtual agent property to Document Note JSON
Lester Bloomberg	15/07/2022	2.7	Added feedback submission