API

A guide for managers & developers

PART 2: Courses Search

Prepared by OfCourseMe

OfCourseMe APIs

1. Courses Search

“Courses Search” is the service that allows you to make OfCourseMe’s search engine for 3rd parties online learning contents available to your employees, wherever you want.

The most common use cases for this service are:

- **I want a brand new section dedicated to learning contents from the Web.** In this section I want a prominent search bar. **When the user types a text search in the bar, I want to show her all the results found by OfCourseMe that are relevant to that query.**

- **I already have a search bar in our website/LMS, when a user types a search we give her our internal results relevant for that search. We do not have relevant content for all possible topics, hence we often show few or no results. I want to beef them up with OfCourseMe results, displaying them after our internal courses if there are any, but making sure we drastically increase our coverage.**

“Courses Search” service comes with filters available. Via our APIs all searches can be filtered by category, language, platform, level, duration etc. which allows you to put those filters on page and let your users filter their queries as they wish.

An example of a very specific filtered search might be:

“All search results for the keyword “Artificial Intelligence”, in English only, from Udacity only, for advanced users, with a minimum duration of 2 hours”. A pretty powerful tool, indeed.

When the user clicks on the “search” button, your systems send a request to our APIs which specifies what text search was submitted, with which filters. **Our APIs respond with a list of contents in milliseconds.** Those contents come with elements such as title, image, syllabus, level, duration, so that your developers get everything they need in order to build the course cards in the search results page, in real time. **You choose which data you want to show for each card, depending on how much space is available, which devices and screens your colleagues usually adopt, and your personal taste.**

OfCourseMe APIs

1. Courses Search

Please find below the key basic information in order to get started, for developers

Description

Call endpoint: `https://api.ofcourse.me/v2.0/courses-search`

Filters

You can filter down your calls to get a text-search or courses in a category. Here are listed all the parameters you can use to filter your calls, they are all optional:

- `query`
set this parameter to make a text-search. Text-search is performed searching for the desired term in course title, course subtitle and course associated categories.
- `category` (multiple values, comma separated)
set this parameter to filter your query by category. Pass one or more ID_CATEGORY divided by comma. You can get all the categories id by calling the /categories endpoint
- `language` (multiple values, comma separated)
set this parameter to filter your query by language. Pass one or more language-code divided by comma. You can get all the language-codes (two-character codes) value by calling the /options endpoint
- `platform` (multiple values, comma separated)
set this parameter to filter your query by platform. Pass one or more platforms divided by comma. You can get all the platforms value by calling the /options endpoint
- `level` (multiple values, comma separated)
set this parameter to filter your query by level. Pass one or more levels divided by comma. Level values are documented in the Key Values chapter.
- `effort` (multiple values, comma separated)
set this parameter to filter your query by effort. Pass one or more effort ranges divided by comma. Effort range values are documented in the Key Values chapter.
- `price_type` (multiple values, comma separated)
set this parameter to filter your query by price type. Price type values are documented in the Key Values chapter.
- `price` (multiple values, comma separated)
set this parameter to filter your query by price ranges. Pass one or more price ranges divided by comma. Price range values are documented in the Key Values chapter. Price value will be ignored if price_type is set only to free ("F").
- `provider` (multiple values contained in double quotes, comma separated)
set this parameter to filter your query by providers. You can get all providers names by calling the /options endpoint.
Pass one or more provider name contained in double quotes and comma separated (syntax

example, **provider:"Amazon","Edx"**).
Be aware you must encapsulate each provider name in double quotes and separate them by comma or you will get an "INVALID-PROVIDER-VALUE" error.
See this examples to understand how to use this filter.

- `course_id` (multiple values, comma separated)
set this parameter to one or more course_id (comma separated) to get only courses specified.
- `edited_after` (UTC timestamp in millisecond. Example 1537793541000)
set this parameter to filter your query by course edited time. Pass a UTC timestamp in milliseconds to get courses edited in our database after this timestamp (including specified millisecond). This parameter is useful to obtain only those courses edited in our database since your last request with the same parameters.
- `lang` (two digit language code)
set this parameter to get course categories in the language needed. You can obtain available languages querying the "option" endpoint with type "translations_languages" (/options?type=translations_languages)

Invalid parameters will raise an `INVALID-PARAMETER` error.
Invalid parameters values will raise an `INVALID-<PARAMETER_NAME>-VALUE` error.

Pagination

Result set is limited to maximum 30 elements per call.

You need to use `start` and `size` parameters to paginate your results:

- `start` is the pointer to the starting record in the result set
- `size` is the length of the result set.

If you don't pass the `start` parameter it will be 0 by default.

If you don't pass the `size` parameter it will be 20 by default.
The maximum value for the `size` parameter is 30.
If you pass a value over 30, it will be changed to 30 as the result set is limited to a maximum of 30 elements per call.

Sorting

You can pass an `order` parameter to sort your result set.
`Order` values are documented in Key Values chapter.
If you don't pass any `order`, the result set will be sorted by default (relevance based on what you searched)

Response Fields

Here are listed the response fields for the course:

- `course_id`: unique ID of the course
- `course_title`: title of the course (string)
- `course_syllabus`: syllabus of the course (string). Syllabus is truncated at 140 characters and sanitized removing html tags inside it to give you a preview of the entire text. To obtain the entire syllabus you must call a single /course endpoint.

- `course_image`: small image of the course (url) for listing purpose. In case no image is provided for a given course, we will serve a fallback image related to that course's category
- `course_hosting_platform`: hosting platform of the course. Call endpoint /options to understand returned values
- `course_owner`: owner of the course (string array)
- `course_provider`: provider of the course (string array)
- `course_level`: level of the course. See Key Values chapter to understand returned values.
- `course_duration`: duration of the course (minutes)
- `course_certificate_available`: defines if the course has a certificate available: "Y" (available) / "N" (unavailable)
- `course_pricing_value`: cost of the course (dollars)
- `course_pricing_type`: type of course pricing. See Key Values chapter to understand returned values.
- `course_active`: defines if the course is still available on the hosting platform: "Y" (active), "N" (inactive).
- `course_release_date`: course_release_date: release date of the course (date)
- `course_categories`: an array of course categories (object array). Each category object will have these attributes:

  `course_category_id`: category id
  `course_category_name`: category name
  `course_category_level`: category level (0 for parent categories, 1 for subcategories)
  `course_category_parent_id`: category parent id (0 for parent categories)
  `course_category_parent_name`: category parent name (null for parent categories)
- `search_count`: total count of courses matching search's criteria.

Examples

`https://api.ofcourse.me/v2.0/courses-search?query=Javascript`
Returns first 20 (default values for start=0, size=20, ordered by relevance) courses matching the "Javascript" term in their title, subtitle or categories.

`https://api.ofcourse.me/v2.0/courses-search?query=Javascript&lang=it`
Returns first 20 (default values for start=0, size=20, ordered by relevance) courses matching the "Javascript" term in their title, subtitle or categories. Course categories are translated in italian

`https://api.ofcourse.me/v2.0/courses-search?query=Javascript&level=1`
Returns first 20 courses matching "Javascript" term AND which level is Beginner

`https://api.ofcourse.me/v2.0/courses-search?query=Javascript&level=1,3`
Returns first 20 courses matching "Javascript" term, AND which level is Beginner OR Advanced

`https://api.ofcourse.me/v2.0/courses-search?query=Javascript&order=ED&start=100&size=10`
Returns 10 (size) courses starting from record 100 (start), ordered by effort (descending), matching "Javascript" term

`https://api.ofcourse.me/v2.0/courses-search?query=Javascript&level=1&platform=1`
Returns first 20 courses matching "Javascript" term AND which level is Beginner AND which platform is Coursera

https://api.ofcourse.me/v2.0/courses-search?query=Javascript&level=1,3&platform=1,3
Returns first 20 courses matching "Javascript" term, AND which level is Beginner OR Advanced, AND which platform is Coursera OR Udacity

https://api.ofcourse.me/v2.0/courses-search?query=Javascript&category=22
Returns first 20 courses matching "Javascript" term which category is Web Development

https://api.ofcourse.me/v2.0/courses-search?category=22
Returns first 20 courses which category is "Web Development"

https://api.ofcourse.me/v2.0/courses-search?category=22,23
Returns first 20 courses which category is "Web Development" or "Programming Languages"

https://api.ofcourse.me/v2.0/courses-search?category=22&effort=1&start=100&size=30
Returns first 30 (size) courses, starting from record 100 (start), which category is "Web Development" and which effort is less than between 20 minutes and 2 hours

https://api.ofcourse.me/v2.0/courses-search?level=1
Returns first 20 courses which level is "Beginner"

https://api.ofcourse.me/v2.0/courses-search?level=1&effort=2,3
Returns first 20 courses which level is "Beginner" AND which effort is between 2 hours and 10 hours OR between 10 hours and 100 hours

https://api.ofcourse.me/v2.0/courses-search?price=1&order=PD
Returns first 20 courses which price is less than 50 dollars, ordered by price descending

https://api.ofcourse.me/v2.0/courses-search?price=1,3&order=PD
Returns first 20 courses which price is less than 50 dollars or between 100 and 300 dollars, ordered by price descending

https://api.ofcourse.me/v2.0/courses-search?course_id=286226,293406
Returns course which course_id is 286226 or 293406

Provider Filter Examples

https://api.ofcourse.me/v2.0/courses-search?provider="Amazon"
Returns first 20 courses which provider is Amazon

https://api.ofcourse.me/v2.0/courses-search?provider="Amazon","Edx"
Returns first 20 courses which provider is Amazon OR Edx

https://api.ofcourse.me/v2.0/courses-search?provider=Amazon
This call is invalid as provider name is not encapsulated in double quotes

https://api.ofcourse.me/v2.0/courses-search?provider="Amazon";"Edx"
This call is invalid as providers names are not comma separated

Subset

Learn more about subset and exclusive content

Your calls could be limited by a defined subset.
Calling this endpoint without any parameter will return you all courses included in your subset.

Assume your subset defines that you can see only courses which are related to category "22" ("Web Development").
This means you can never retrieve courses associated ONLY to other categories, but you can always retrieve courses related to other categories AND category "22".
Assume also that there exists at least a course which is associated with "Web Development" ("22") AND "Programming Languages" ("23"), this course is part of your subset.
Assume your subset defines that you can see only courses which are related to platform "3" ("Udacity").
This means you can never retrieve courses associated with other platforms, but only those courses served by Udacity.

Subset could be defined by a combination of multiple parameters.
Assume your subset defines that you can see only courses which are related to category "22" ("Web Development") and platform "3" ("Udacity").
This means you can never retrieve courses associated ONLY to other categories, but you can always retrieve courses related to other categories AND category "22".
It also means that those courses should be served by "Udacity".
Assume that there's at least a course which is associated with "Web Development" ("22") AND "Programming Languages" ("23") served by platform "1" ("Coursera"), this course is not part of your subset.
Assume also that there's at least a course which is associated with "Web Development" ("22") AND "Programming Languages" ("23") served by platform "3", this course is part of your subset.
You should not do anything to limit your subset, as it is automatically computed on your API-key or IP address. See below example for more information

`https://api.ofcourse.me/v2.0/courses-search`
Assume your subset defines that you should see only courses related to category "22" and exist at least one course which is related to category "22" AND another category,
returns all courses directly associated to category "22", courses associated to category "22" AND other categories are included

`https://api.ofcourse.me/v2.0/courses-search?category=23`
Assume your subset defines that you should see only courses related to category "22" and exist only one course which is related to category "22" AND category "23",
returns only the course associated to category "22" AND category "23"

`https://api.ofcourse.me/v2.0/courses-search`
Assume your subset defines that you should see only courses served by platform "3",
returns all courses served by platform "3"

`https://api.ofcourse.me/v2.0/courses-search?platform=4`
Assume your subset defines that you should see only courses served by platform "3",
returns no courses.