API

A guide for managers & developers

PART4: **Categories**

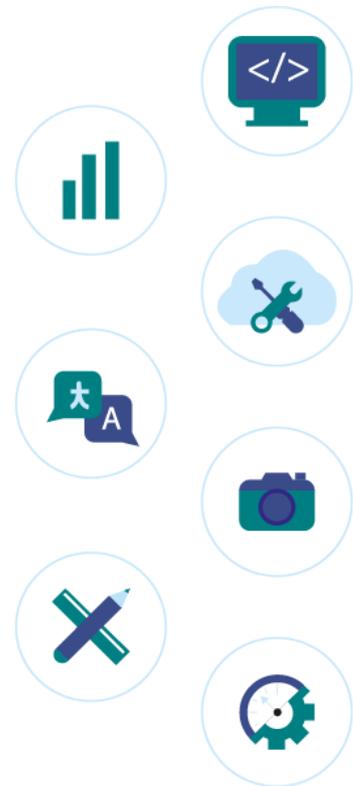OfCourseMe APIs

3. Categories

> "Categories" is the service that allows you to implement a menu with icons and links to OfCourseMe's categories.

When you click on a link you are redirected to a search results page (SERP) with all the contents available for that category. Key features of the categories menu are:

- The menu is structured on 2 levels:
  - 20 macro categories,
  - >150 sub-categories

- If you have decided to limit the categories available for your employees to a given subset, you will retrieve only categories belonging to that subset.

- When the user clicks on a link to a given category, the search engine requests our API endpoint courses-search all the results available for that category in your subset.

- The same content might be tagged in different categories: i.e "History of philosophy" will be both in History and Philosophy.

OfCourseMe APIs
3.Categories
Please find below the key basic information in order to get started, for developers

Goal
Endpoint "/categories" is meant to retrieve more details about ALL the categories (topics) associated with at least one course in your subset.
If your subset includes "web development" category but does not include "game design" category, and there is one course about "web developmen" applied to "game design" which is in your subset and is associated to both, endpoint "/categories" will retrieve details about both.

Endpoint "/categories" IS NOT meant to retrieve by default details about the categories in your subset ONLY.
If you want to retrieve details about your selected categories only, please use the list of categories in your subset as a filter in your call to the endpoint "/categories". See details in section "Subset" below.

Description
Call endpoint `https://api.ofcourse.me/v2.0/categories`.

No parameters are needed for this method, but if you want to receive details for a selected categories only then you are allowed to pass the category parameter.

This parameter should be filled with categories ids divided by comma or you'll get the "INVALID-CATEGORIES-IDS-PARAMETER" error.

If you pass a category id which does not correspond to any categories in our database, you'll get a "NO-CATEGORIES-FOUND" error.

You can pass a parameter "lang" (two digit lang code) to get categories in the language needed. You can obtain available languages quering the "option" endpoint with type "translations_languages" (/options?type=translations_languages)

Response fields
Here are listed all response fields for a single category
- `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)
- `course_category_image`:OfCourseMe image for this category (subcategories will have same image of their parent).

Examples

https://api.ofcourse.me/v2.0/categories
Returns all categories related to your subset

https://api.ofcourse.me/v2.0/categories?lang=it
Returns all categories related to your subset, translated in italian

https://api.ofcourse.me/v2.0/categories?category=119,130
Returns categories which id is 119 and 130

Subset
[Learn more about subset and exclusive content](#)

Your calls could be limited by a defined subset of courses based on the categories they are associated with.
Assume your subset defines that you can access only to contents which are related to category "22" ("Web Development").
This means you can retrieve all courses associated to category "22"; It also 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.

This behavior also affects this endpoint.
Calling /categories endpoint with no params will result in ANY category related to courses defined in your subset.
For the same reason, calling /categories endpoint will not necessarily result only in retrieving the categories selected in order to determine your subset.
In this case, your subset is limited only to courses related category "22".
So, calling this endpoint, you will NOT receive only category "22" but ALL categories which are related to courses directly related to category "22".
As we have assumed that there exists at least a course which is associated to "Web Development" ("22") AND "Programming Languages" ("23"), this endpoint response will also include category "23" as well as other categories related to all courses in category "22".

Note also that parent categories are always returned by this endpoint.
If your subset is defined on a parent category all child categories will be returned by this endpoint.
If your subset is defined on a child category, its parent category will be returned by this endpoint.

Should you need to ONLY show category "22" in your platform using this endpoint, you should **filter this endpoint using a category parameter.**

Important notice:
You can use every category returned by this endpoint in /courses-search category parameter, always not exceeding your subset definition.
In this case, if you ask for courses related to category "23" ("Programming Languages"), as your

subset defines these courses being related also to category ("22"), you'll get only courses related to category "22" and "23", even not specifying category "22" in your query.
See below examples for more detail.

https://api.ofcourse.me/v2.0/categories
As 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 category "23",
returns category "22", category "23" and their parent categories

https://api.ofcourse.me/v2.0/categories?category=22
Returns category 22
You can prevent applying your subset to get all available OfCourseMe categories passing a parameter filter_by_subset with value 0. See example below.

https://api.ofcourse.me/v2.0/categories?filter_by_subset=0
Get all OfCourseMe categories not limited by your subset

User tracking parameters
It is possible, if desired, to identify a single user by appending a "user_id" parameter to your API-calls.
You can use a session id or an internal id (integer/string) to set the "user_id" parameter.
Also, if desidered, you could identify a user group, family or entity by appending a "entity_id" parameter to your API-calls.
You can use an internal group, family or entity id (integer/string) to set the "entity_id" parameter.
This will help us computing API-usage statistics and metrics by single user and / or by user group / or by user family / or by user entity.
User tracking parameters are optionals.

https://api.ofcourse.me/v2.0/categories?user_id=USER-ID
The "user_id" parameter is properly set to identify a single user.

https://api.ofcourse.me/v2.0/categories?entity_id=GROUP-ID
The "entity_id" is properly set to identify user group.

https://api.ofcourse.me/v2.0/categories?user_id=USER-ID&entity_id=GROUP-ID
The "user_id" parameter is properly set to identify a single user and "entity_id" is properly set to identify user group.

https://api.ofcourse.me/v2.0/categories?user_id=USER-ID&entity_id=FAMILY-ID
The "user_id" parameter is properly set to identify a single user and "entity_id" is properly set to identify user family.

https://api.ofcourse.me/v2.0/categories?user_id=USER-ID&entity_id=ENTITY-ID
The "user_id" parameter is properly set to identify a single user and "entity_id" is properly set to identify user entity.

DEVELOPERS

Code Examples

- JS (jQuery AJAX)
- PHP (CURL)
- C# (WebRequest)
- Java (HttpUrlConnection)

For testing purpose only

```
$.ajax({
    url: "https://api.ofcourse.me/v2.0/categories",
    type: GET,
    headers: {"x-api-key": "********************************IcAuSy"}
// insert your API Key
})
.done(function(response) {
    console.log(response);
})
.fail(function(err) {
    console.log(err);
});
```