

API

A guide for
managers &
developers

PART 7: Recommendations

Prepared by OfCourseMe

6. Recommendations

“Recommendation” service is a powerful tool to deliver automatically a custom learning experience to each single user.

Just like on Spotify or Netflix, the use case is the following:

- I want to provide the user with a section “courses you might also like” which displays contents in line with that user’s preferences and behaviours, in order to maximize her likelihood to embrace new contents and ultimately her engagement

In short, OfCourseMe tracks user interactions with the contents (searches, clicks etc.), then our algorithms weigh those interactions and proactively suggest new contents.

Easier done than said!



OfCourseMe APIs

6. Recommendations

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

Description

Call endpoint <https://api.ofcourse.me/v2.0/recommendations>

This endpoint is useful to get recommended courses for a specific user based on previous performed and tracked searches.

To use this endpoint you should before have tracked your /courses-search API-calls appending **user_id** parameter (see User tracking parameters)

To use this endpoint you must pass the same **user_id** parameter (mandatory) which you use to track the user /courses-search

If you don't pass a **user_id** parameter or you pass an invalid value, the API will return an "INVALID-USER-ID" error.

Examples

https://api.ofcourse.me/v2.0/recommendations?user_id=1

Returns recommended courses for user identified by id 1 based on previous performed and tracked searches

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 recommended.

User tracking parameters

It is possible, if desired, to identify a single user by appending a **user_id** parameter to your API-calls.

To use this endpoint you should before have tracked your /courses-search API-calls appending **user_id** parameter.

You can use a session id or an internal id (integer/string) to set **user_id parameter for /courses-search API-calls**.

Having tracked your /courses-search API-calls with a **user_id**, we are able to retrieve recommendations for your user if you pass the same **user_id** to this endpoint.

Also, if desired, 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, but "user_id" is needed to track search and get recommendations.

Here is a basic example about the "/courses-search" endpoint.

https://api.ofcourse.me/v2.0/courses-search?user_id=USER-ID

The "user_id" parameter is properly set to identify a single user

Code Examples

[JS \(jQuery AJAX\)](#)

[PHP \(CURL\)](#)

[C# \(WebRequest\)](#)

[Java \(URLConnection\)](#)

For testing purpose only

```
$.ajax({
  url: "https://api.ofcourse.me/v2.0/recommendations?user_id=1",
  type: GET,
  headers: {"x-api-key": "*****lcAuSy"} // insert your API Key
})
.done(function(response) {
  console.log(response);
})
.fail(function(err) {
  console.log(err);
});
```