How to create, edit and manage course related data

Intro

This document will go through the necessary steps to create a course. As an example we will be using our Acceptance environment. For reference, our Azure API Management (APIM) is located (https://acc-developers.edums.nl/)

A course is a type of product in EduMS, the only one for now actually and more details can be found (Course object model - EduMS)

As a provider, you can create, update and remove courses in the EduMS platform. Courses can be added in 3 different ways:

1. Imported by publishing course data through EDU-DEX (https://edudex.nl/en/contact/#join)
2. Manually through the GUI for Providers.
3. Sending a POST request to the **POST Course endpoint on the Provider API**.

The model details can be found on the **POST Course endpoint** in the **Request body** section. Several properties listed in the model are of type "enum". Those properties only accept a specific range of enumerated values and a link to the full enum overview is also available on the endpoint documentation. (Enumerations across all APIs)

When creating courses we distinguish 3 types of courses:

1. **Planned**: The course is bound to specific dates on which classes or events take place.
2. **Self Planned**: The student can make their own planning and consume content on-demand.
3. **Blended**: A combination of specific dates for classes/events and on-demand content.

The main building blocks of the course

Segments

This is the way to categorize the course. As an example we can create a course in which the segment might be AutomationAndIT and the sub segment AutomIT_Training, which means we further categorize the course into training (subSegment) in automation and Technology Information (Segment).

You can only add sub segments if the segment they belong to is already selected for that course

How to add Segments to the course

1. Perform a GET list for segments (ProviderAPI - Segments List)
2. Save the segment name
3. Perform a PUT request but now add the segments to the payload and the new segment name as the value
4. The request should be successful

Learning Method

This section contains the planning, startmoments and invoice items, which store the information about the date and location and the different cost types of the course.

Startmoments

There is the need to save the information about where and when will the classes take place when the courses are planned or blended, so startmoments do exactly that.

Invoice Items

You can find invoice items under the learning method/costs/details objects.

Save the information related to costs applied to the course which can be generic or per startmoment.

Generic

A generic invoice item is the concept of applying a cost to the whole course.

Per startmoment

There is the need to specify costs that are only for that location and time, something like a lunch which only happens that day for instance.

Content

In the content area we store the blocks which hold the course information to be displayed by our integrators.

You can fill more generic information in the content description and introduction and more specific information in the blocks, by adding as much as you need, filling in the type, body and bodyHtml. We do strip the text from the html and put it into the body if posted empty.

What you need for every request

1. Get a token - Find the Identity Server API, and get a token with your client_id and client_secret
2. Add a new header under the section Headers, and type Authorization and then Bearer followed by the token

How to create a course from scratch

This is the minimal sample, refer to the Course object model - EduMS for more properties

1. Create a provider and save its Id
2. Use the below sample, replace the providerId and you should get a successful request

```
{
    "externalId": "123externalId",
    "name": "hbo Bachelor Bedrijfskunde 987",
    "providerId": "8a6eeaf9-0c1f-4426-a06d-110e613b9416",
    "learningMethod": {
        "type": "SelfPlanned",
        "costs": {
            "details":
            [
                {
                "invoiceItems": [
                    {
                        "priceBeforeTax": 200,
                        "vatAmount": 42,
                        "vatCountry": null,
                        "vatPercentage": null,
                        "vatExemptAmount": 211,
                        "vatExemptAmountVAT": 0,
                        "costType": "accommodation"
                    },
                    {
                        "priceBeforeTax": 8490,
                        "vatAmount": 1782.9,
                        "vatCountry": null,
                        "vatPercentage": null,
                        "vatExemptAmount": 9290,
                        "vatExemptAmountVAT": 0,
                        "costType": "lunch"
                    }
                ]
            }
        ]
        },
        "duration": {
            "value": 36.5,
            "unit": "Days"
        }
    }
}
```

How to edit a course using the PUT request

The PUT request is technically more simple than the PATCH when it comes to updating collections or many properties at once.

1. Perform a GET detail for that specific course id
2. Copy the response and paste it on the PUT endpoint
3. Change the desired property and perform the PUT request and you should have edited the course successfully

How to edit a course using the PATCH request

In EduMS we follow the JsonPatchDocument format, JSON Patch | jsonpatch.com. The PATCH request is more suitable for updating only one property instead of the whole document.

An example on how to update the course name:

```
[
  {
    "value": "Updated course name",
    "path": "name",
    "op": "replace"
  }
]
```

What happens when you delete a course

When you delete a course using the only endpoint provided in APIM we soft delete this course, meaning, you can still see it in the get detail and get list endpoints but with a property set to true, the deleted property.

We also update the course index, delete related assorted products and send out webhook notifications. You can also find courses in the course index (ProviderAPI - Courses Index) and in the assorted product index which is in Account API (Account API - Assorted Products Index).

The following have been deprecated

- status
- categoryCode
- categoryCodeDisplayValue
- domainCode
- domainCodeDisplayValue
- domainCodeRijksoverheid
- domainCodeRijksoverheidDisplayValue

How to add SubSegments to the course

You can only post SubSegments that belong to a Segment

1. Perform a GET list for segments (ProviderAPI - Segments List)
2. Save the segment id
3. Perform a GET list for subSegments (ProviderAPI - SubSegments List) with the following filter

```
segmentId eq 3716a4d4-3fd9-42ab-b2af-018148b9dbef
```

4. Save the subSegments name
5. Perform a PUT request with the segments and subSegments. An example below:

```
{
    "id": "c7bbd01e-be37-49d0-9693-1322deb48389",
    "externalId": "sampleExternalId",
    "internalId": null,
    "name": "sample name",
    "shortName": null,
    "edudexProgramUrl": null,
    "status": "New",
    "statusDisplayValue": "Nieuw",
    "source": "Manual",
    "sourceDisplayValue": null,
    "categoryCode": null,
    "categoryCodeDisplayValue": null,
    "domainCode": null,
    "domainCodeDisplayValue": null,
    "domainCodeRijksoverheid": null,
    "domainCodeRijksoverheidDisplayValue": null,
    "educationLevelCode": null,
    "educationLevelCodeDisplayValue": null,
        "segments": [{
                "value": "Communication",
                "subSegments": [{
                        "value": "Comm_CommunicationAndDesign"
                    },
                    {
                        "value": "Comm_CommunicationTraining"
                    }
                ]
            },
            {
                "value": "AutomationAndIT",
                "subSegments": [{
                    "value": "AutomIT_Training"
                }]
            }
        ],
    "providerId": "35f51e37-776f-4a53-96cb-f1d7acaa9b12",
    "learningMethod": {
        "id": "5362a72d-f379-41b4-ae8e-6092b38f29c7",
        "type": "SelfPlanned",
        "typeDisplayValue": "Individueel georganiseerd",
        "format": null,
        "formatDisplayValue": null,
        "language": null,
        "languageMaterial": null,
        "studyLoad": null,
        "contactSessions": null,
        "costs": {
            "id": "c4e98561-0925-48ad-a8d4-36e3ff8d69e5",
            "details": [{
                "id": "54a2acb5-1651-44db-820f-c3a5c9d6ba6e",
                "startDate": "2022-01-05T00:00:00.0000000Z",
                "endDate": null,
                "currencyCode": "Euro",
                "currencyCodeDisplayValue": null,
                "invoiceItems": [{
                        "id": "f30e0b66-5f53-4638-9b89-03081f81869b",
                        "quantity": 1,
                        "priceBeforeTax": 200,
                        "vatAmount": 42,
                        "vatCountry": null,
                        "vatPercentage": null,
                        "vatExemptAmount": 211,
                        "vatExemptAmountVAT": 0,
                        "costType": "Accommodation",
                        "costTypeDisplayValue": "Accommodatie",
                        "isOptional": null
                    },
                    {
                        "id": "7081363a-8d06-425e-9f85-8722b8eaed6a",
                        "quantity": 1,
                        "priceBeforeTax": 8490,
                        "vatAmount": 1782.9,
                        "vatCountry": null,
                        "vatPercentage": null,
                        "vatExemptAmount": 9290,
                        "vatExemptAmountVAT": 0,
                        "costType": "Lunch",
                        "costTypeDisplayValue": "Arrangementen (Lunch)",
                        "isOptional": null
                    }
                ]
            }]
        },
        "duration": {
            "value": "36.5",
            "unit": "Days",
            "unitDisplayValue": "Dagen"
        },
        "planning": null
    },
    "diplomaType": null,
    "diplomaTypeDisplayValue": null,
    "crebo": null,
    "croho": null,
    "content": [],
    "embedded": null,
    "startDate": null,
    "endDate": null,
    "commercialPartner": "BrandOwner",
    "commercialPartnerDisplayValue": null,
    "credit": null,
    "requirements": null,
    "targetSector": [],
            "targetSectorRijksoverheid": [],
            "providerLmsLink": null,
            "competences": [],
            "subsidies": [],
            "competencesRijksoverheid": [],
            "accreditation": null,
            "productType": "Regular",
            "productTypeDisplayValue": "Open aanbod",
            "isProfessionalCourse": false,
            "searchword": [],
            "deleted": false,
            "dateDeleted": null
        }
```

How to create a course from an existing one

1. Find Provider API and perform a GET list request
2. Select a course which is not deleted and save its id
3. Perform a POST request and the course should be created successfully