STUDYTUBE

API manual for providers

This manual will guide you through the process of setting up and using the Studytube API. It includes essential information on authentication, available endpoints, and best practices to ensure a smooth integration.

Table of Contents

1. Overview

The Studytube API provides a robust way for providers to access and manage product information on the Studytube platform. It is designed to be developer-friendly and follows modern RESTful design principles.

Key Links:

- **Product API Overview:** Studytube Product API
- **API Documentation:** Provider API Docs v2
- **Access Token URL:** https://backend.studytube.nl/gateway/oauth/token
- **Public API Base URL:** https://public-api.studytube.nl/provider-api/v2/

---

2. Getting Started

Before diving into the API, ensure you have the following prerequisites:

- **Provider Account:** You need to be registered as a provider with Studytube.
- **Request API credentials:** Send an email to implementatie@studytube.nl with a request for Provider API credentials (client ID and client secret) from Studytube.
- **Development Environment:** Set up your development environment with your preferred programming language and tools (e.g., Postman, curl, or any HTTP client library).

---

3. Authentication

The API uses OAuth for authentication. Here's how to obtain an access token:

1. **Endpoint:**
   Send a POST request to the access token URL:
   `https://backend.studytube.nl/gateway/oauth/token`

2. **Parameters:**
   Include the necessary parameters, such as:

   - `client_id`
   - `client_secret`
   - `grant_type` (commonly `"client_credentials"`)

3. **Example Request (using curl):**

```
Unset
curl -X POST \
  https://backend.studytube.nl/gateway/oauth/token \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d
'client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET&gra
nt_type=client_credentials'
```

4. **Response:**
   A successful request will return a JSON response with an `access_token` that you must include in the header of your subsequent API requests.

---

4. API Endpoints and Base URL

All API endpoints are built upon the Public API Base URL:

- **Base URL:**
  `https://public-api.studytube.nl/provider-api/v2/`

When making requests, append the specific endpoint to this base URL. For example, if you want to access a list of products, your URL might look like:

```
Unset
https://public-api.studytube.nl/provider-api/v2/products
```

See the Provider API Documentation for a detailed list of available endpoints, request parameters, and response structures.

---

5. Making Your First API Call

Here's a step-by-step workflow for a basic API call:

1. **Obtain an Access Token:**
   Follow the instructions in the Authentication section to get your token.

2. **Set Up the Request:**
   Use your preferred tool (e.g., Postman, curl) to configure your API call.

   - **Method:** GET (or POST/PUT/DELETE as required by the endpoint)
   - **URL:** Use the Public API Base URL plus the specific endpoint (e.g., `/products`).
   - **Headers:** Include `Authorization: Bearer YOUR_ACCESS_TOKEN`.
3. **Example Request (using curl):**

```
Unset
curl -X GET \
  'https://public-api.studytube.nl/provider-api/v2/products' \
  -H 'Authorization: Bearer YOUR_ACCESS_TOKEN'
```

4. **Review the Response:**
   The API will return a JSON response with the requested data. Use the interactive documentation to interpret the data fields and format.

---

6. Error Handling and Best Practices

To ensure a robust integration, consider the following:

- **HTTP Status Codes:**
  Familiarize yourself with standard HTTP status codes. For example:

  - `200 OK` – Successful call
  - `401 Unauthorized` – Authentication failure (check your access token)
  - `404 Not Found` – The requested resource does not exist
  - `500 Internal Server Error` – A server error occurred (retry or contact support)
- **Rate Limiting:**
  Check the API documentation for any rate limiting policies. Avoid sending excessive requests in a short time period.

- **Error Messages:**
  Always inspect the error messages in the response body. They typically contain

details that help in diagnosing the issue.

- **Logging:**
  Implement logging in your application to capture both successful and failed API calls for easier troubleshooting.

- **Security:**
  Never expose your client credentials or access tokens. Securely store them and ensure all API calls are made over HTTPS.

---

7. Explanation of calls

Classroom Trainings Endpoints

These endpoints are focused on managing classroom training sessions. Providers use them to create, retrieve, update, or delete details of training sessions that are held in a classroom setting. Key points include:

- **Listing Trainings:**
  A **GET** request to `/classroom-trainings` typically returns a list of all classroom training sessions. You can often filter this list by parameters such as training date, status, or location. This is useful for providers to review upcoming or past training events.

- **Retrieving a Specific Training:**
  A **GET** request to `/classroom-trainings/{id}` is used to fetch detailed information about a single training session, identified by its unique ID. The response usually includes data like training title, description, schedule, location, available slots, and any additional metadata.

- **Creating a Training Session:**
  A **POST** request to `/classroom-trainings` allows a provider to create a new classroom training session. The body of the request should include all the necessary details (such as training name, description, dates, and other relevant parameters). The API expects a well-structured JSON payload following the schema described in the documentation.

- **Updating a Training Session:**
  A **PUT** or **PATCH** request to `/classroom-trainings/{id}` lets you update details

of an existing training. This call is used when you need to change information such as the training time, location, or content.

- **Deleting a Training Session:**
  A **DELETE** request to `/classroom-trainings/{id}` is available if you need to remove a training session from the system. This call usually requires careful handling since it will permanently delete the session.

User Training Requests Endpoints

This set of endpoints deals with requests made by users related to training sessions. They enable providers to monitor and manage user interest or enrollment requests. Here are the key aspects:

- **Listing Training Requests:**
  A **GET** request to `/user-training-requests` returns a list of training requests submitted by users. These requests might include information such as the training session the user is interested in, user details, and the current status of the request (e.g., pending, approved, or rejected).

- **Retrieving a Specific Request:**
  A **GET** request to `/user-training-requests/{id}` is used to fetch detailed information about a specific training request. This call returns data including the request ID, user details, associated training session, and any comments or additional data provided by the user.

- **Managing Requests:**
  Although many systems offer a simple read-only view of these requests, some APIs allow additional actions such as updating the status of a request (for example, marking it as approved or declined). If such functionality is available, you might see a **PUT** or **PATCH** request to `/user-training-requests/{id}` where the provider can change the status or add notes.

- **Creation of Requests:**
  In some systems, there might be a **POST** endpoint for creating a user training request. However, in a provider-focused API, this endpoint is more likely used by the front-end or another service that collects user input rather than by the provider directly.