**Отчет об исследовании логики создания ссылок на встречи в Yandex Телемост для автоматизации**

# Введение

Целью данного исследования является анализ логики создания ссылок на встречи в Yandex Телемост с целью разработки бота, который сможет автоматически генерировать такие ссылки. Основное внимание уделяется возможностям, предоставляемым официальным API Yandex Телемост.

# Доступ к API Yandex Телемост

Яндекс Телемост авторизует приложения с помощью OAuth-токенов. Для работы с API необходимо:

1. **Зарегистрировать OAuth-приложение:**
   * Открыть страницу создания приложения.
   * Указать название сервиса и прикрепить иконку (при необходимости).
   * Выбрать платформы, на которых будет работать приложение, и указать их параметры.
   * В разделе **Доступ к данным** указать необходимые права доступа:
   * telemost-api:conferences.create — для создания встреч и трансляций.
   * telemost-api:conferences.read — для просмотра данных о встречах и трансляциях.
   * telemost-api:conferences.update — для редактирования данных о встречах и трансляциях.
   * Указать электронную почту для связи.
   * Нажать **Создать приложение** и скопировать **ClientID**.
2. **Получить OAuth-токен:**
   * Для отладочного токена перейти по ссылке: https://oauth.yandex.ru/authorize? response\_type=token&client\_id=<идентификатор\_приложения> (заменив <идентификатор\_приложения> на полученный ClientID).
   * Авторизоваться на Яндексе.

Полученный токен необходимо передавать в HTTP-заголовке Authorization при каждом вызове API в формате: --header 'Authorization: OAuth <OAuth-токен>'

**Ограничение:** Работа с API Телемоста доступна только для пользователей Яндекс 360 для бизнеса с аккаунтами на домене организации.

# Создание видеовстречи в Yandex Телемост через API

Для создания видеовстречи или трансляции в Телемосте приложению требуется разрешение telemost-api:conferences.create .

## Запрос

**Метод:** POST

**URL:** https://cloud-api.yandex.net/v1/telemost-api/conferences

## Заголовки

* Authorization: OAuth <токен>
* Content-Type: application/json

## Тело запроса (JSON)

JSON

{

"access\_level"

:

"string"

,

*// deprecated, используется для обратной*

*совместимости*

"waiting\_room\_level"

:

"string"

,

"live\_stream"

:

{

"access\_level"

:

"string"

,

"title"

:

"string"

,

"description"

:

"string"

}

,

"cohosts"

:

[

{

"email"

:

"string"

}

]

}

**Поля тела запроса:**

* access\_level (string, необязательно, deprecated): В предыдущих версиях использовался для настройки ограничения доступа. Игнорируется, если передан

waiting\_room\_level .

* PUBLIC — комната ожидания выключена.
* ORGANIZATION — комната ожидания включена для внешних пользователей.
* waiting\_room\_level (string, необязательно): Настройки комнаты ожидания.
* PUBLIC — комната ожидания выключена.
* ORGANIZATION — комната ожидания включена для внешних пользователей.
* ADMINS — комната ожидания включена для всех пользователей, кроме организаторов и соорганизаторов.
* live\_stream (object, необязательно): Параметры трансляции. Если не указаны, трансляция не создается.
* live\_stream.access\_level (string, необязательно): Настройки ограничения доступа для трансляции.
* PUBLIC — для всех пользователей.
* ORGANIZATION — только для сотрудников.
* live\_stream.title (string, необязательно): Название трансляции (макс. 1024 символа).
* live\_stream.description (string, необязательно): Описание трансляции (макс. 2048 символов).
* cohosts (array of object, необязательно): Список соорганизаторов встречи (макс. 30 элементов).
* cohosts.email (string, необязательно): Адрес электронной почты соорганизатора. Только пользователи с аккаунтом на Яндексе.

**Примеры запросов:**

**Видеовстреча без трансляции (публичная):**

JSON

{

"waiting\_room\_level"

:

"PUBLIC"

}

**Видеовстреча для сотрудников организации с трансляцией для всех пользователей:**

JSON

{

"waiting\_room\_level"

:

"ADMIN"

,

"live\_stream"

:

{

"access\_level"

:

"PUBLIC"

,

"title"

:

"Example conference created via API"

,

"description"

:

"Some description of example conference created via API"

}

}

**Видеовстреча с двумя соорганизаторами (публичная):**

JSON

{

"waiting\_room\_level"

:

"PUBLIC"

,

"cohosts"

:

[

{

"email"

:

"user1@yandex.ru"

}

,

{

"email"

:

"user2@org-domain.ru"

}

]

}

## Результат

При

успешном

запросе

возвращается

ответ

с

кодом

201

Created

и

JSON-

телом

,

содержащим

информацию

о

созданной

видеовстрече

:

JSON

{

"id"

:

"string"

,

"join\_url"

:

"string"

,

"live\_stream"

:

{

"watch\_url"

:

"string"

}

}

**Поля успешного ответа:**

* id (string): Идентификатор видеовстречи.
* join\_url (string): Ссылка для участников для присоединения к видеовстрече.
* live\_stream (object, необязательно): Данные о трансляции. Отсутствует, если встреча без трансляции.
* live\_stream.watch\_url (string): Ссылка для зрителей для просмотра трансляции.

**Примеры успешных ответов:**

**Видеовстреча без трансляции:**

JSON

{

"id"

:

"12345678901234"

,

"join\_url"

:

"https://telemost.yandex.ru/j/12345678901234"

}

**Видеовстреча с трансляцией:**

JSON

{

"id"

:

"12345678901234"

,

"join\_url"

:

"https://telemost.yandex.ru/j/12345678901234"

,

"live\_stream"

:

{

"watch\_url"

:

"https://telemost.yandex.ru/live/123456789abcdef0123456789abcdef0"

}

}

## Ответ с ошибкой

В случае ошибки возвращается сообщение следующего содержания:

JSON

{

"error"

:

"string"

,

"message"

:

"string"

,

"description"

:

"string"

,

"details"

:

"object"

}

# Методы автоматизации создания ссылок на встречи

На основе анализа документации API Яндекс Телемоста, основным методом автоматизации создания ссылок на встречи является использование REST API.

**Ключевые аспекты автоматизации:**

1. **Авторизация (OAuth 2.0):**
   * Для взаимодействия с API требуется получение OAuth-токена. Это включает регистрацию OAuth-приложения в Яндекс и получение ClientID .
   * Затем необходимо получить сам OAuth-токен, который будет использоваться для авторизации запросов к API.
   * Токен передается в HTTP-заголовке Authorization в формате OAuth <токен> .
2. **Создание видеовстречи через API:**
   * **Метод:** POST запрос.
   * **URL:** https://cloud-api.yandex.net/v1/telemost-api/conferences .
   * **Заголовки:** Authorization: OAuth <токен> и Content-Type: application/json .
   * **Тело запроса (JSON):** Позволяет настроить параметры встречи, такие как уровень комнаты ожидания ( waiting\_room\_level ), параметры трансляции

( live\_stream с заголовком и описанием) и список соорганизаторов ( cohosts ).

* + Пример минимального запроса для публичной встречи без трансляции:
  + **Разрешения:** Приложение должно иметь разрешение telemostapi:conferences.create .

1. **Получение ссылки на встречу:**
   * При успешном создании встречи API возвращает HTTP-статус 201 Created и JSON-объект, содержащий id встречи и join\_url (ссылку для присоединения участников).
   * Если была создана трансляция, также будет возвращена watch\_url .

**Общая логика работы бота:**

1. **Инициализация:** Бот должен быть настроен с ClientID Яндекс OAuth-приложения.
2. **Авторизация:** Бот должен иметь механизм для получения и обновления OAuthтокена. Это может быть ручное получение токена и его хранение, или автоматизированный процесс, если Яндекс предоставляет такие возможности для серверных приложений.
3. **Формирование запроса:** На основе пользовательских данных (например, название встречи, участники, необходимость трансляции) бот формирует JSONтело запроса.
4. **Отправка запроса:** Бот отправляет POST запрос к API Яндекс Телемоста с соответствующими заголовками и телом.
5. **Обработка ответа:** Бот парсит ответ от API, извлекает join\_url и предоставляет ее пользователю.
6. **Обработка ошибок:** Бот должен уметь обрабатывать ошибки, возвращаемые API (например, неверные разрешения, некорректные параметры запроса).

Таким образом, автоматизация создания ссылок на встречи в Яндекс Телемост полностью реализуема через его REST API, при условии корректной OAuthавторизации и формирования запросов.