

Инструкция по интеграции с Wahelp API

Этот документ описывает шаги для интеграции с платформой **Wahelp** и отправки первого сообщения через WhatsApp или Telegram. Информация основана на примере из Swagger-документации API.

Цель

Наша цель — отправить первое сообщение пользователю через WhatsApp или Telegram, используя Wahelp API. Для этого потребуется выполнить несколько последовательных действий: получить токен доступа, создать проект, подключить канал, создать (или импортировать) пользователя и отправить сообщение. Дополнительно можно настроить получение истории сообщений и веб-хуков.

1. Авторизация

Wahelp использует OAuth-подобный механизм для выдачи токена доступа. Без токена выполнение запросов API невозможно.

Метод:

POST /api/app/user/login
Content-Type: application/json

```
{
  "login": "your_login",
  "password": "your_password"
}
```

Ответ:

```
{
  "success": true,
  "data": {
    "token_type": "bearer",
    "access_token": "<token>",
    "expires_in": 3600
  }
}
```

Значение access_token — ваш API-токен. В каждом последующем запросе используйте заголовок:

Authorization: Bearer <token>

Для части методов потребуется заголовок X-Project с ID проекта, который мы получим на следующем шаге.

2. Создание проекта

Проект объединяет каналы и пользователей. Для каждого нового проекта необходимо задать название.

Метод:

POST /api/app/projects
Authorization: Bearer <token>
Content-Type: application/json

```
{  
  "name": "My First Project"  
}
```

Ответ:

```
{  
  "success": true,  
  "data": {  
    "id": 101,  
    "name": "My First Project"  
  }  
}
```

ID проекта (id) необходимо указывать в заголовке X-Project при вызове методов, связанных с каналами, пользователями и отправкой сообщений.

3. Подключение канала

Канал — это источник, через который отправляются сообщения (например, WhatsApp или Telegram). На платформе можно подключить несколько каналов к одному проекту.

Создание канала:

POST /app/projects/{projectId}/channels
Authorization: Bearer <token>
X-Project: {projectId}
Content-Type: application/json

```
{  
  "name": "My WhatsApp Channel",  
  "type": "whatsapp"  
}
```

Ответ:

```
{  
  "success": true,  
  "data": {  
    "id": 201,  
    "name": "My WhatsApp Channel",  
    "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
    "active": false,  
    "type": "whatsapp",  
    "settings": {}  
  }  
}
```

```
}  
}
```

После создания канала необходимо пройти авторизацию канала. Для WhatsApp это обычно сканирование QR-кода (аналогично подключению WhatsApp Web). После успешной авторизации поле `active` станет `true`.

4. Создание или импорт пользователя

В Wahelp пользователь — это конечный абонент, которому вы отправляете сообщения. Пользователя можно создать вручную, передав его номер телефона и имя. Если пользователь уже существует, можно использовать его ID.

Создание пользователя:

```
POST /api/app/projects/{projectId}/channels/{channel_uuid}/users  
Authorization: Bearer <token>  
X-Project: {projectId}  
Content-Type: application/json
```

```
{  
  "name": "John Doe",  
  "phone": "+1234567890"  
}
```

Ответ:

```
{  
  "success": true,  
  "data": {  
    "id": 301,  
    "name": "John Doe",  
    "phone": "+1234567890"  
  }  
}
```

Если в ответ приходит ошибка «у номера нет нужного мессенджера», значит, по указанному телефону не удалось найти аккаунт WhatsApp/Telegram.

5. Получение истории сообщений (опционально)

Для просмотра диалога с пользователем можно запросить историю сообщений.

Метод:

```
GET /api/app/projects/{projectId}/channels/{channel_id}/messages  
Authorization: Bearer <token>  
X-Project: {projectId}
```

Результатом будет список сообщений с временными метками и статусами доставки. Этот шаг опционален и чаще используется в CRM-системах.

6. Отправка сообщения

Чтобы отправить текстовое сообщение пользователю, используйте метод `send_message`.

```
POST /api/app/projects/{projectId}/channels/{channel_uuid}/send_message/{user_id}
Authorization: Bearer <token>
X-Project: {projectId}
Content-Type: application/json
```

```
{
  "text": "Привет, это тестовое сообщение!"
}
```

Ответ содержит объект сообщения с его ID, текстом и статусом отправки. Для отправки файлов и шаблонов существуют дополнительные параметры.

7. Настройка веб-хуков (опционально)

Если вам нужно получать входящие сообщения и статусы доставки, настройте веб-хук:

```
PUT /api/app/projects/{projectId}/hook
Authorization: Bearer <token>
X-Project: {projectId}
Content-Type: application/json
```

```
{
  "url": "https://example.com/webhook"
}
```

После настройки Wahelp будет отправлять POST-запросы на указанный URL при поступлении новых сообщений или изменении статуса доставки.

8. Минимальный рабочий сценарий («Hello World»)

Для отправки первого сообщения достаточно выполнить следующие шаги:

1. **Авторизация.** Запрос `POST /api/app/user/login` — получить `access_token`.
2. **Создание проекта.** Запрос `POST /api/app/projects` — получить `projectId`.
3. **Создание канала.** Запрос `POST /api/app/projects/{projectId}/channels` — получить `channel_uuid` и авторизовать канал (сканировать QR-код).
4. **Создание пользователя.** Запрос `POST /api/app/projects/{projectId}/channels/{channel_uuid}/users` — получить `user_id`.
5. **Отправка сообщения.** Запрос `POST /api/app/projects/{projectId}/channels/{channel_uuid}/send_message/{user_id}` с текстом сообщения.

Следуя этим шагам, вы сможете отправить первое сообщение через Wahelp API. Далее можно реализовать обработку входящих сообщений, интеграцию со своей системой и автоматизацию диалогов.