



Общество с ограниченной ответственностью
«Эндоскопическая видеоаналитика Лаб»

Руководство администратора

Polyptron.AI

Установка клиента	3
Минимальные системные требования	3
Подготовка системы	3
Настройка среды выполнения	4
Установка Polytron.AI	4
Настройка клиента	5
Соединение с сервером	5
Запуск клиента	5
Аргументы командной строки	7
Установка сервера	8
Минимальные системные требования	8
Подготовка системы	8
Распаковка файлов сервера	9
Запуск сервера	9
Настройка сервера	11
Подключение клиента	11
Создание почтовой рассылки	12

Установка клиента

Минимальные системные требования

- **ОС:** Ubuntu 20.04 и выше
- **ЦП:** 4-х ядерный процессор с тактовой частотой не менее 3 ГГц, например AMD Ryzen 3 3200G или аналог от Intel
- **ОЗУ:** 16 Гб
- **ГП:** NVidia RTX 2080 или выше, ГП от AMD не поддерживаются
- **Накопитель:** HDD или SSD со свободным пространством не менее 20 Гб

Подготовка системы

Перед установкой клиентского приложения Polytron.AI необходимо установить в систему ряд зависимостей, в числе которых интерпретатор для выполнения программ написанных на языке программирования Python и ряд инструментов для разработки(система контроля версий, набор компиляторов и библиотек). Для этого последовательно выполните в окне терминала следующие команды:

```
apt-get update
```

```
DEBIAN_FRONTEND=noninteractive apt-get install -y build-essential git python3.9-full  
wget libxml2 libglib2.0-0
```

Далее следует установить пакет программ CUDA Toolkit от компании NVidia. Следующий набор команд скачает и запустит установщик:

```
wget  
https://developer.download.nvidia.com/compute/cuda/12.0.0/local\_installers/cuda\_12.0.0\_525.60.13\_linux.run
```

```
sh cuda_12.0.0_525.60.13_linux.run --toolkit --silent
```

или если в системе не установлены драйвера от NVidia:

```
sh cuda_12.0.0_525.60.13_linux.run --driver --toolkit --silent
```

Перезагрузите компьютер.

Настройка среды выполнения

После перезапуска необходимо создать и настроить виртуальную среду для выполнения программ на языке Python. Выполните в терминале следующую команду:

```
python3.9 -m venv polyptron-venv
```

Где **polyptron-venv** - директория, в которой будут сохранены файлы виртуальной среды.

После создания виртуальной среды можно приступить к её настройке. Активируйте среду, обновите пакетный менеджер pip и установите пакеты, которые необходимы для работы клиентского приложения Polyptron.AI. Для этого выполните следующие команды:

```
source polyptron-venv/bin/activate
```

```
pip install --upgrade pip
```

```
pip install packaging nvidia-pyindex nvidia-tensorrt torch torchvision --extra-index-url  
https://download.pytorch.org/whl/cu116
```

Установка Polyptron.AI

После того как в систему были установлены зависимости и настроена среда выполнения, можно приступить к установке пакетов клиентского приложения Polyptron.AI. Если по каким-то причинам виртуальная среда Python была деактивирована, её необходимо активировать повторно командой:

```
source polyptron-venv/bin/activate
```

После этого установите Polyptron.AI:

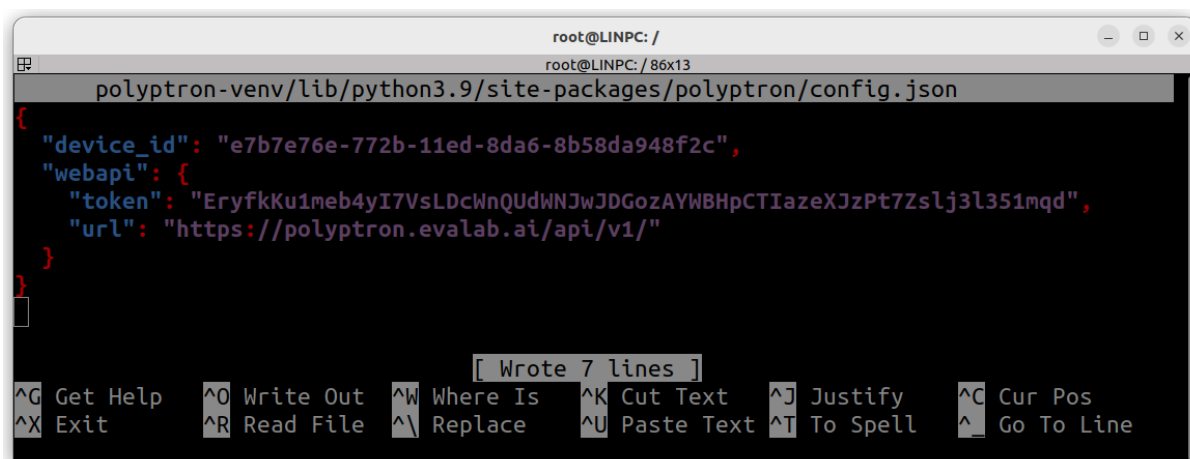
```
pip install polyptron-ml-1.0.0.tar.gz
```

```
pip install polytron-1.0.0.tar.gz
```

Настройка клиента

Соединение с сервером

Для того чтобы клиентское приложение могло отправлять данные на сервер необходимо настроить сетевое соединение. Для этого откройте файл конфигурации `<venv-dir>/lib/python3.9/site-packages/polyptron/config.json`, где `<venv-dir>` - корневая директория файлов виртуальной среды созданной ранее. В файле конфигурации укажите три значения: идентификатор устройства - `device_id`, секретный ключ - `token`¹ и точку доступа к серверному api - параметр `url`, который должен иметь вид `[http|https]://<host>:<port>/api/v1/`, где `<host>` - доменное имя, или IP адрес сервера. Порт указывать не обязательно. Сохраните изменения. При следующем запуске клиентское приложение автоматически установит соединение с сервером. С этого момента данные, полученные во время проведения колоноскопических процедур, начнут отправляться на сервер. Пример корректно заполненного файла конфигурации изображён на рисунке 1.



```
root@LINPC: /
root@LINPC: /86x13
polyptron-venv/lib/python3.9/site-packages/polyptron/config.json
{
  "device_id": "e7b7e76e-772b-11ed-8da6-8b58da948f2c",
  "webapi": {
    "token": "EryfkKu1meb4yI7VsLDcWnQUdWNJwJDGoZAYWBHпCTIazeXJzPt7Zslj3l351nqd",
    "url": "https://polyptron.evalab.ai/api/v1/"
  }
}
[ Wrote 7 lines ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^_ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Рисунок 1 - Файл конфигурации клиентского приложения Polyptron.AI.

Запуск клиента

Для запуска клиентского приложения Polyptron.AI необходимо в начале активировать виртуальную среду для выполнения программ на языке Python. Виртуальная среда должна быть создана в момент установки клиентского приложения Polyptron.AI на компьютер(см. руководство администратора Polyptron.AI).

¹ О том как получить идентификатор устройства и секретный ключ смотрите раздел [Подключение клиента](#).

Для активации среды выполните в терминале следующую команду:

```
source <venv-dir>/bin/activate
```

Где **<venv-dir>** - корневая директория файлов виртуальной среды.

После этого можно запускать клиентское приложение Polyrtron.AI. Для запуска выполните команду:

```
python -m polytron.app --source 0
```

В этом режиме Polyrtron.AI стартует с параметрами по-умолчанию, рисунок 2.

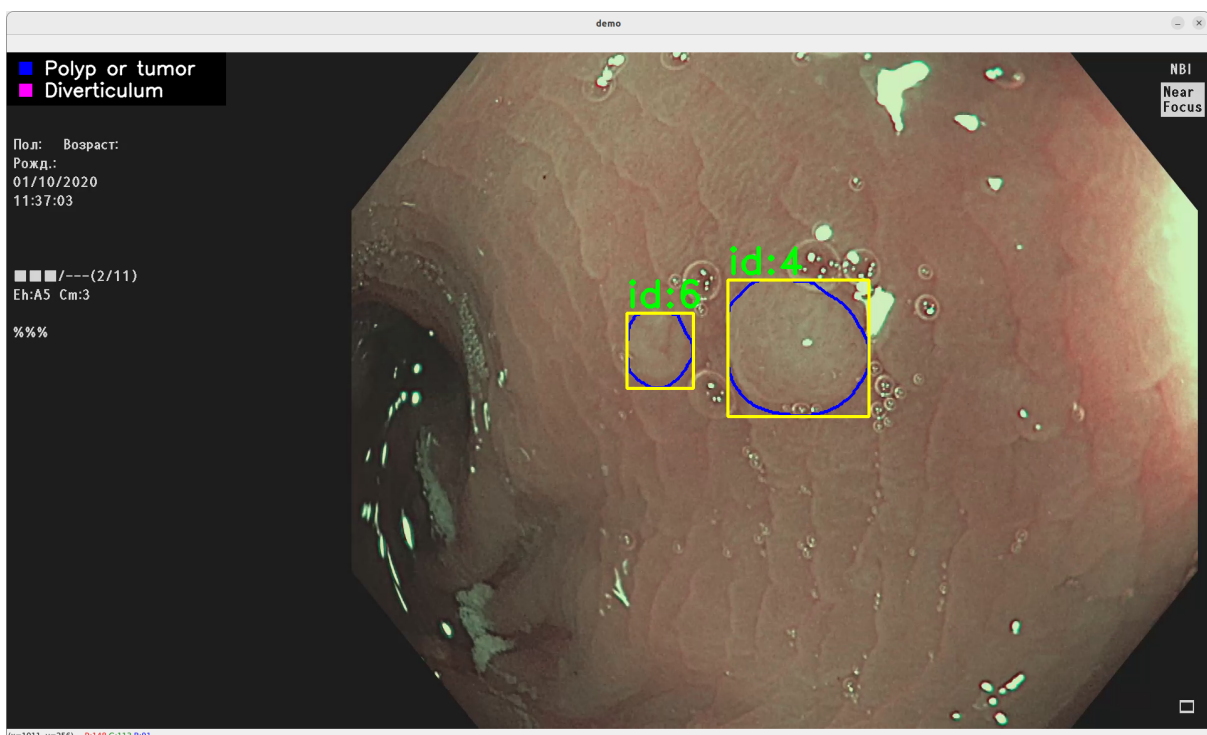


Рисунок 2 - Изображение главного окна клиентского приложения Polyrtron.AI.

На рисунке 2 представлено главное окно приложения, в котором выводится текущий кадр из видеопотока. На кадре рисуются:

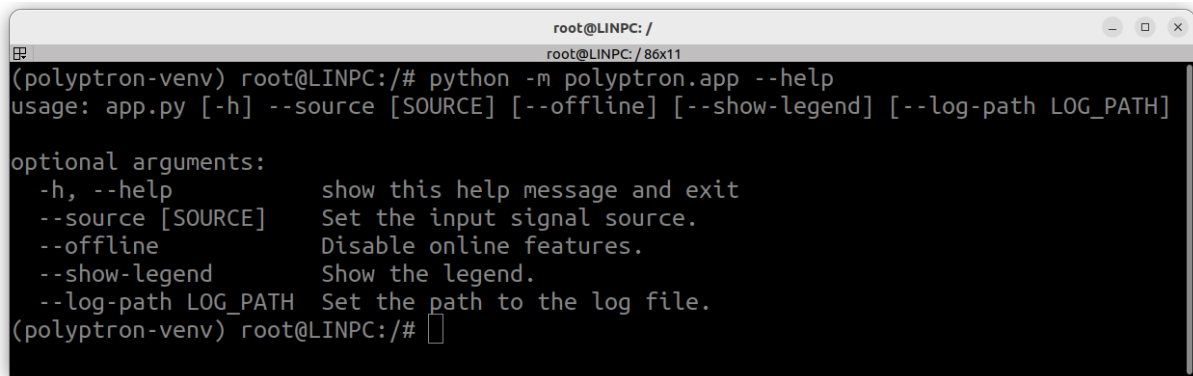
- Легенда, на которой приведено соответствие цвета обводки и класса новообразования;
- Ограничивающие рамки для новообразования;
- Обводка новообразования соответствующим ему цветом;
- Уникальный, в рамках процедуры, идентификатор новообразования.

Аргументы командной строки

Для получения справки выполните команду:

```
python -m polyptron.app --help
```

Результат выполнения команды на рисунке 3.



```
root@LINPC: /
root@LINPC: /86x11
(polytron-venv) root@LINPC:/# python -m polyptron.app --help
usage: app.py [-h] --source [SOURCE] [--offline] [--show-legend] [--log-path LOG_PATH]

optional arguments:
  -h, --help            show this help message and exit
  --source [SOURCE]    Set the input signal source.
  --offline             Disable online features.
  --show-legend        Show the legend.
  --log-path LOG_PATH  Set the path to the log file.
(polytron-venv) root@LINPC:/#
```

Рисунок 3 - Вывод справки клиентского приложения Polytron.AI.

При запуске клиентского приложения Polytron.AI вы можете использовать следующие аргументы командной строки:

- **source** - Задаёт источник входного сигнала. Это может быть видео файл, или любой другой источник сигнала подключенный к системе. В случае видео файла source принимает абсолютный путь до файла. В случае с иным источником сигнала, например эндоскопом, подключённым через плату захвата, source принимает целое беззнаковое число - порядковой номер устройства N в файлах /dev/video{N};
- **offline** - Отключает сетевые функции. В этом режиме данные о найденных патологиях не отправляются на сервер, а клиентское приложение только визуально отмечает новообразования;
- **show-legend** - Визуализирует легенду;
- **log-path** - Перенаправляет вывод лога программы в файл. Принимает путь до файла.

Установка сервера

Минимальные системные требования

- **ОС:** Ubuntu 20.04 и выше
- **ЦП:** 2-х ядерный процессор с тактовой частотой не менее 3 ГГц
- **ОЗУ:** 4 Гб
- **Накопитель:** HDD или SSD со свободным пространством не менее 4 Тб

Подготовка системы

Сервер Polyrtron.AI и его зависимости поставляется в виде набора готовых образов Docker. Для запуска и управления используется набор инструментов, входящих в состав Docker Compose.

Перед установкой Docker необходимо установить в систему ряд зависимостей. Выполните в терминале следующие команды:

```
apt-get update
```

```
apt-get install -y ca-certificates curl gnupg lsb-release unzip
```

Далее следует установить в систему официальный GPG:

```
mkdir -p /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

После установке ключа можно подключать репозиторий с Docker:

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```


Следующими двумя командами обновите индекс пакетного менеджера apt и установите сам Docker с плагином Docker Compose:

```
apt-get update
```

```
apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Перезагрузите систему.

Распаковка файлов сервера

Распакуйте архив **polyptron-server.zip** в удобную для вас директорию, это можно сделать следующей командой:

```
unzip polyptron-server.zip -d polyptron-server
```

После распаковки архива в указанной директории должны появиться три файла(как на рисунке 4):

- **.env** - файл конфигурации docker compose с перечислением используемых переменных окружения
- **docker-compose.yaml** - основной файл конфигурации docker compose
- **polyptron-server.docker.tar** - docker образ сервера в виде tar архива

```
drwxr-xr-x 2 root    root          4096 фев  7 17:50 .
drwxr-xr-x 3 stranger stranger      4096 фев  7 17:50 ..
-rw-r--r-- 1 root    root          1121 фев  7 17:29 docker-compose.yaml
-rw-r--r-- 1 root    root           352 фев  7 15:23 .env
-rw----- 1 root    root      3052623360 фев  7 17:38 polyptron-server.docker.tar
```

Рисунок 4 - содержимое архива **polyptron-server.zip**.

Запуск сервера

Перед запуском сервера необходимо настроить переменные окружения, которые используются сервером и его зависимостями. Откройте файл **.env** в удобном для вас текстовом редакторе, укажите в нём следующие переменные:

- **BASE_URL** - Базовая часть для всех ссылок, которые будут выдаваться пользователю, например ссылка на протокол обследования в email уведомлениях. Значение переменной должно иметь вид **[http|https]://<host>**, где **<host>** - имя домена, по которому будет доступен сервер. Например, при локальном развёртывании значение переменной будет иметь вид **http://localhost**.

- **POSTGRES_DB** - Имя базы данных в СУБД PostgreSQL, которую будет использовать сервер.
- **POSTGRES_USER** - Логин пользователя в СУБД PostgreSQL, от имени которого сервер будет подключаться к базе данных.
- **POSTGRES_PASSWORD** - Пароль для базы данных
- **SMTP_SERVER** - Доменное имя или ip адрес smtp сервера, который будет использоваться для рассылки email уведомлений.
- **SMTP_PORT** - Порт smtp сервера.
- **SMTP_USER** - Имя пользователя для логина на smtp сервер.
- **SMTP_PASSWORD** - Пароль пользователя для логина на smtp сервер.
- **EMAIL_FROM** - Почтовый адрес, который будет подставлен в поле from для email уведомлений, рассылаемых сервером Polytron.AI.

Также вы можете настроить доступ к pgadmin. Этот шаг опционален. По-умолчанию web-интерфейс pgadmin доступен по адресу <http://localhost:8281>. Для получения доступа в файле .env следует указать логин и пароль.

- **PGADMIN_DEFAULT_EMAIL** - Email, который будет использоваться в качестве логина в web-интерфейс PGAdmin.
- **PGADMIN_DEFAULT_PASSWORD** - Пароль от web-интерфейса PGAdmin.

Далее необходимо восстановить docker образ сервера из tar архива, делается это с помощью команды:

```
docker load -i polytron-server.docker.tar
```

После всех настроек возможно запустить сервер. Выполните следующую команду в терминале:

```
docker compose up -d
```

Настройка сервера

Подключение клиента

Для того чтобы подключить клиента необходимо иметь идентификатор устройства и секретный ключ. Для их получения на сервере создаётся запись об организации и для неё регистрируется устройство. На одну организацию может быть зарегистрировано множество устройств. Для создания организации выполните в терминале следующую команду:

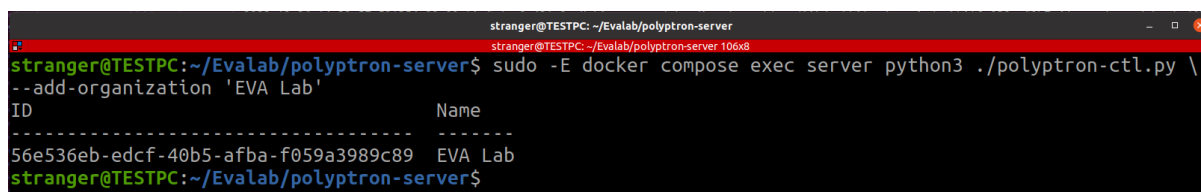
```
docker compose exec server python3 ./polytron-ctl.py --add-organization <org. name>
```

где **<org. name>** - название организации. При успешном выполнении команды в терминал будет выведена таблица с ID и названием организации (пример на рисунке 5).

После этого на организацию можно зарегистрировать устройство. Выполните в терминале:

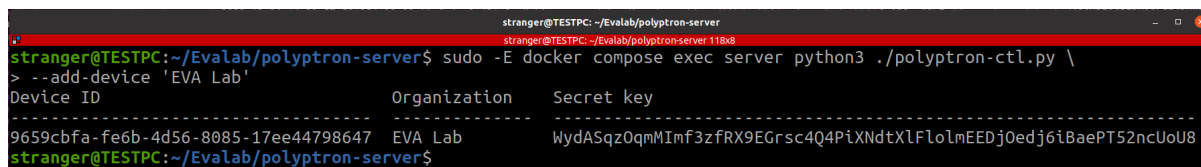
```
docker compose exec server python3 ./polytron-ctl.py --add-device <org. name>
```

где **<org. name>** - название организации, на которую регистрируется новое устройство. При успешном выполнении команды в терминал будет выведена таблица с идентификатором устройства, секретным ключом и названием организации, на которую зарегистрировано данное устройство (пример на рисунке 6).



```
stranger@TESTPC: ~/Evalab/polytron-server
stranger@TESTPC: ~/Evalab/polytron-server 106x8
stranger@TESTPC:~/Evalab/polytron-server$ sudo -E docker compose exec server python3 ./polytron-ctl.py \
--add-organization 'EVA Lab'
ID                                     Name
-----
56e536eb-edcf-40b5-afba-f059a3989c89  EVA Lab
stranger@TESTPC:~/Evalab/polytron-server$
```

Рисунок 5 - Добавление новой организации.



```
stranger@TESTPC: ~/Evalab/polytron-server
stranger@TESTPC: ~/Evalab/polytron-server 118x8
stranger@TESTPC:~/Evalab/polytron-server$ sudo -E docker compose exec server python3 ./polytron-ctl.py \
> --add-device 'EVA Lab'
Device ID                               Organization      Secret key
-----
9659cbfa-fe6b-4d56-8085-17ee44798647  EVA Lab          WydASqz0qmMImf3zfRX9EGrsc4Q4PiXNdtXLFloImEEDj0edj6iBaePT52ncUoU8
stranger@TESTPC:~/Evalab/polytron-server$
```

Рисунок 6 - Добавление нового устройства.

Удалить устройство можно командой:

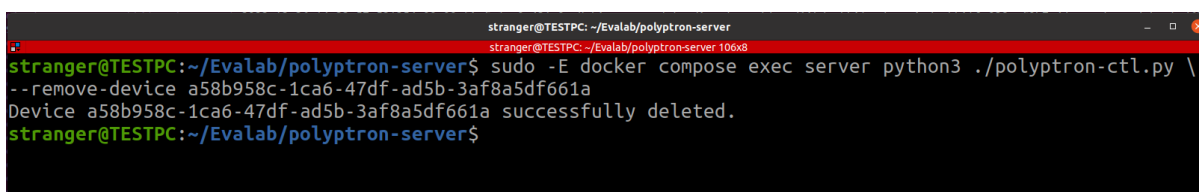
```
docker compose exec server python3 ./polytron-ctl.py --remove-device <device_id>
```

где **<device_id>** - идентификатор устройства, которое вы хотите удалить.

Аналогично удаляется организация:

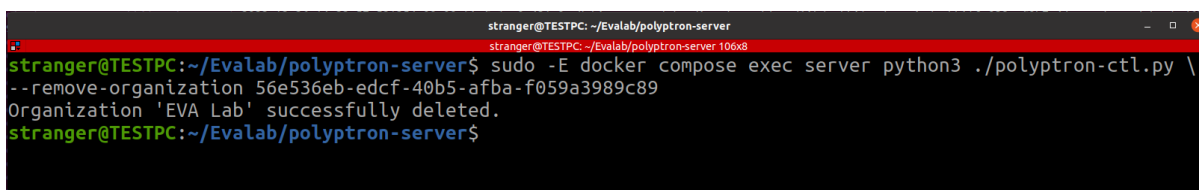
```
docker compose exec server python3 ./polytron-ctl.py --remove-organization <name | id>
```

где **<name | id>** - идентификатор или название организации, которую вы хотите удалить.



```
stranger@TESTPC:~/Evalab/polytron-server
stranger@TESTPC:~/Evalab/polytron-server 106x8
stranger@TESTPC:~/Evalab/polytron-server$ sudo -E docker compose exec server python3 ./polytron-ctl.py \
--remove-device a58b958c-1ca6-47df-ad5b-3af8a5df661a
Device a58b958c-1ca6-47df-ad5b-3af8a5df661a successfully deleted.
stranger@TESTPC:~/Evalab/polytron-server$
```

Рисунок 7 - Удаление устройства.



```
stranger@TESTPC:~/Evalab/polytron-server
stranger@TESTPC:~/Evalab/polytron-server 106x8
stranger@TESTPC:~/Evalab/polytron-server$ sudo -E docker compose exec server python3 ./polytron-ctl.py \
--remove-organization 56e536eb-edcf-40b5-afba-f059a3989c89
Organization 'EVA Lab' successfully deleted.
stranger@TESTPC:~/Evalab/polytron-server$
```

Рисунок 8 - Удаление организации.

Создание почтовой рассылки

Система Polytron.AI поддерживает рассылку уведомлений на электронную почту. Для добавления ящика электронной почты в список рассылки выполните в терминале следующие команды:

```
docker compose exec server python3 ./polytron-ctl.py --add-email <device_id> <email>
```

где **<device_id>** - идентификатор устройства, **<email>** - адрес электронного почтового ящика.

```
stranger@TESTPC: ~/Evalab/polypton-server
stranger@TESTPC:~/Evalab/polypton-server$ sudo -E docker compose exec server python3 ./polypton-ctl.py --add-email \
> 8c40285c-9f87-4f85-824b-24c3b7c0e3bb test@mail.com
Organization      Device ID          Email
-----
EVA Lab           8c40285c-9f87-4f85-824b-24c3b7c0e3bb  test@mail.com
stranger@TESTPC:~/Evalab/polypton-server$
```

Рисунок 9 - Добавление новой почтовой рассылки.

Для отключения рассылки воспользуйтесь командой:

```
sudo -E docker compose exec server python3 ./polypton-ctl.py --remove-email \
<device_id> <email>
```

```
stranger@TESTPC: ~/Evalab/polypton-server
stranger@TESTPC:~/Evalab/polypton-server$ sudo -E docker compose exec server python3 ./polypton-ctl.py --remove-email \
> 8c40285c-9f87-4f85-824b-24c3b7c0e3bb test@mail.com
Email 'test@mail.com' successfully deleted for device 8c40285c-9f87-4f85-824b-24c3b7c0e3bb.
stranger@TESTPC:~/Evalab/polypton-server$
```

Рисунок 10 - удаление почтовой рассылки.