

# eXpress

Система  
коммуникаций

## Руководство администратора

Развертывание чат-бота/SmartApp

13.05.2024



© Компания «Анлимитед продакшен», 2024. Все права защищены.

Все авторские права на эксплуатационную документацию защищены.

Без специального письменного разрешения компании «Анлимитед продакшен» этот документ или его часть в печатном или электронном виде не могут быть подвергнуты копированию или передаче третьим лицам с коммерческой целью.

Информация, содержащаяся в этом документе, может быть изменена разработчиком без специального уведомления, что не является нарушением обязательств по отношению к пользователю со стороны компании «Анлимитед продакшен».

Почтовый адрес:	127030, г. Москва, ул. Новослободская, д. 24, стр. 1
Телефон:	+7 (499) 288-01-22
Email:	sales@express.ms
Web:	<a href="https://express.ms/">https://express.ms/</a>

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>ПРОЦЕСС РАЗВЕРТЫВАНИЯ .....</b>	<b>5</b>
<b>Предварительные условия.....</b>	<b>5</b>
Системные требования .....	5
Проверка работы Docker .....	5
<b>Регистрация чат-бота в панели администратора .....</b>	<b>5</b>
<b>Подготовка к запуску .....</b>	<b>7</b>
Загрузка образа.....	7
Создание общего хранилища .....	8
Директория для чат-бота/SmartApp .....	8
<b>Запуск/остановка бота .....</b>	<b>9</b>
<b>Проверка работоспособности чат-бота.....</b>	<b>9</b>
<b>ОБНОВЛЕНИЕ ОБРАЗА ПРИЛОЖЕНИЯ .....</b>	<b>10</b>
Обновление до определенной версии .....	10
Обновление до последней версии .....	10

## ВВЕДЕНИЕ

Руководство предназначено для администраторов изделия «Система коммуникаций «Express», DevOps-инженеров и системных программистов. В нем содержатся сведения, необходимые для развертывания чат-бота или SmartApp.

**Служба технической поддержки.** Связаться со службой технической поддержки можно по электронной почте [support@express.ms](mailto:support@express.ms). Страница службы технической поддержки на сайте компании «Анлимитед продакшен» <https://express.ms/faq/>.

**Сайт в интернете.** Информация о продукте компании «Анлимитед продакшен» представлена на сайте <https://express.ms/>.

## ПРОЦЕСС РАЗВЕРТЫВАНИЯ

### ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ

Для чат-ботов и SmartApp рекомендуется использовать отдельный сервер. Между этим бот-сервером и CTS должно быть настроено сетевое взаимодействие в одну и другую сторону.

### СИСТЕМНЫЕ ТРЕБОВАНИЯ

Сервер, на котором выполняется развертывание, должен удовлетворять следующим требованиям:

- ОС Linux;
- установленное программное обеспечение:
  - Docker;
  - Docker Compose;
  - PostgreSQL v15 (docker-образ или сервер);
  - Redis v7 (docker-образ или сервер);
- открытый свободный порт в диапазоне 1024-65535.

### ПРОВЕРКА РАБОТЫ DOCKER

#### Для проверки работы Docker:

1. Проверьте статус Docker:

```
systemctl status docker
```

2. Если статус равен «inactive», запустите Docker командой:

```
sudo systemctl enable --now docker
```

Команда «enable» обеспечивает автозапуск Docker после включения системы. Контейнеры с параметром «restart: always» при этом также будут запущены.

**Примечание.** Если чат-бот/SmartApp должен взаимодействовать с информационной системой заказчика, то по вопросам сетевых взаимодействий обратитесь к разработчику, чтобы уточнить дополнительные требования.

### РЕГИСТРАЦИЯ ЧАТ-БОТА/SMARTAPP В ПАНЕЛИ АДМИНИСТРАТОРА

Перед запуском необходимо зарегистрировать чат-бот/SmartApp в панели администратора CTS.

**Примечание.** Данные для доступа в панель администратора CTS можно получить у разработчиков СК «Express».

#### Для регистрации чат-бот/SmartApp в панели администратора:

1. Перейдите в раздел «Боты» с помощью бокового меню ([Рисунок 1](#)).
2. Нажмите кнопку «Создать бота» в правом верхнем углу.

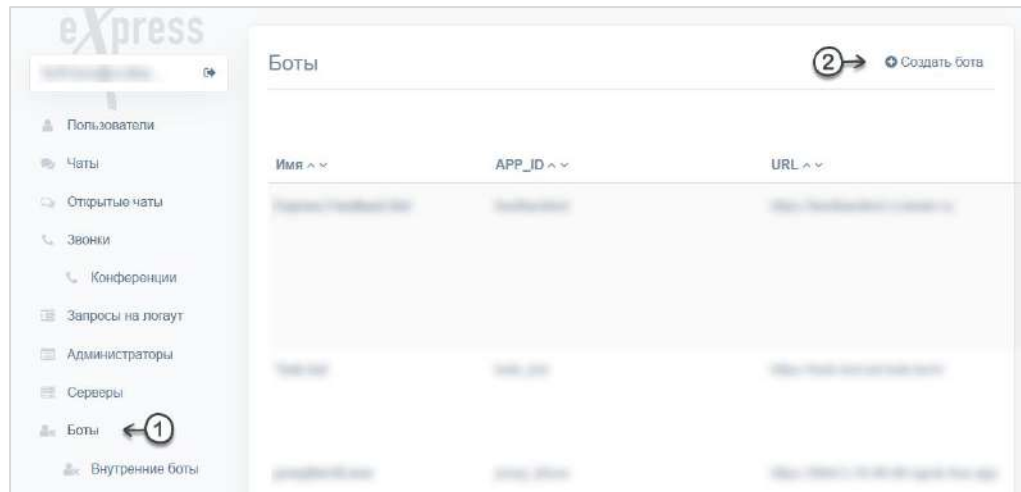


Рисунок 1

3. На странице создания чат-бота заполните все поля и сохраните изменения (Рисунок 2).

Чат-бот появится в общем списке.

**Примечание.** При заполнении поля «URL» убедитесь, что ссылка доступна извне.

Пример URL: «https://<bot server address>:8000».

Если на сервере установлено несколько чат-ботов, используется порт, указанный в файле `docker-compose.yml` конкретного чат-бота.

Рисунок 2

4. Нажмите на пиктограмму  напротив названия чат-бота (Рисунок 3).

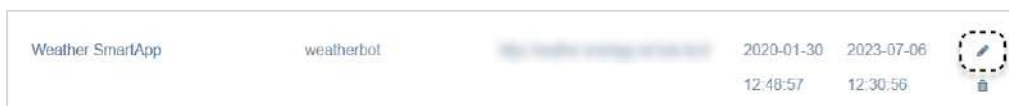


Рисунок 3

Откроется страница редактирования чат-бота.

5. Сохраните значения полей «ID» и «Secret key». Они понадобятся при [создании директории](#) для чат-бота/SmartApp.
6. Для SmartApp заполните поля в соответствующем блоке (Рисунок 4) и сохраните изменения.

Рисунок 4

## ПОДГОТОВКА К ЗАПУСКУ

Разместите на сервере следующие файлы, полученные у разработчиков:

- docker-compose.storages.yml
- docker-compose.yml
- example.env или .env

Дальнейшая подготовка включает следующие шаги:

1. [Загрузка образа.](#)
2. [Создание общего хранилища.](#)
3. [Создание директории для чат-бота/SmartApp.](#)

## ЗАГРУЗКА ОБРАЗА

Docker-образы чат-ботов и SmartApp доступны в публичном хранилище Docker Registry по адресу registry.public.express. Образ приложения указан в файле docker-compose.yml.

**Примечание.** В качестве логина и пароля используются значения Login и Password, которые выдаются разработчиком.

### Для загрузки образа:

1. Авторизуйтесь в Docker Registry, выполнив следующую команду:

```
docker login -u <Login> -p <Password> registry.public.express
```

2. Выполните команду:

```
docker compose up -d
```

Образ приложения будет загружен автоматически.

---

## СОЗДАНИЕ ОБЩЕГО ХРАНИЛИЩА

---

**Примечание.** Если общее хранилище было создано ранее (при развертывании других чат-ботов) – пропустите этот шаг.

Сформируйте отдельную docker-сеть, чтобы каждый чат-бот/SmartApp не создавал новый экземпляр PostgreSQL и Redis. Каждое приложение при этом будет использовать свою базу данных.

### Для формирования общего хранилища:

1. Создайте директорию для PostgreSQL и Redis:

```
mkdir -p /opt/express/bots/storages
```

2. Скопируйте в созданную директорию приложенный файл `docker-compose.storages.yml`.

3. В этой же директории создайте файл `.env` с содержимым:

```
POSTGRES_USER="postgres" # Общий пользователь PostgreSQL, у бота будет свой собственный
POSTGRES_PASSWORD="<GENERATE>"
```

**Примечание.** Для генерации пароля используйте команду «`openssl rand -hex 32`».

4. Запустите контейнеры командой:

```
docker compose -f docker-compose.storages.yml up -d
```

5. Проверьте статус контейнеров командой:

```
docker compose -f docker-compose.storages.yml ps
```

Статус должен иметь значение «up».

6. Проверьте, что записи в log-файлах хранилищ не содержат информацию об ошибках:

```
docker compose -f docker-compose.storages.yml logs
```

---

## ДИРЕКТОРИЯ ДЛЯ ЧАТ-БОТА/SMARTAPP

---

### Для формирования директории:

1. Создайте директорию с помощью команды:

```
mkdir -p /opt/express/bots/your_bot
```

2. Скопируйте в созданную директорию приложенный файл `docker-compose.yml`. Отредактируйте его в соответствии с комментариями в файле.

3. Создайте базу данных и пользователя для чат-бота:

```
docker exec storages-postgres-1 psql -U postgres -c "create user '<your_bot_user>'"
docker exec storages-postgres-1 psql -U postgres -c "alter user '<your_bot_user>' with password '<GENERATE>'"
docker exec storages-postgres-1 psql -U postgres -c "create database 'your_bot_db' with owner '<your_bot_user>'"
```

**Примечание.** Для генерации пароля используйте команду «`openssl rand -hex 32`».





## ОБНОВЛЕНИЕ ОБРАЗА ПРИЛОЖЕНИЯ

Каждый образ приложения размещен в публичном хранилище Docker Registry под тегом, соответствующим версии приложения. Последние доступные версии образов дополнительно помечаются тегом «latest».

### ОБНОВЛЕНИЕ ДО ОПРЕДЕЛЕННОЙ ВЕРСИИ

**Для загрузки определенной версии приложения** укажите соответствующий тег в файле docker-compose.yml. Если указанной версии образа нет на вашем сервере, Docker автоматически загрузит ее при запуске контейнеров приложения.

**Для загрузки образа вручную** используйте команду:

```
docker pull <image:tag>
```

### ОБНОВЛЕНИЕ ДО ПОСЛЕДНЕЙ ВЕРСИИ

Для обновления образа с тегом «latest» необходимо сначала удалить образ, имеющийся на сервере.

**Примечание.** Перед удалением образа приложения убедитесь, что контейнеры с приложением чат-бота/SmartApp остановлены.

**Для обновления последней версии приложения** удалите имеющийся образ с помощью команды:

```
docker rmi <image:latest>
```

Docker автоматически загрузит новый образ при запуске контейнеров приложения.

**Для загрузки образа вручную** используйте команду:

```
docker pull <image:latest>
```