

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«Покеткей Рус»
(ООО «Покеткей Рус»)

**Инструкция по установке
Модуль «Rocketkey Rocketoffice»**

г. Москва
2025

Определения и сокращения

1. АСУ ТП – автоматизированная система управления технологическим процессом;
2. ОС – Операционная система
3. БД – база данных;
4. СУБД – система управления базами данных;
5. ПАС – Программно–аппаратное средство
6. ПК – Персональный компьютер
7. ПО – Программное обеспечение
8. ППО – Прикладное программное обеспечение
9. Администратор – роль Пользователя, наделенная полными правами доступа к административному интерфейсу ПО, может управлять структурой Системы, управлять реестром Пользователей, иметь доступ к осталь–ным сервисам Системы
10. Пользователь – физические и юридические лица, так или иначе взаимодействующие с ПО посредством использования её оборудования и/или пользовательского интерфейса.
11. Терминал – аппаратное устройство, управляемое с помощью ППО или модуля ППО.

Данная инструкция предназначена для полного описания процесса установки модуля «Pocketkey Pocketoffice» (далее – «модуль», «Pocketoffice») в составе ППО «Pocketkey ACS» (далее – «ППО», «Программа») на Linux–подобные, Windows и macOS операционные системы.

В связи с постоянным развитием и улучшением продукта, разработчик оставляет за собой право вносить изменения в инструкцию по установке без предварительного уведомления. Для получения актуальной и полной информации рекомендуется обращаться в ООО «Покеткей Рус».

1. Общие сведения

Программный модуль «Pocketoffice» в составе ППО «Pocketkey ACS» (далее – «ППО», «Программа»), предназначен для реализации следующего функционала:

- Управления персоналом и отделами;
- Контроля доступа в помещения по расписаниям;
- Ведения отчетности по рабочему времени;
- Обработки гостевых заявок;
- Интеграции с внешними системами (1С, CRM).

Интеграция модуля с ППО обеспечивает:

- Централизованное управление терминалами через веб–интерфейс;
- Синхронизацию данных с терминалами доступа;
- Взаимодействие с CRM для проверки прав доступа;
- Логирование событий в реальном времени.

Примечание – Для работы модуля требуется предварительная настройка сервера администратором.

Pocketoffice предоставляет комплексное решение для контроля доступа, позволяя эффективно управлять персоналом, обеспечивать безопасность помещений и получать детальную отчетность.

2. Системные требования

2.1 Минимальные требования:

- Docker версии 20.10+
- Docker Compose версии 2.0+
- Операционная система: Ubuntu 16.04, Debian GNU/Linux 9 и выше.
- Свободное место: минимум 5 ГБ
- Оперативная память: минимум 2 ГБ

2.2 Рекомендуемые требования:

- SSD диск для базы данных: 1ТБ
- Свободное место: минимум 20 ГБ

- Оперативная память: 4 ГБ

3. Подготовка к установке

ВАЖНО! Перед созданием облачного сервера необходимо согласовать запрос с администратором.

Процедура запроса:

- Отправьте письмо администратору с запросом на создание облачного сервера;
- Укажите в письме все необходимые данные (см. пример ниже);
- Дождитесь подтверждения и получения доступа.

Пример письма для запроса:

Тема: Запрос на создание облачного сервера

Прошу создать облако для Pocketoffice.

Данные для создания:

- Клиент: [Название клиента];
- Имя хоста: [hostname].pocketoffice.pro.

После создания сервера получите:

- Ссылку на облако (имя хоста в формате [hostname].pocketoffice.pro);
- Креденциалы для SSH-подключения (логин, пароль или SSH-ключ).

Примечание— Если требуются дополнительные настройки или специфичные параметры сервера, укажите их в запросе.

4. Порядок установки

3.1 Подготовка системы

Проверьте установку Docker и Docker Compose:

```
```bash
```

```
docker —version
```

```
docker-compose —version
```

```
```
```

Если не установлен:

****Ubuntu/Debian:****

```
``bash
```

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

```
sudo sh get-docker.sh
```

```
sudo usermod -aG docker $USER
```

3.2 Получение кода

Скопируйте проект в рабочую директорию.

```
``bash
```

```
cd /home/pocketkey
```

```
git clone git@github.com:vkorey/accessbook.git
```

```
cd accessbook/accessbook/
```

3.3 Настройка конфигурации

3.3.1 Установка редактора

```
``bash
```

```
sudo apt-get install nano
```

```
...
```

3.3.2 Генерация паролей

```
``bash
```

Пароль для БД

```
python3 -c 'import secrets; print(secrets.token_urlsafe(10))'
```

Секретный ключ Django

```
python3 -c 'import secrets; print(secrets.token_urlsafe(50))'
```

3.3.3 Создание файла окружения

```
``bash
```

```
mv .env.exmpl .env
nano .env
'''
```

Отредактируйте параметры:

```
'''ini
Настройки базы данных
DB_PASSWORD=secure_password_here
SECRET_KEY=your-very-secret-key
```

Настройки домена (замените your-domain.com)

```
SERVER_NAME=your-domain.com
SERVERNAMES=your-domain.com localhost 127.0.0.1
CSRF_TRUSTED_ORIGINS=https://your-domain.com
```

Остальные параметры можно оставить по умолчанию

```
'''
```

3.3.4 Развертывание

Запуск проекта

```
'''bash
chmod +x deploy.sh
sudo ./deploy.sh
'''
```

Создание администратора

```
'''bash
docker exec -it pocketoffice ./manage.py createsuperuser
'''
```

Введите данные:

– Имя пользователя: `pocketkey`;

- Email: ваш email;
- Пароль: надежный пароль.

3.3.5 Проверка

Проверка сервисов

```
```bash
docker-compose logs app # логи приложения
docker-compose logs db # логи базы данных
```
```

Проверка веб-интерфейса

Откройте `http://localhost` – должна появиться страница авторизации.

Проверка API

```
```bash
curl http://localhost/ping/
Ответ: {"status": "ok"}
```

## 4. Настройка SSL

### 4.1 Автоматическое получение сертификатов

```
```bash
docker-compose -f docker-compose.ssl.yml up -d
```
```

### 4.2 Ручная настройка

- Поместите сертификаты в `nginx/ssl/`:
  - 1) `certificate.crt`;
  - 2) `private.key`.
- Перезапустите: `docker-compose restart`.

## 5. Первоначальная настройка

### 5.1 Создание пользователя для клиента

- Перейдите в админ–панель: `https://домен/admin/`;
- Войдите с учетными данными администратора;
- Создайте нового пользователя в разделе «Users»;
- Установите соответствующие права доступа;
- Передайте клиенту логин, пароль и URL входа.

## 5.2 Настройка системы

### Помещения и терминалы

- СКУД → Помещения → Добавить помещения;
- СКУД → Терминалы → Добавить терминалы:
  - 1) Название устройства;
  - 2) ID устройства;
  - 3) Помещение;
  - 4) Роль (Вход/Выход).

### Сотрудники

- Сотрудники → Отделы → Создать отделы;
- Сотрудники → Сотрудники → Добавить сотрудников:
  - 1) ФИО, должность, отдел;
  - 2) ID карты доступа.

### Расписания

- Рабочие смены - для расчета рабочего времени;
- Смены доступа - для контроля доступа в помещения.

### Настройка терминалов

Используйте конфигуратор `pkcfg.ru`:

- IP сервера: домен PocketOffice;
- Использовать SSL: Вкл.;
- Чекпоинт проверки: `checkaccess/` или `check\_in\_out/`;

- Метод: GET;
- Параметры: `card\_id`, `device`.

## 6. Резервное копирование

### 6.1 Настройка автоматического бэкапа

```
```bash
chmod +x backup.sh setup-cron.sh
./setup-cron.sh
```
```

### 6.2 Ручной бэкап

```
```bash
```

База данных

```
docker-compose exec db pg_dump -U pocketoffice_user pocketoffice >
backup_$(date +%Y%m%d).sql
```

Медиафайлы

```
tar -czf media_backup_$(date +%Y%m%d).tar.gz media/
```
```

## 7. Обновление

```
```bash
git pull origin main
docker-compose build --no-cache
docker-compose down
docker-compose run --rm app python manage.py migrate
docker-compose up -d
```
```

## 8. Устранение неисправностей

В процессе установки Pocketoffice могут возникнуть ошибки. Список ошибок и способы их устранения приведён в таблице 1.

Таблица 1

|                               |                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------|
| Контейнеры не запускаются     | <pre>&lt;pre&gt;docker-compose logs app</pre>                                                          |
|                               | <pre>docker-compose config&lt;/pre&gt;</pre>                                                           |
| Ошибки базы данных            | 1. Проверьте настройки в .env                                                                          |
|                               | 2. Убедитесь, что база данных запущена                                                                 |
| Статические файлы не работают | <pre>&lt;pre&gt;docker-compose exec app python manage.py<br/>collectstatic --noinput&lt;/pre&gt;</pre> |

## 7. Техническая поддержка

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе эксплуатации, осуществляются производителем и службой технической поддержки.

При возникновении проблем:

- Укажите описание проблемы;
- Предоставьте логи системы;
- Предоставьте информацию о конфигурации системы (без паролей).

Служба технической поддержки: ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ПОКЕТКЕЙ РУС», 109044, г. Москва, ул. Воронцовская, д. 20.

Тел.: +7–495–107–09–10, доб.2

E-mail: [support@pocketkey.ru](mailto:support@pocketkey.ru).