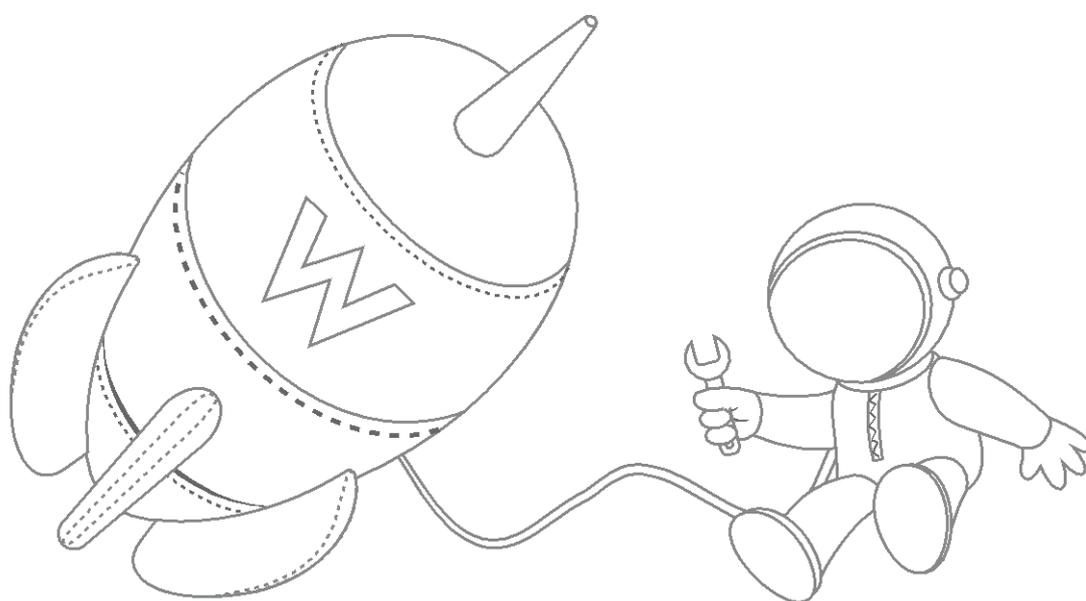


# Инструкция по установке экземпляра программного обеспечения Аним



**адрес**  
Москва, пер. Нижний Сусальный, 9/6

**сайт**  
[ittensive.ru](http://ittensive.ru)

**телефон**  
+7 925 8884500



## Аннотация

Настоящий документ разработан в рамках исполнения требований Постановления Правительства Российской Федерации от 28.12.2022 N 2461 и содержит описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения.





## Содержание

1 Общие положения и системные требования.....	4
2 Последовательность шагов установки ПО Аним.....	5
3 Настройка и запуск загрузки исторических данных.....	6
4 Настройка ежедневного обновления данных (выделенный сервер).....	7
5 Настройка ежедневного обновления данных (Яндекс.Облако).....	8
6 Создание витрин данных с бизнес-показателями.....	9





# 1. Общие положения

Настоящий документ относится к программному обеспечению, разработанному ООО «АйТиТенсив» и внесенному в Единый реестр российских программ для электронных вычислительных машин и баз данных:

- Автоматизация обработки данных и подготовки отчетности - Аним (далее ПО Аним)

## Системные требования

- Локальный компьютер или виртуальный/выделенный сервер с установленным интерпретатором Python (память не менее 1 Гб, частота CPU не менее 1 GHz - для загрузки исторических данных)
- СУБД MySQL/MariaDB, или PostgreSQL, или Clickhouse, или Oracle, или SQLite - для хранения данных и создания витрин.
- Виртуальный/выделенный сервер с установленным интерпретатором Python и планировщиком задач, или аккаунт Яндекс.Облака - для ежедневного обновления данных.

При возникновении вопросов по установке/развертыванию ПО Аним можно связаться с техническим специалистом по телефону: +7 926 7281964



## 2. Последовательность шагов установки ПО Аним

Полное развертывание ПО Аним на инфраструктуре заказчика состоит из следующих шагов:

1. Загрузка исходного код ПО Аним, расположенного по адресу <https://ittensive.ru/aneem.master.zip>
2. Распаковка архива ПО Аним в рабочую папку. Содержание архива:
  - a. settings.example.ini - шаблон файла настроек
  - b. requirements.txt - список требуемых модулей Python3
  - c. LICENSE - текст лицензии открытого программного обеспечения
  - d. initial - папка с модулями загрузки исторических данных
  - e. daily - папка с модулями загрузки ежедневных данных
  - f. datamart - папка с витринами данных, содержащих бизнес-показатели
3. Копирование файла settings.example.ini в файл settings.ini в рабочей папке и заполнение последнего актуальными настройками.  
Для загрузки данных в базу данных необходимо заполнить блок [DB] в файле settings.ini следующими значениями.
  - a. TYPE - тип СУБД. Возможные значения: MYSQL, POSTGRESQL, MARIADB, ORACLE, CLICKHOUSE, SQLITE
  - b. HOST - адрес хоста СУБД (включая порт для всех типов, кроме CLICKHOUSE).
  - c. USER - имя пользователя СУБД.
  - d. PASSWORD - пароль пользователя СУБД.
  - e. DB - имя базы в СУБД, предназначенной для хранилища данных.
4. Установка дополнительных модулей Python по списку из файла requirements.txt в корне рабочей папки.
5. Загрузка исторических данных из нужных источников (стр. 6).
6. Настройка ежедневного обновления данных из нужных источников (стр. 7).
7. Создание витрин данных, содержащих бизнес-показатели (стр. 8).



## 3. Настройка и запуск загрузки исторических данных

Модуля загрузки исторических данных находятся в папке `initial` в рабочей папке.

Загрузка исторических данных выполняется независимыми модулями, имеющимися отдельные группы настройки в файле `settings.ini`. Для запуска каждого требуемого модуля загрузки данных необходимо заполнить его блок в файле `setings.ini`.

В большинстве случаев требуется указать ключ API (получаемый по инструкции на сервисе, с которого загружаются данные) - `ACCESS_TOKEN / KEY` - и целевую таблицу, в которую данные будут загружены - `TABLE-{POSTFIX}`.

Для ряда модулей также доступны настройки:

- `DELTA` - размер периода загружаемых данных (в днях),
- `PERIODS` - количество пер иодов загрузки данных,
- `LOGIN / COUNTER_ID / APPLICATION_ID / SITEID` - логин или аккаунт в сервисе (предоставляющем данные),
- `WEBHOOK` - адрес точки доступа API, содержащий `ACCESS_TOKEN`,
- `ENCODING` - кодировка исходных данных,
- `GEO` - ограничения по географии собираемых данных

и ряд других.

После заполнения всех настроек отдельного модуля, загружающего исторические данные, необходимо его запустить в консоле, используя интерпретатор Python не ниже версии 3.12

```
python <имя_модуля>
```

например

```
python users.bitrix24.py
```

По ходу загрузки данных будет выдаваться статистика о количество полученных записей.



## 4. Настройка ежедневного обновления данных (выделенный сервер)

Для настройки ежедневного обновления данных на выделенном (или виртуальном) сервере необходимо скопировать рабочую папку ПО Аним (содержащую все файлы) на сервер и заполнить файл `settings.ini` в корне рабочей папки требуемыми настройками (соответствующим всем модулям, которые были использованы для загрузки исторических данных).

После этого необходимо настроить планировщик задач для ежедневного запуска всех ежедневных модулей. Пример планировщика задач (который необходимо отредактировать под актуальный список модулей) содержится в файле `daily/standalone/crontab` рабочей папки.



## 5. Настройка ежедневного обновления данных (Яндекс.Облако)

В качестве альтернативы использования выделенного/виртуального сервера для обновления данных можно получить тот же результат, используя облачные функции и триггеры Яндекс.Облака.

Для создания нужных облачных функций (соответствующих модулям, используемым для загрузки исторических данных) используйте документацию:

<https://yandex.cloud/ru/docs/functions/operations/function/function-create>

Исходный код облачных функций (включая файл requirements.txt для каждой функции и рекомендуемые ограничения по ресурсам) находится в папке daily/yandex.cloud в рабочей папке проекта.

При создании облачной функции необходимо также заполнить Переменные окружения, их список и ожидаемые значения указаны в коде каждого файла с исходным кодом облачной функции и дублируют по сути значения настроек из файла settings.ini в корне проекта.

После создания требуемых облачных функций необходимо настроить их ежедневное выполнение через триггеры (таймеры):

<https://yandex.cloud/ru/docs/functions/operations/trigger/timer-create>

Для времени запуска можно руководствоваться примерами из файла initial/standalone/crontab в папке проекта.

Для таймеров рекомендуется выставлять не менее 3 повторных выполнений (в случае ошибки).



## 6. Создание витрин данных (с бизнес-показателями)

По наличию данных в СУБД (например, после загрузки исторических данных, стр. 6) возможно создать витрины (организованные представления) для облегчения потребления данных в BI-системах.

Для создания витрин необходимо перейти в папку datamart в рабочей папке, выбрать нужную СУБД и выбрать требуемые файлы витрин, содержащие SQL код. Выбранный SQL код необходимо выполнить в консоли СУБД, используемой как хранилище данных. Среди доступных витрин:

- Вспомогательные таблицы с календарями.
- Статистика посещаемости сайтов и установок мобильного приложения.
- Статистика рекламных расходов из кабинетов Яндекс.Директ и ВК.
- Статистика лидов и сделок Битрикс24.
- Текущее состояние всех имеющихся источников данных.
- Базовая и продвинутая атрибуция (8 моделей) сквозной аналитики маркетинга.

Витрины построены на значениях таблиц данных по умолчанию (заданных в файле settings.ini). При изменении названий таблиц с данными потребуются изменение кода SQL запроса витрин для обеспечения их работоспособности.

