



**ПРОГРАММНЫЙ ИНТЕРФЕЙС (API)  
ПРИЛОЖЕНИЙ ЦЕНТРА ОБРАБОТКИ ВЫЗОВОВ**

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

На 15 листах

Москва, 2022 г.

## СОДЕРЖАНИЕ

1. Общие сведения о документе.....	3
2. Общие сведения о программном обеспечении «Программный интерфейс» (API) приложений Центра обработки вызовов» .....	3
3. Первоначальная настройка необходимых компонент .....	4
3.1 Требования к серверному обеспечению и операционной системе .....	4
3.2 Установка Веб-сервера Nginx.....	4
3.3 Настройка Веб-сервера Nginx .....	5
3.4 Установка PHP 7 .....	8
3.5 Установка СУБД MariaDB .....	9
3.6 Настройка СУБД MariaDB.....	10
3.7 Проверка работы СУБД MariaDB.....	11
4. Установка телефонной платформы FreeSWITCH.....	11
5. Установка экземпляра ПО.....	11
6. Проверка работоспособности экземпляра .....	13
7. Справочная информация .....	14
8. Перечень сокращений.....	14

## 1. Общие сведения о документе

Настоящий документ содержит рекомендации для администратора по установке серверной составляющей ПО «Программный интерфейс (API) приложений Центра обработки вызовов» (далее – Программный интерфейс).

## 2. Общие сведения о программном обеспечении «Программный интерфейс (API) приложений Центра обработки вызовов».

Для решения задач интеграций различных информационных систем используется функционал API. API (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface, API [эй-пи-ай]) — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

**Программный интерфейс (API) приложений Центра обработки вызовов** — это набор процедур, разработанных отечественной ИТ-компанией ООО «Пигато», позволяющий взаимодействовать через единую точку входа с открытой телефонной платформой FreeSWITCH и внешними контент-провайдерами (агрегаторами), предоставляющими сервис по отправке SMS. Разработанное решение позволяет упростить процесс управления процессами обслуживания клиентов в ЦОВ.

Новизной и важной отличительной характеристикой описываемого ПО является наличие интуитивно понятного графического интерфейса (GUI) для конфигурирования API. Наличие графического интерфейса позволяет менеджеру проекта без участия ИТ-специалиста управлять доступностью модулей API, фактически конфигурируя функционал API ПО «Программный интерфейс» (Low-code технология).

### 3. Первоначальная настройка компонент.

#### 3.1 Требования к серверному обеспечению и операционной системе.

Необходимые минимальные требования к аппаратному обеспечению:

- RAM 2Gb
- HDD 50Gb
- CPU 2 core от 2GHz
- LAN 100 Мбит/сек.

Необходимые требования для работы программного обеспечения:

- Операционная система Debian
- Веб-сервер Nginx
- PHP версия не ниже 7.0
- СУБД MariaDB

Поддерживаемые браузеры Google Chrome, Internet Explorer версии 8 и выше.

В начале работ необходимо убедиться, что на сервере существует обычный не-рутовый (non-root) пользователь с привилегиями sudo. В случае отсутствия такого пользователя его необходимо создать и все команды выполнять из-под его учетной записи.

Перед разворачиванием компонент ПО «Программный интерфейс», необходимо обновить ОС до актуальной версии, выполнив команду:

```
$ sudo apt update -y && apt upgrade -y
```

#### 3.2 Установка Веб-сервера Nginx

Производитель гарантирует работу ПО «Программный интерфейс» с Веб-сервером Nginx.

Установка ПО Веб-сервера Nginx из репозитория Linux командой:

```
$ sudo apt install nginx -y
```

В результате выполнения команды пакет `apt` установит Nginx и все необходимые для его работы пакеты на сервер.

ПО Веб-сервера установлено, переходим к этапу его настройки.

### 3.3 Настройка Веб-сервера Nginx.

Для корректной работы ПО «Программный интерфейс» необходимо внести следующие изменения в конфигурационный файл Веб-сервера Nginx:

```
server {  
    listen 81;  
    set $root_path /var/www;  
    access_log /var/log/nginx/api-v2_access.log;  
    error_log /var/log/nginx/api-v2_error.log;  
  
    client_max_body_size 80M;  
    client_body_buffer_size 128k;  
  
    location /api-v2/index.php {  
        root /var/www/html/api-v2;  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
        fastcgi_index index.php;  
        fastcgi_param SCRIPT_FILENAME $root_path$fastcgi_script_name;  
        include fastcgi_params;  
        fastcgi_param DOCUMENT_ROOT $root_path;  
        fastcgi_buffers 4 256k;  
        fastcgi_busy_buffers_size 256k;  
        fastcgi_temp_file_write_size 256k;  
        # large_client_header_buffers 4 64k;
```

```
}

location / {
    root /var/www/html/api-v2;
    try_files $uri $uri index.php?$args;
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    #fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    index index.php;
}

location /api-client {
    root /var/www/api-client/;
    try_files $uri $uri index.php?$args;
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    #fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    index index.php;
}

location ~ ^/(app|models|controllers) {
    deny all;
}

location = /api-v2/index.php {
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    #fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    #fastcgi_pass 127.0.0.1:9000;
    fastcgi_read_timeout 15m;
    fastcgi_index index.php;
    include fastcgi_params;
    #fastcgi_param DOCUMENT_ROOT $root_path;
    fastcgi_param SCRIPT_FILENAME
/var/www$fastcgi_script_name;
}

location ~ \.php$ {
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
```

```
#fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
#fastcgi_pass 127.0.0.1:9000;
fastcgi_read_timeout 15m;
fastcgi_index index.php;
include fastcgi_params;
#fastcgi_param DOCUMENT_ROOT $root_path;
fastcgi_param SCRIPT_FILENAME /var/www/html/api-
v2/$fastcgi_script_name;
    }
}
```

```
server {
    listen 82;
    set $root_path /var/www/html;
    access_log /var/log/nginx/api-admin-panel_access.log;
    error_log /var/log/nginx/api-admin-panel_error.log;
    root /var/www/html/api-admin-panel/web/;

    index    index.php;

    client_max_body_size 80M;
    client_body_buffer_size 128k;

    location / {
        try_files $uri $uri /index.php$is_args$args;
    }
    location ~ ^/assets/.*\.php$ {
        deny all;
    }

    # отключаем обработку запросов фреймворком к несуществующим
    статичным файлам
    location ~ /\.(js|css|svg|png|jpg|gif|swf|ico|pdf|mov|fla|zip|rar)$ {
        try_files $uri =404;
```

```
}

location ~ /\.php$ {
    fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    #fastcgi_pass 127.0.0.1:9000;
    fastcgi_read_timeout 15m;
    fastcgi_index index.php;
    include fastcgi_params;
#    fastcgi_param DOCUMENT_ROOT $root_path;
try_files $uri =404;
    fastcgi_param SCRIPT_FILENAME /var/www/html/api-admin-
panel/web/$fastcgi_script_name;
}

}
```

Предварительная настройка ПО Веб-сервера Nginx выполнена.

Более подробная документация по установке и настройке Веб-сервера Nginx при необходимости может быть получена на официальном сайте <https://nginx.org/ru/>

### 3.4. Установка PHP 7

Т.к. в качестве серверной ОС используется Debian воспользуемся его официальным репозиторием.

Для установки PHP для Веб-сервера Nginx необходимо выполнить команду:

```
$ sudo apt install php7.0-fpm -y
```

Проверить полученную и установленную версию можно следующей командой:

```
$ php -v
```

Результат вывода команды будет примерно такой:

```
PHP 7.0.20-1~ubuntu16.10.1+deb.sury.org+1 (cli) (built: Jun  9 2017 08:09:54) (
NTS )
Copyright (c) 1997-2017 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
    with Zend OPcache v7.0.20-1~ubuntu16.10.1+deb.sury.org+1, Copyright (c) 1999
-2017, by Zend Technologies
    with Xdebug v2.5.4, Copyright (c) 2002-2017, by Derick Rethans
```

Далее, переходим к следующему этапу – разворачиванию СУБД.

### 3.5 Установка СУБД MariaDB.

Установим пакет MariaDB-server:

```
$ sudo apt install mariadb-server -y
```

По умолчанию в репозиторий пакетов АРТ в Debian включена последняя версия MariaDB.

### 3.6 Настройка СУБД MariaDB

В случае новой установки MariaDB необходимо выполнить скрипт безопасности. Он изменяет некоторые настройки по умолчанию на более

безопасные, например, удалённый вход для пользователей root и пользователи, созданные по умолчанию.

Для этого выполните скрипт безопасности командой:

```
$ sudo mysql_secure_installation
```

После выполнения вышеуказанной команды вам будет предложено указать различные параметры.

### 3.7 Проверка функционирования СУБД MariaDB.

После установки СУБД MariaDB, мы можем управлять сервисом с помощью службы Systemd.

Чтобы проверить статус MariaDB необходимо использовать следующую команду:

```
$ sudo systemctl status mariadb
```

Попробуем подключиться к СУБД используя команду:

```
$ sudo mysqladmin version
```

Вывод при успешном подключении к СУБД MariaDB:

```
mysqladmin Ver 9.1 Distrib 10.5.11-MariaDB, for debian-linux-gnu on x86_64  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Server version 10.5.11-MariaDB-1
```

```
Protocol version 10
Connection Localhost via UNIX socket
UNIX socket /run/mysqld/mysqld.sock
Uptime: 3 hours 45 min 24 sec

Threads: 1 Questions: 47 Slow queries: 0 Opens: 17 Open tables: 8 Queries per second avg: 0.036
```

Если СУБД MariaDB не запущена, ее можно запустить с помощью команды:

```
$ sudo systemctl start mariadb
```

#### 4. Установка телефонной платформы FreeSWITCH;

Развертывание открытой телефонной VoIP-платформы FreeSWITCH происходит либо из исходных кодов, либо из официального Freeswitch-репозитория, который предоставляет бинарные пакеты (deb/rpm и др), для доступа к нему требуется регистрация и получение аутентификационных данных.

Подробнее, см. официальную документацию:

<https://freeswitch.org/confluence/display/FREESWITCH/Debian>

<https://freeswitch.org/confluence/pages/viewpage.action?pageId=61210850>

<https://freeswitch.org/confluence/display/FREESWITCH/HOWTO+Create+a+SignalWire+Personal+Access+Token>

#### 5. Установка экземпляра ПО

Для работы ПО «Программный интерфейс» необходимо произвести первоначальное создание внутренней базы данных. Для этого предназначена

система миграции, которая позволяет создать необходимые таблицы и устанавливать обновления.

1. Необходимо перейти в директорию `/var/www/` и распаковать там предоставленный по договору поставки экземпляр ПО.

2. В результате будут созданы 2 директории:

```
/var/www/api-admin-panel/
```

```
/var/www/api-v2
```

3. Далее необходимо перейти в директорию `/var/www/api-admin-panel/` и выполнить команды:

```
$sudo composer update  
$chmod 0777 /runtime  
$chmod 0777 /web/assets
```

4. Изменение настроек подключения к СУБД MariaDB в файле `/var/www/api-admin-panel/config/db.php`

5. Далее необходимо осуществить миграцию с помощью скрипта:

```
$php yii migrate
```

В процессе миграции будут выводиться информационные сообщения о ходе выполнения процедуры.

5. На следующем этапе необходимо создать директорию для локального кеша:

```
$mkdir /var/www/tmp/
```

6. Далее необходимо сконфигурировать домен. Корневой директорией домена будет являться: /var/www/api-admin-panel/web/

## 6. Проверка работоспособности экземпляра ПО

Для проверки работоспособности экземпляра ПО необходимо обратиться по адресу с указанием порта указанного ранее в конфигурационном файле:  
[http://адрес\\_вашего\\_WEBсервера:82](http://адрес_вашего_WEBсервера:82)

В случае успешной инсталляции будет предложено залогиниться в ПО:

API для Центра обработки вызовов

Dashboard Настройки Пользователи Документация Войти

### Авторизация

Введите данные для авторизации:

Логин  Необходимо заполнить «Username».

Пароль

Запомнить меня

© 2022 «Pigato» Limited liability company («Pigato» LLC). All rights reserved.  
[О продукте](#)

Рис. 1. Стартовая страница интерфейса административной панели ПО

Для дальнейшей работы с ПО воспользуйтесь документами: «Программный интерфейс (API) приложений центра обработки вызовов. Инструкция администратора» и «Программный интерфейс (API) приложений центра обработки вызовов. Инструкция пользователя»

## 7. Справочная информация

Технологический стек, примененный при разработке данного ПО:

СУБД: MariaDB, язык программирования интерфейсов: PHP, основной протоколы взаимодействия: Rest API.

Для получения расширенной информации о функционале или заказе демонстрации ПО «Программный интерфейс» напишите нам на:

info@pigato.ru — общие вопросы,

support@pigato.ru — технические вопросы,

либо, позвоните по телефону: +7 499-216-63-61 (с 09:00 до 18:00)

## 8. Перечень сокращений, принятых в настоящем документе:

CDR — (англ. Call Data Records) — записи данных об обработанных вызовах в телекоммуникационной сфере,

GUI — (англ. Graphical User Interface), графический интерфейс пользователя,

SMS — (сокращение от англ. Short Message Service) — «служба коротких сообщений», СМС) — технология приёма и передачи коротких текстовых сообщений,

URL — (сокращение от англ. Uniform Resource Locator) — адрес, который выдан уникальному ресурсу в интернете,

СУБД — система управления базами данных;

Компания — производитель ПО ООО «Пигато»,

КЦ — Контактный центр,

ПК — Персональный компьютер,

ПО — Программное обеспечение,

ТП — Техническая поддержка,

ЦОВ — центр обработки вызовов.