

FleetGarage API server install manual

The latest version of this document can be found at:

<https://fleetgarage.eu/content/documents/FleetGarage-API-server-install-manual.pdf>

Preparatory actions

Before starting installation, you need to prepare:

1. Login and password of the first user, which you specify during the installation of ubuntu.
2. Access to files (requests, work orders, etc.).
 - a. File access method (samba/ftp).

If access is supposed to be done via ftp, install ftp-server (for example, FileZilla).
 - b. Local network addresses for which simplified authorization will be possible.
 - c. Server address, login and password for the file storage location to the server to access these files.
 - d. The main folder with files on the server must contain subfolders: ACCOUNTS_FILES and CUSTOMER_FILES.
3. Router through which requests from the Internet go:
4. Router address (it will need to be excluded from local network addresses);
5. b. should not change the outgoing address.
6. 4. Make sure the following files are available: libltdl.la, laravel.conf, laravel.cron, laravel-sv.conf.
7. 5. Make sure you have a GitHub access token to get the API server repository.

Lubuntu 22.04 setup

Download an official copy of the installation media from the Internet and install Lubuntu 22.04 Desktop from the distribution kit. Lubuntu is a lightweight version of the main ubuntu operating system that uses LXDE as its desktop environment.

If in the future all support for the API server will be carried out only by the customer's own resources, then it is possible to install a server version without a graphical environment. To be able to provide support from the developer, you must install the Desktop version.

Go through the keyboard configuration, because often after installation it is impossible to switch keyboard layouts, for which in the superuser terminal run:

```
dpkg-reconfigure keyboard-configuration
```

Ok



Installing server web components

In a superuser terminal, run:

```
apt update && apt upgrade -y
```

```
add-apt-repository -y ppa:ondrej/php
```

```
add-apt-repository -y ppa:x2go/stable
```

```
apt install -y openssh-server mc krusader kate konsole apache2 kde-cli-tools x2goserver lxde-session  
smbclient cifs-utils
```

```
/etc/init.d/x2goserver restart
```

Copy the file **libltdl.la** from FleetGarage distribution kit **to /usr/lib/x86_64-linux-gnu**

In an open superuser terminal, **run:**

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
```

```
curl -o /etc/apt/sources.list.d/microsoft.list
```

```
"https://packages.microsoft.com/config/ubuntu/$(lsb_release -sr)/prod.list"
```

```
apt update
```

```
apt install -y php8.2 php8.2-zip php8.2-common php8.2-sqlite3 php8.2-dev php8.2-xml php8.2-curl  
php8.2-mbstring php8.2-intl php8.2-bcmath php8.2-mysql php8.2-fpm php8.2-raphf php8.2-redis  
php8.2-gd
```

```
apt install -y composer curl unixodbc-dev redis git
```

```
phpenmod xml curl mbstring
```

choose **php 8.2** as the main php if there are several of them:

```
update-alternatives --config php
```

Run:

```
pecl uninstall -r sqlsrv
```

```
pecl uninstall -r pdo_sqlsrv
```

```
pecl -d php_suffix=8.2 install sqlsrv
```



```
pecl -d php_suffix=8.2 install pdo_sqlsrv
```

```
ACCEPT_EULA=Y apt-get install msodbcsql17 sqlcmd
```

```
wget https://www.openssl.org/source/openssl-1.1.1t.tar.gz -O openssl-1.1.1t.tar.gz
```

```
tar -zxvf openssl-1.1.1t.tar.gz
```

```
cd openssl-1.1.1t
```

```
./config
```

```
make
```

```
make install
```

```
ldconfig
```

```
ln -sf /usr/local/bin/openssl `which openssl`
```

```
openssl version
```

Make sure **version 1.1.1t** is currently installed.

Edit /etc/ssl/openssl.cnf: at the end of file in the string
CipherString = DEFAULT:@SECLEVEL=2

change =2 to =0

In a superuser terminal, run:

```
printf "; priority=30\nextension=sqlsrv.so\n" > /etc/php/8.2/mods-available/sqlsrv.ini
```

```
printf "; priority=30\nextension=pdo_sqlsrv.so\n" > /etc/php/8.2/mods-available/pdo_sqlsrv.ini
```

```
phpenmod sqlsrv pdo_sqlsrv
```

```
a2enmod ssl rewrite headers proxy proxy_fcgi
```

```
systemctl restart php8.2-fpm
```

```
systemctl enable php8.2-fpm
```

```
a2dismod mpm_prefork
```



```
a2enmod mpm_worker
```

```
mkdir /www
```

```
chmod 0775 /www
```

```
chgrp www-data /www
```

```
usermod -a -G www-data <admin user name, on behalf of which further installation and configuration will be carried out, not root>
```

```
composer self-update
```

Log out, then log in. No reboot required.

Setting up access to the main program files via samba

If access via samba was selected, you will need to do the following in the superuser terminal (you will need a login, password and domain name):

```
mkdir /mnt/api-v2-share
```

```
mkdir /mnt/api-v2-share/workorder
```

```
mount.cifs //<server>/<share> /mnt/api-v2-share/workorder -o  
user=<user>,domain=<domain>,password=<password>,rw,file_mode=0777,dir_mode=0777
```

Make sure the folder is mounted

Edit **/etc/fstab**

Add string:

```
"//<server>/<share> /mnt/api-v2-share/workorder cifs  
rw,user=<user>,pass=<pass>,domain=<domain>,file_mode=0777,dir_mode=0777"
```

Setting up access to the main program files via ftp

No additional steps are required.

Setting up apache and php

Copy the **laravel.conf** file to the folder **/etc/apache2/sites-enabled/laravel.conf**

Enter your server domain name and email.

Copy the file **laravel.cron** to **/etc/cron.d**

Edit the file **/etc/php/8.2/fpm/php.ini** (from root account)



Increase your upload file size (`upload_max_filesize`) to 32 or 64 (for uploading photos and videos through mobile applications).

Increase the total post request size (`post_max_size`) to 33 or 65 Mb (for uploading photos and videos through mobile applications).

Increase the memory size for php (`memory_limit`) to 4096M.

Edit `/etc/php/apache2/mods-available/mpm_worker.conf`

Increase `StartServers` to 5

Increase `MinSpareThreads` to 30-50 (depending on the expected load)

Increase `MaxSpareThreads` to 200

Installing the API Server

In a regular user terminal, run the commands:

```
cd /www
```

```
composer create-project --prefer-dist laravel/laravel:10 api-v2
```

```
cd api-v2
```

```
composer require firebase/php-jwt predis/predis ext-json tymon/jwt-auth workerman/workerman  
intervention/image awobaz/compoships phplrt/phplrt:3.5.2 guzzlehttp/guzzle ext-libxml ext-  
simplexml ext-redis ext-fileinfo ext-intl amphp/amp amphp/parallel ext-pdo illuminate/database  
illuminate/hashing league/flysystem-ftp kirschbaum-development/eloquent-power-joins
```

Run

git clone <https://github.com/CONE-FleetGarage/api-v2-l10-release.git>, , select a token using the authorization method, enter the sent token (or any login, password - token).

Copy or move the contents of a folder `api-v2-l10-release` to `www/api-v2`.

API Server Configuration

Edit the `config/database.php` file, change the path to the sqlite database. Enter your data: SQL server address, user name and password from which API requests to the database will be made.

There are 2 such places:

- `sqlsrv;`
- `sqlsrv_log.`

The `sqlsrv_general` section can be left unchanged or commented out.



Add string

'trust_server_certificate'=>true,

in these 3 points (sqlsrv, sqlsrv_log, sqlsrv_general).

Edit the config/app.php file, set the correct timezone

Edit the .env file, change DB_CONNECTION to "sqlite", change QUEUE_CONNECTION to "redis".

Edit the **config/database.php** file, add it to the section

"redis->options" after "cluster" line:

```
'prefix' => env('REDIS_PREFIX', '')
```

Edit the config/intranet.php file, add a line after

"return [" with the router address in the form:

```
'!192.168.1.1',
```

An exclamation point before an address means that the address should be excluded from local ones.

Change the following line to include the current subnet, requests from which should be considered local.

Edit the file

vendor/laravel/framework/src/Illuminate/Database/Query/Grammars/SqlServerGrammar.php
in the function getDateFormat change format to **"Y-m-d\\TH:i:s.v"**

Edit the file

vendor/nesbot/carbon/src/Carbon/Traits/Creator.php

, in function rawCreateFromFormat after

```
// First attempt to create an instance, so that error messages are based on the unmodified format.
```

```
$date = self::createFromFormatAndTimezone($format, $time, $tz);
```

```
$lastErrors = parent::getLastErrors();
```

Add

```
$time = preg_replace('/(\\.\d{6})(\d*)/', '$1', $time);
```

```
if ($lastErrors and $lastErrors['error_count'])
```

```
$date = self::createFromFormatAndTimezone($format, $time.'.0', $tz);
```

```
$lastErrors = parent::getLastErrors();
```

```
if ($lastErrors and $lastErrors['error_count'])
```



```

        $date = self::createFromFormatAndTimezone(str_replace('\T', ' ', $format), $time, $tz);
        $lastErrors = parent::getLastErrors();
        if ($lastErrors and $lastErrors['error_count'])
            $date = self::createFromFormatAndTimezone(str_replace('\T', ' ', $format), $time.'.0', $tz);
        $lastErrors = parent::getLastErrors();
        if ($lastErrors and $lastErrors['error_count'] and strpos($time, ' ')===false and strpos($time,
'T')===false)
            $date = self::createFromFormatAndTimezone('H:i:s.u', $time, $tz);
        $lastErrors = parent::getLastErrors();
        if ($lastErrors and $lastErrors['error_count'] and strpos($time, ' ')===false and strpos($time,
'T')===false)
            $date = self::createFromFormatAndTimezone('H:i:s.u', $time.'.0', $tz);
        $lastErrors = parent::getLastErrors();

```

Edit the files:

- /vendor/laravel/framework/src/Illuminate/Database/Query/Grammars/Grammar.php
- /vendor/laravel/framework/src/Illuminate/Database/Query/Builder.php

similar to those attached in .Changed, see comment lines
"// 2024-02-01"

Edit the file

/vendor/kirschbaum-development/eloquent-power-joins/src/Mixins/RelationshipExtraMethods.php
similar to those attached in .Changed, see comment lines

Perform the migration:

./artisan migrate

Then run

sudo chgrp -R www-data /www/api-v2

sudo chmod -R g+rw /www/api-v2

Configure an SSL certificate (if missing, get one from Let's Encrypt).



Reboot server apache:

sudo service apache2 restart

sudo systemctl restart php8.2-fpm

Reboot cron:

sudo /etc/init.d/cron restart

Install supervisor:

sudo apt install supervisor

Copy file:

laravel-sv.conf to folder /etc/supervisor/conf.d

Install supervisor:

sudo service supervisor restart

Install a data backup system.