

# МАРКЕТПЛЕЙС «СВОЕ РОДНОЕ»

---

**Руководство по установке и настройке**

# СОДЕРЖАНИЕ

Содержание2

Определения и сокращения3

Определения3

Сокращения3

1 Назначение системы4

2 Структура системы4

3 Технические требования5

3.1 Требования к конфигурации серверов5

3.2 Операционная система5

3.3 Состав дистрибутива системы5

4 Установка и развертывание системы7

4.1 Установка подсистемы управления контентом7

4.2 Установка сервера приложений12

5 Установка мобильного приложения для покупателя15

# ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

## Определения

Термин	Определение
Продавец	Пользователь маркетплейса «Свое Родное» размещающий товары и осуществляющий коммерческую деятельность по реализации данных товаров на маркетплейсе. Размещение товаров осуществляется на платформе «Своефермерство».
Покупатель	Пользователь маркетплейса «Свое Родное», который имеет возможность поиска и выбора товаров и оформления заказов для приобретения данных товаров.

## Сокращения

Обозначение, сокращение	Значение
API	Application Programming Interface - программный интерфейс приложения
CMS	Content Management System – система управления контентом.
REST	Representational State Transfer – архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
REST API	API, реализованное с использованием REST архитектуры.
БД	База данных
ПО	Программное обеспечение
СУБД	Система управления базами данных

# 1 НАЗНАЧЕНИЕ СИСТЕМЫ

Маркетплейс «Свое Родное» предназначен для обеспечения эффективного взаимодействия между фермерскими хозяйствами, которые производят потребительские товары и розничными покупателями по реализации продукции фермерских хозяйств в сети Интернет.

Основные функциональные элементы маркетплейса «Свое Родное»:

- мобильное приложение обеспечивающее поиск, выбор и приобретение товаров покупателем;
- сайт в сети Интернет обеспечивающий поиск, выбор и приобретение товаров покупателем.

*Примечание: регистрация продавцов на торговой площадке и размещение товаров осуществляется на платформе «Своефермерство», которая входит в Экосистему для предприятий АПК.*

# 2 СТРУКТУРА СИСТЕМЫ

Маркетплейс «Свое Родное» (далее Система) включает в свой состав:

- подсистему управления контентом, включающую хранилище данных и веб приложение, предназначенное для информационного администратора Системы;
- сервер приложений для веб интерфейса и мобильного приложения покупателя;
- мобильное приложение, предназначенное для поиска, выбора и приобретения товаров покупателем.

Схема структуры Системы приведена на рисунке ниже (см. Рисунок 1).

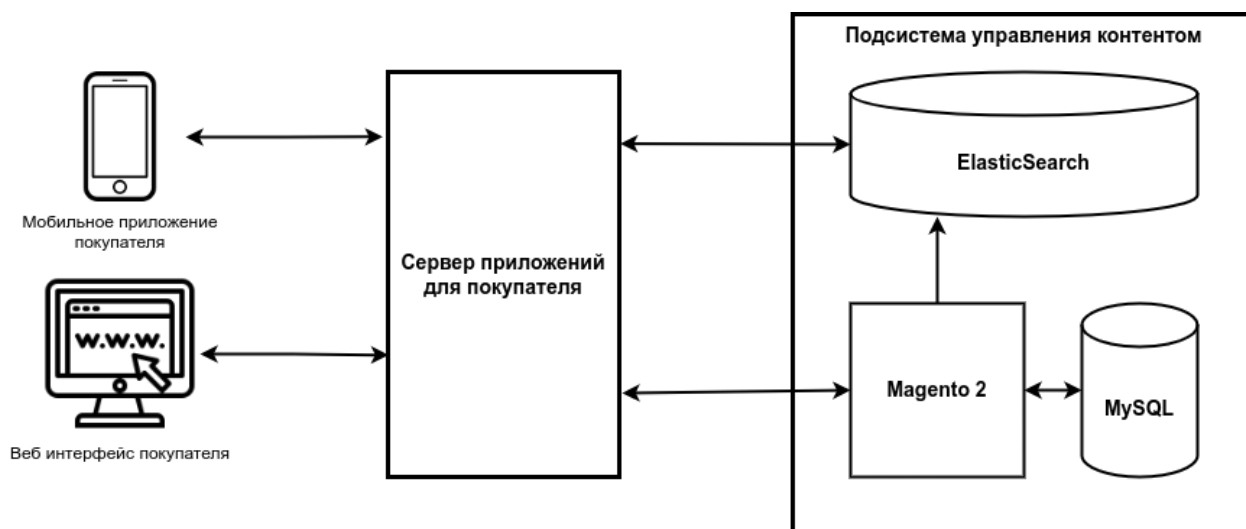


Рисунок 1

## 3 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

### 3.1 Требования к конфигурации серверов

Для функционирования системы необходимо использовать:

- сервер для подсистемы управления контентом;
- сервер для сервера приложений.

#### 3.1.1 Сервер подсистемы управления контентом

№ п/п	Наименование параметра	Рекомендуемые требования	Минимальные требования
1.	Процессор (CPU)	16 ядер	4 ядра
2.	Память (RAM)	64Гб	16Гб
3.	Диск (HDD)	5Тб	2Тб

#### 3.1.2 Сервер для сервера приложений

№ п/п	Наименование параметра	Рекомендуемые требования	Минимальные требования
1.	Процессор (CPU)	8 ядер	4 ядра
2.	Память (RAM)	64Гб	16Гб
3.	Диск (HDD)	3Тб	1Тб

### 3.2 Операционная система

Debian 11

### 3.3 Состав дистрибутива системы

#### 3.3.1 Общее ПО

№ п/п	Наименование ПО	Лицензия	Информация о дистрибутиве
1.	MySQL 8	GNU GPL 2	Репозиторий MySQL см. п. 4.1.1
2.	Apache 2	Apache 2.0	Входит в репозиторий Debian
3.	Redis 5	BSD	Входит в репозиторий Debian
4.	ElasticSearch 6	Apache 2.0	Репозиторий ElasticSearch см. п. 4.1.4
5.	Magento 2	Open Software License 3.0	Используется доработанная в рамках проекта сборка ПО Magento 2

### 3.3.2 Прикладное ПО

#### Состав комплекта дистрибутива ПО Системы

№ п/п	Наименование ПО	Файл дистрибутива	Описание
<b>1.</b>	<b>Подсистема управления контентом</b>		
1.1.	Magento 2	magento2_svoe_rodnoe.tar.gz	Доработанная в рамках проекта сборка ПО Magento 2
1.2.	БД основного хранилища	sr_db.sql.gz	Дамп тестовой БД для MySQL
<b>2.</b>	<b>Сервер приложений</b>		
2.1.	storefront-api	storefront-api.tar.gz	Доработанный компонент vue-storefront-api. API сервера приложений для веб интерфейса продавца
2.2.	storefront	storefront.tar.gz	Доработанный компонент vue-storefront Прикладные функции сервера приложений для веб интерфейса продавца
2.3.	fresh-proxy	fresh-proxy.tar.gz	API сервера приложений для мобильного приложения и веб интерфейса продавца
2.4.	nuxt-front	nuxt-front.tar.gz	Прикладные функции сервера приложений для веб интерфейса покупателя
<b>3</b>	<b>Мобильное приложение покупателя</b>	<b>svoe_rodnoe.apk</b>	

## 4 УСТАНОВКА И РАЗВЕРТЫВАНИЕ СИСТЕМЫ

### 4.1 Установка подсистемы управления контентом

Установка подсистемы управления контентом выполняется на сервер для данной подсистемы.

#### 4.1.1 Установка СУБД MySQL

1. Для установки необходимо подключить репозиторий MySQL .

```
sudo wget -O- http://repo.mysql.com/RPM-GPG-KEY-mysql-2022 | gpg --dearmor | sudo tee /usr/share/keyrings/mysql.gpg  
echo 'deb [signed-by=/usr/share/keyrings/mysql.gpg] http://repo.mysql.com/apt/debian bullseye mysql-8.0' | sudo tee /etc/apt/sources.list.d/mysql.list
```

2. После добавления репозитория обновите кэш пакетов apt и установите пакет сервера MySQL. При этом также будут установлены пакеты для клиента и другие зависимости.

```
sudo apt update  
sudo apt install mysql-server
```

3. Во время установки появится диалоговое окно конфигурации, в котором нужно будет установить пароль пользователя root для MySQL. Введите пароль, а затем подтвердите его. После этого вас предупредят о новой системе аутентификации, на основе SHA256, использующейся в MySQL, нажмите Ok. Выберите плагин аутентификации (оставьте опцию по умолчанию, чтобы использовать рекомендуемый плагин) и нажмите кнопку Enter, чтобы завершить процесс установки.
4. Когда установка пакетов завершится, установщик запустит службу MySQL и настроит для неё автозагрузку. Чтобы убедиться, что служба MySQL работает, проверьте ее состояние с помощью следующей команды.

```
sudo systemctl status mysql
```

5. Для повышения безопасности рекомендуется выполнить сценарий mysql\_secure\_installation.

```
sudo mysql_secure_installation
```

6. Введите пароль пользователя root, который вы задали во время установки пакета. Далее укажите следующие ответы:

```
VALIDATE PASSWORD COMPONENT - N  
Change the password for root ? - N  
Remove anonymous users? - Y  
Disallow root login remotely? - Y  
Remove test database and access to it? - Y
```

```
Reload privilege tables now? - Y
```

## 7. Перейдите в оболочку MySQL

```
mysql -u root -p
```

## 8. Создайте БД magento2, которая будет использоваться для работы Magento 2

```
create database magento2;  
use magento2;  
exit;
```

## 9. Импортируйте БД из приложенной резервной копии (перед этим распакуйте архив с БД из состава дистрибутива в папку ~/public/rshb)

```
mysql -u root -p magento2 < ~/distrib/sr_db.sql
```

## 10. Установите права на БД magento2 для пользователя root

```
create user 'root'@'%' identified by 'ПАРОЛЬ_ROOT';  
grant all privileges on *.* to 'root'@'%' with grant option;  
flush privileges;
```

### 4.1.2 Установка веб сервера Apache2

#### 1. Установка веб сервера Apache2 выполняется из стандартного репозитория Debian

```
sudo apt install apache2
```

#### 2. Чтобы убедиться, что служба Apache2 работает, проверьте ее состояние с помощью следующей команды.

```
sudo systemctl status apache2
```

*Примечание: по умолчанию Apache2 использует 80 порт, который будет использоваться Magento 2, поэтому настройку портов при инсталляции не проводим.*

### 4.1.3 Установка СУБД Redis

#### 1. Установка СУБД Redis выполняется из стандартного репозитория Debian

```
sudo apt install redis-server
```

#### 2. Чтобы убедиться, что служба Redis работает, проверьте ее состояние с помощью следующей команды.

```
sudo systemctl status redis-server
```

### 4.1.4 Установка ElasticSearch

#### 1. Установите Java, которая предусмотрена в репозитории по умолчанию.



```
sudo apt install default-jre default-jdk
```

## 2. Перейдите в папку distrib и установите Elasticsearch из deb пакета

```
cd ~/distrib/magento  
sudo dpkg -i elasticsearch-7.6.2-amd64.deb
```

## 3. Установите plugin lucene

```
cd /usr/share/elasticsearch/  
sudo bin/elasticsearch-plugin install analysis-icu-7.6.2.zip  
sudo bin/elasticsearch-plugin install file:///home/sr/distrib/analysis-icu-7.6.2.zip
```

4. При необходимости отредактируйте файл конфигурации `/etc/elasticsearch/elasticsearch.yml` (в данном случае при инсталляции оставляем настройки по умолчанию и данный пункт пропускаем).
5. Создайте папку `/etc/elasticsearch/hunspell/ru_RU` и скопируйте в нее файлы словарей `ru_RU.dic` и `ru_RU.aff`

```
sudo -s  
cd /etc/elasticsearch/  
mkdir hunspell  
cd hunspell  
mkdir ru_RU  
exit  
cd ~/distrib/magento  
sudo cp ru_RU.* /etc/elasticsearch/hunspell/ru_RU
```

## 6. Установите права на папку

```
sudo chown elasticsearch:elasticsearch /etc/elasticsearch
```

## 7. Запустите службу Elasticsearch

```
sudo systemctl daemon-reload  
sudo systemctl enable elasticsearch.service  
sudo systemctl start elasticsearch.service
```

8. Чтобы убедиться, что служба Elasticsearch работает, проверьте ее состояние с помощью следующей команды.

```
sudo systemctl status elasticsearch
```

9. Проверьте доступность Elasticsearch по http, для этого в браузере перейдите по адресу `http://localhost:9200` . Должна отразиться информация об установленном Elasticsearch.

### 4.1.5 Установка Magento 2

#### 1. Установите необходимые зависимости

```
sudo apt install curl git unzip
```

```
sudo apt install php-cli php-mbstring php-curl libapache2-mod-php7.4
sudo apt install php7.4-dom php7.4-pdo php7.4-mysql php7.4-intl php7.4-bcmath php7.4-bz2 php7.4-gd php7.4-phar php7.4-zip
```

2. Composer предоставляет написанный на PHP инсталлятор. Перейдите в папку `distrib` и извлеките инсталлятор при помощи `curl`

```
cd ~/distrib
curl -sS https://getcomposer.org/installer -o composer-setup.php
```

3. Перейдите на страницу <https://composer.github.io/pubkeys.html> Скопируйте хэш и поместите его в переменную оболочки (вместо ... необходимо указать скопированный хэш)

```
HASH=...
export HASH
```

4. Запустите скрипт PHP, чтобы убедиться, что хэш SHA-384 пакета совпадает с хэшем инсталлятора.

```
php -r "if (hash_file('SHA384', 'composer-setup.php') === '$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

Если значения совпали, команда вернёт:

```
Installer verified
```

5. Выполните команду, которая загрузит пакет менеджера Composer и выполнит его глобальную установку в `/usr/local/bin`

```
sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

6. Чтобы убедиться в том, то установка прошла успешно, запустите

```
composer
```

7. Извлеките Magento 2 из файла дистрибутива в папку `/var/www/html/`

```
sudo tar -xzf magento2_svoe_rodnoe.tar.gz -C /var/www/html/
```

8. Укажите пользователя БД `mysql root` и его пароль в `app/etc/env.php`

```
sudo nano /var/www/html/magento/magento/app/etc/env.php
```

9. Скопируйте файл `magento.conf` в `/etc/apache2/sites-available`

```
sudo cp ~/distrib/magento/magento.conf /etc/apache2/sites-available/
```

10. Установите права для папки с `magento`

```
- sudo chown -R www-data:www-data /var/www/html/magento
- sudo chmod -R 755 /var/www/html/magento
```

11. В корневой директории проекта `/var/www/html/magento/magento` выполните установку зависимостей через `композер`

```
sudo composer install --ignore-platform-reqs
```

## 12. Выполните установку, последовательно выполнив команды

```
cd /var/www/html/magento/magento
sudo -s
bin/magento setup:upgrade --keep-generated
bin/magento cache:clean
bin/magento cache:flush
bin/magento setup:di:compile
bin/magento setup:static-content:deploy ru_RU -f
bin/magento setup:static-content:deploy -f
bin/magento indexer:reindex
exit
```

13. Проверьте доступность административной части Magento 2 в браузере перейдя по ссылке <https://localhost/admin> . Должно отразиться окно входа в административную часть Magento2.

## 4.2 Установка сервера приложений

Установка сервера приложений выполняется на сервер для данной подсистемы.

### 1. Установите curl

```
sudo apt install curl
```

### 2. Выполните установку nvm

```
curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
```

### 3. Установка storefront-api

#### a. Установите Node версии 12.20.2

```
nvm install 12.20.2  
nvm use 12.20.2
```

#### b. Создайте папку для storefront-api

```
cd ~/Public  
mkdir rshb  
cd rshb  
mkdir storefront-api
```

#### c. Извлеките storefront-api из файла дистрибутива и перейдите в папку для storefront-api

```
tar -xzf ~/distrib/storefront-api.tar.gz -C ~/Public/rshb/  
cd ~/Public/rshb/storefront-api/
```

#### d. Проверьте корректность настроек в файле config/local.json (все необходимые настройки для корректной работы указаны в файле, который входит в дистрибутив).

#### e. Установите storefront-api, выполнив следующие команды

```
npm i yarn -g  
yarn  
yarn build  
yarn start
```

#### f. Для проверки работоспособности storefront-api, перейдите в браузере по адресу <http://localhost:8080/api>. В результате должна открыться страница со следующим содержимым:

```
{"version":"1.0.0-rc.2"}
```

### 4. Установка fresh-proxy

#### a. Установите Node версии 16.3.0.

```
nvm install 16.3.0  
nvm use 16.3.0
```

#### b. Создайте папку для fresh-proxy

```
cd ~/Public/rshb
mkdir fresh-proxy
```

- c. Извлеките fresh-proxy из файла дистрибутива и перейдите в папку fresh-proxy

```
tar -xzf fresh-proxy.tar.gz -C ~/public/rshb/fresh-proxy/
cd ~/Public/rshb/fresh-proxy
```

- d. В файле .env укажите пароль для доступа к БД, который был создан при установке MySQL
- e. Установите fresh-proxy, выполнив следующие команды

```
npm i yarn -g
yarn
yarn start:prod
```

- f. Для проверки работоспособности fresh-proxy, перейдите в браузере по адресу <http://localhost:3003/api>. В результате должна открыться страница со Swagger документацией к API

## 5. Установка nuxt-front

- a. Установите Node версии 14.19.0 (nvm была установлена ранее).

```
nvm install 14.19.0
nvm use 14.19.0
```

- b. Создайте папку для nuxt-front

```
cd ~/Public/rshb
mkdir nuxt-front
```

- c. Извлеките nuxt-front из файла дистрибутива и перейдите в папку nuxt-front

```
tar -xzf nuxt-front.tar.gz -C ~/public/rshb/nuxt-front/
cd ~/Public/rshb/nuxt-front
```

- d. Проверьте корректность настроек в файле .env (все необходимые настройки для корректной работы указаны в файле, который входит в дистрибутив).
- e. Установите nuxt-front, выполнив следующие команды

```
npm i yarn -g
yarn
yarn build
-- на вопрос - Are you interested in participating? - выбираем n
yarn start
```

- f. Для проверки работоспособности nuxt-front, перейдите в браузере по адресу <http://localhost:3001/fresh>. В результате должна открыться главная страница маркетплейса «Свое Родное».

## 6. Установка storefront

- a. Активируйте Node версии 12.20.2

```
nvm use 12.20.2
```

- b. Создайте папку для storefront

```
cd ~/Public/rshb  
mkdir storefront
```

- c. Извлеките storefront из файла дистрибутива и перейдите в папку storefront

```
tar -xzf storefront.tar.gz -C ~/public/rshb/storefront/  
cd ~/Public/rshb/storefront
```

- d. Проверьте корректность настроек в файле config/local.json (все необходимые настройки для корректной работы указаны в файле, который входит в дистрибутив).

- e. Установите storefront, выполнив следующие команды

```
npm i yarn -g  
yarn  
yarn build  
yarn start
```

- f. Для проверки работоспособности storefront, перейдите в браузере по адресу <http://localhost:3000/category>. В результате должна открыться страница с каталогом продуктов.

Для запуска всех модулей после перезагрузки системы выполните скрипт sr-server-start.sh из папки sr-server-start дистрибутива

```
/home/sr/distrib/sr-server-start/sr-server-start.sh
```

Дополнительно для удобства работы может быть установлено следующее ПО:

- Chrome и дополнение ElasticSearch Head - для просмотра данных в ElasticSearch;
- DBEaver (или иной DB клиент) - для просмотра данных в MySQL.

## 5 УСТАНОВКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОКУПАТЕЛЯ

Мобильное приложение для покупателя предоставляет те же возможности, что и веб-приложение, а именно: выбор интересующих его товаров, оформление заказа, оплата и отслеживание выполнения своих заказов.

Мобильное приложение реализовано с использованием фреймворка Flutter (фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux с использованием языка программирования Dart) и взаимодействует с REST API сервера приложений.

Опубликованы версии мобильного приложения для iOS (поддерживаются iOS 12.0 и более поздние версии) и Android (поддерживаются Android 4.4W и более поздние версии).

Для установки приложения необходимо перейти на смартфоне в App Store (для iOS) или в Google Play (для Android), выполнить поиск приложения по словам «свое родное» и установить приложение..

После того, как приложение будет установлено и открыто пользователю будет представлена процедура первого запуска приложения (онбординг) на четвертом шаге которого предусмотрена возможность выбора местоположения покупателя (см. Рисунок 2).

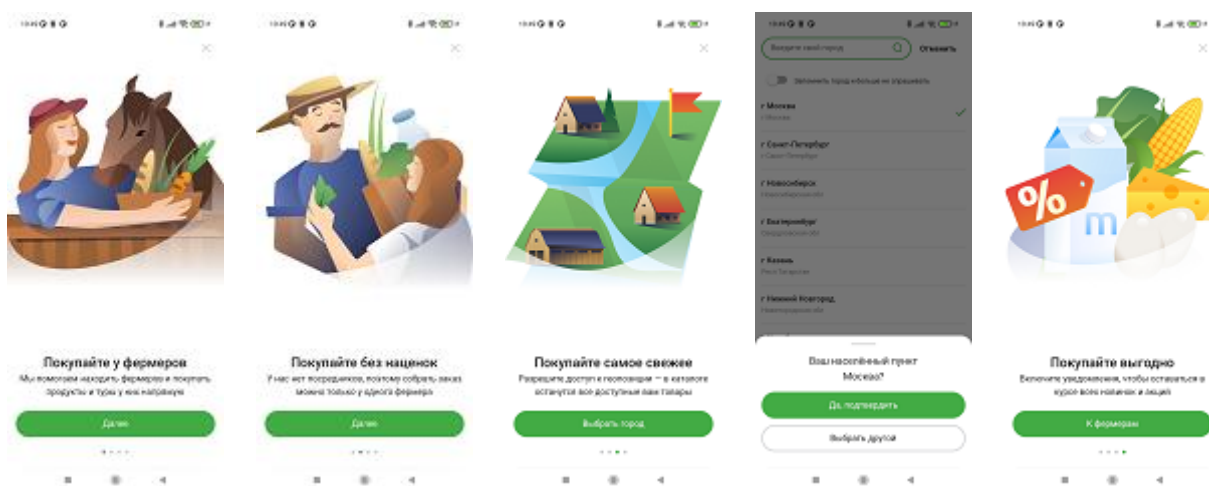


Рисунок 2

После прохождения всех экранов онбординга будет открыта главная страница приложения.