

Инструкция по установке и запуску Навигатора

Структура Навигатора

Навигатор состоит из двух частей: Site и Booking, и нескольких инфраструктурных приложений.

Site реализует работу веб-сайта Навигатора (запись детей на учебные программы – для родителей). Booking – функционал личного кабинета (публикация учебных программ и управление системой – для организаторов программ и администраторов).

Основные шаги

- 1 Подготовка к установке Навигатора
 - 1.1 Соблюсти требования к аппаратному обеспечению и операционной системе
 - 1.2 Зарегистрировать домены
 - 1.3 Создать ящики электронной почты
 - 1.4 Получить API-ключ DaData.ru
 - 1.5 Создать облачное хранилище для ресурсов Навигатора
 - 1.6 [Опционально] Создать рабочее пространство в Slack, получить Webhook URL
- 2 Развертывание Навигатора
 - 2.1 Скопировать и распаковать архив на хост-машину
 - 2.3 Указать в файле settings.ini настройки для конфигурирования Навигатора
 - 2.4 Запустить файл conf_maker_centos.py
 - 2.5 Внести список муниципалитетов в conf/dicts/municipality.json
 - 2.6 Запустить установочный скрипт `***-navi-installer-py.sh`
 - 2.7 Дождаться окончания развертывания

1 Подготовка к установке Навигатора

1.1 Соблюсти требования к аппаратному обеспечению и операционной системе

На хост-машине должна быть установлена операционная система CentOS 7 с сетевым доступом (открыты порты 80 и 443 для доступа к машине по протоколам http и https, порт 22 – для ssh) или другая операционная система семейства Линукс, например, Астра, Роса, Альт.

1.2 Зарегистрировать домены

Для работы Навигатора требуется, чтобы на его хост-машину были направлены два разных доменных имени.

Это могут быть два разных домена,

Примеры:

- <http://www.dopportal.ru> – веб-сайт (для родителей) + <http://booking.dopportal.ru> – часть для организаторов и "админка"
- <http://navigator.ru> – веб-сайт (для родителей) + <http://adminka.ru> – часть для организаторов и "админка"

либо домен с поддоменом.

Примеры:

- <http://dopportal.ru> – веб-сайт (для родителей) + <http://booking.dopportal.ru> – часть для организаторов и "админка"
- <http://navi.ru> – веб-сайт (для родителей) + <http://adminka.navi.ru> – часть для организаторов и "админка"

1.3 Создать ящики электронной почты

Для работы Навигатора необходимо завести ящики доменной почты:

- для отправки уведомлений пользователям, организаторам, администраторам (например, noreply@имя_домена);
- куда будут приходить копии всех писем домена (report.copу@имя_домена);
- служба поддержки Навигатора (support@имя_домена).
- для встроенного пользователя-администратора в букинге (admin@имя_домена).

Почтовый сервер можно использовать как свой, так и сторонний. Мы рекомендуем воспользоваться сервисом Яндекс.ПДД (почта для доменов) https://pdd.yandex.ru/domains_add/

1.4 Получить API-ключ DaData.ru

Навигатор использует сервис DaData.ru для автозавершения ввода и унификации хранения адресов. Необходимо зарегистрироваться по адресу <https://dadata.ru/api/clean/>, подтвердить регистрацию аккаунта и получить API-ключ к сервису.

1.5 Создать облачное хранилище для ресурсов Навигатора

Для хранения данных Навигатор использует хостинг со SWIFT OpenStack (протокол, по которому работают различные хостинг-провайдеры, например, selectel.ru).

Для работы Навигатора на хостинге необходимо создать облачный сервер, на котором завести два контейнера:

- images – для хранения изображений;
- resourcers – для хранения загружаемых пользователями файлов.

Для развертывания Навигатора нужно получить настройки для доступа к облачному серверу и URL контейнеров.

1.6 [Опционально] Создать рабочее пространство в Slack, получить Webhook URL

Система Навигатор умеет по webhook отправлять запросы с отладочной информацией (в случае возникновения ошибок).

Webhook можно использовать, например, в связке с мессенджером Slack (<https://api.slack.com/incoming-webhooks>).

2 Развертывание Навигатора

2.1 Скопировать и распаковать архив на хост-машину

Команда для распаковки архива: `tar -xf 01.Distributive.tar.gz`

Установочный архив 01.Distributive.tar.gz состоит из:

- 1) code/ – папка с программным кодом Навигатора: Booking + Site;
- 2) confs/ – папки конфигурационных файлов (конфигов) для запуска приложений;
- 3) dockerfiles/ – папки с файлами для сборки docker-образов, необходимых для работы приложений;
- 4) conf_maker_centos.py – файл со скриптом для подготовки конфигурационных файлов;
- 5) settings.ini – файл настроек для подготовки конфигурационных файлов;
- 6) navi-installer-py.sh – файл с установочным скриптом Навигатора.

2.3 Указать в файле settings.ini настройки для конфигурирования Навигатора

Необходимо открыть файл settings.ini в редакторе и заполнить параметры.



Внимание!

Уделите внимание правильности вводимых данных для настроек. Из-за неверных настроек Навигатор может работать некорректно.

1. Application – настройки приложения
 - a. bookingURL – доменное имя для букинга (Booking).
 - b. siteURL – Доменное имя для сайта (Site).
 - c. dadataKey – API-ключ, полученный после регистрации в сервисе dadata.ru
 - d. WebhookURL – URL для подключения Slack-канала оповещения об ошибках.
2. DataBase – настройки для БД
 - a. dbRegionname – название региона латинскими буквами.
 - b. dbUserPassword – пароль для доступа к БД Навигатора. Имя пользователя БД по умолчанию устанавливается как "root".
3. Email – настройки электронной почты Навигатора
 - a. emailSmtServer – адрес почтового сервера Навигатора.
 - b. emailSmtPort – порт почтового сервера.
 - c. emailPassword – пароль к почтовому ящику.
 - d. emailNoreply – адрес почтового ящика, с которого будут отправляться уведомления пользователям.
 - e. emailReportCory – адрес ящика, куда будут приходить копии всех писем.

- f. emailSupport – адрес службы поддержки Навигатора.
- 4. Salt – соли для паролей. Введите 4 разные произвольные последовательности. Рекомендуем длину 13 символов, строчные и прописные буквы латинского алфавита и цифр от 0 до 9.
 - a. siteSaltStart.
 - b. siteSaltFinish.
 - c. bookingSaltStart.
 - d. bookingSaltFinish.
- 5. Swift – настройки для авторизации и работе по протоколу SWIFT, полученные от хост-провайдера/
 - a. OS_AUTH_URL – URL точки входа на хостинге
 - b. OS_TENANT_ID – tenant ID
 - c. OS_TENANT_NAME – tenant name.
 - d. OS_USERNAME – Имя учетной записи
 - e. OS_PASSWORD – Пароль к учетной записи
 - f. OS_REGION_NAME

2.4 Запустить файл conf_maker_centos.py



Внимание!

Скрипт следует запускать на хост-машине с CentOS 7, из-под пользователя с sudo-правами.

Команда для запуска: `python conf_maker_centos.py`

Скрипт предложит проверить настройки. Если все правильно, но в ответ на вопрос введите 'yes'. Если нужно что-то поменять – введите 'no'. Скорректируйте settings.ini и запустите conf_maker_centos.py еще раз.

После того как конфигурационные файлы будут готовы имя файла settings.ini изменится на имя файла с указанным именем региона. А файл с установочным скриптом navi-installer-py.sh станет называться `***-navi-installer-py.sh`, где `***` – указанное имя региона.

2.5 Внести список муниципалитетов в confs/dicts/municipality.json

Файл уже содержит шаблон для оформления списка муниципалитетов. Муниципалитеты нужно добавлять в файл строго по указанному шаблону.

2.6 Запустить установочный скрипт ***-navi-installer-py.sh



Внимание!

Скрипт следует запускать на хост-машине с CentOS 7, из-под пользователя с sudo-правами.

Команда для запуска: `chmod +x ***-navi-installer-py.sh && ./***-navi-installer-py.sh`,

где `***` – имя региона.

2.7 Дождаться окончания развертывания

Установочный скрипт устанавливает docker, создаёт виртуальную docker-сеть, docker-контейнеры с приложениями, дерево каталогов для работы системы; затем копирует код, пробрасывает доступ к необходимым папкам в контейнеры, поднимает контейнер с СУБД и интерпретатором PHP, выполняет разметку таблиц и импортирует в БД данные из справочников, а затем поднимает остальные контейнеры.

Функционирование приложения

Развернутый Навигатор представляет собой комплекс из приложений, запущенных в docker-контейнерах (см. табл. 1). Контейнеры взаимодействуют между собой по виртуальной сети с именем "mysql". Маршрутизация между контейнерами выполняется по их именам.

Команда для получения списка имеющихся контейнеров: `sudo docker ps`

Таблица 1 – Приложения Навигатора

№	Приложение	Имя контейнера
1	СУБД	percona
2	Booking под управлением cron	php-fpm

3	Site на node.js под управлением pm2	pm2
4	nginx (проху-взаимодействие между пользователями и приложениями Навигатора)	nginx
5	sphinx (полнотекстовый поиск и стандартизация данных)	sphinx

Контейнеры поднимаются в отказоустойчивом режиме: падение приложения приведёт к перезапуску контейнера. При этом сам сервис докера добавляется в автозагрузку, что гарантирует, что контейнеры поднимутся даже после перезагрузки (в том числе аварийной) хост-машины.