

Руководство по установке, настройке и эксплуатации программного комплекса

1. Назначение и область применения

Данный документ содержит полный цикл инструкций по установке, настройке и эксплуатации программной системы, реализованной по архитектуре клиент-сервер. Документ предназначен для предоставления в составе регистрационного пакета отечественного ПО.

1.1. Описание системы

Система состоит из двух основных компонентов:

- **Серверная часть** (NodeJS + Express + TypeDI), отвечающая за бизнес-логику, интеграцию с внешними сервисами, работу с базами данных, предоставление API.
- **Клиентская часть** (Vue 2 + Quasar v1), реализующая web-интерфейс, через который конечный пользователь взаимодействует с системой.

Вся серверная и клиентская функциональность развёртывается на собственной инфраструктуре заказчика без вовлечения зарубежных облачных сервисов. Все hosting/хранение компонентов и исходных кодов производится на территории РФ.

2. Архитектурная схема и структура приложения

Система реализует двухзвенную (клиент-серверную) архитектуру с проксированием запросов и возможностью горизонтального масштабирования серверной части, а также поддержкой резервных (слейв) серверов.

Основные компоненты:

- Сервер NodeJS (Express, TypeDI) запущен под управлением PM2; обслуживает HTTP API, фоновые задачи, интеграции с внешними сервисами;
- Клиентское SPA-приложение на базе Vue 2 + Quasar v1, собирается в статические файлы и публикуется через Apache2, либо может также работать через встроенные реализации NodeJS;

- Две СУБД — MySQL (реляционная, основное хранение данных) и MongoDB (NoSQL, для спец. data-feature);
- Apache2 — для обратного проксирования, безопасного SSL и выдачи фронтенда;
- Файл конфигурации каждой части описан через `.env`;
- Хранение исходного кода (репозиторий) и артефактов осуществляется на сервере организации или в сертифицированной частной облачной инфраструктуре на территории РФ.

3. Требования к программным и техническим средствам

3.1. Сервер

- ОС: Любая *nix-система (Linux Ubuntu 20.04+ рекомендуется); либо современная Windows Server
- NodeJS: version 16 LTS+
- MySQL: version 8.0+
- MongoDB: version 5.0+
- Apache2: version 2.4+
- PM2: для управления процессами
- ОЗУ: от 4 ГБ, при увеличении числа воркеров — линейное увеличение потребления (например, 1 воркер $\approx$ 500 МБ RAM)
- Хранилище: от 20 ГБ SSD/HDD
- Сетевая инфраструктура: открытые порты (HTTP API, MySQL, MongoDB, Apache2)

3.2. Рабочие места пользователей

- Любая ОС (Windows, MacOS, Linux)
- Современный браузер: Google Chrome 90+, Mozilla Firefox 90+, MS Edge 90+
- Не требуется установка дополнительного ПО — только доступ к браузеру

4. Порядок и этапы установки

4.1. Установка серверной части

1. Подготовка серверной инфраструктуры:

- Установить NodeJS 16+, MySQL 8+, MongoDB 5+, Apache2
- Разрешить необходимые порты, настроить firewall

2. Размещение артефактов

- Поместить исходные и скомпилированные файлы сервера в каталог (например, /opt/api)
- Подраздел /generated содержит артефакты клиентского кода, генерируемые серверной частью.

3. Установка зависимостей:

- Перейти в директорию сервера, выполнить `yarn install` или `npm install`

4. Конфигурирование

- Создать и заполнить файл `.env` в корне сервера (см. раздел 6)

5. Регистрация сервисов:

- Подготовить файл `ecosystem.config.cjs` в корне проекта (пример входит в поставку)
- Запустить сервис через `pm2` командой:

```
pm2 start ecosystem.config.cjs --only server
```

- Для автозапуска настроить `pm2 startup` (описано по ссылке <https://pm2.keymetrics.io/docs/usage/startup/>)

6. Настройка ProxyPass:

- Apache2 выступает фронт-прокси для API и клиентских запросов, пример настройки:

```
ProxyPass /api http://localhost:8003/  
ProxyPassReverse /api http://localhost:8003/
```

4.2. Установка клиентской части

1. Подготовка окружения:

- Установить NodeJS (только для сборки). Сборка должна производиться на защищённом сервере организации.

2. Размещение файлов:

- Поместить клиентский исходный код в отдельный каталог, например, `/opt/client`
- Переместить сгенерированный сервером код из `/generated` сервера в подкаталог `/api` клиента (этот код обеспечивает динамическую интеграцию с сервером)

3. Установка зависимостей и сборка:

```
yarn install  
yarn build
```

В результате файлы фронтенда окажутся в папке `/opt/client/dist/spa`

4. Деплой:

- Скопировать содержимое `/opt/client/dist/spa` в каталог, обслуживаемый Apache2 (например, `/var/www/html`)

5. Конфигурирование:

- Создать файл `.env` с единственным параметром:

```
API_URL=http(s)://<адрес_сервера>:<порт>
```

- Для управления клиентским сервисом используется тот же `pm2` и файл `ecosystem.config.cjs`:

```
pm2 start ecosystem.config.cjs --only client
```

5. Системные переменные и структура конфигурации

5.1. Конфиг клиента (`.env`)

- `API_URL` — сетевой адрес и порт серверной части, по которому клиент будет обращаться к API

5.2. Конфиг сервера (.env)

```
PORT=8003
DB_HOST=127.0.0.1
DB_SYNC=true
DB_NAME=omega
DB_PORT=3306
DB_USER=root
DB_PASSWORD=
DB_REFRESH_COMPUTED=1
MEGAFON_AUTH_KEY=
MEGAFON_OUTER_AUTH_KEY=
TELEGRAM_BOT_TOKEN=
TELEGRAM_SCHEDULER_MINUTES=5
WORKER_THREADS=1
NOMINATIM_API_URL=
VK_ADS_CLIENT_ID=
VK_ADS_CLIENT_SECRET=
STOP_CODEGEN=0
YANDEX_GEOCODER_API_KEY=
AI_URL=
AI_SECRET=
LLM_URL=
LLM_TOKEN=
ENABLE_DAILY_DUMPS=0
ENABLE_SHORT_DUMPS=0
IS_SLAVE=
SLAVES=
```

Обязательные параметры:

- Данные доступа к MySQL, количество воркеров, адрес nominatim (URL)

Опциональные:

- Интеграции с внешними сервисами (MEGAFON, TELEGRAM, VK ADS), параметры резервирования и дампов, доп. сервисы (AI, LLM)
- IS_SLAVE и SLAVES — для настройки раздачи записи на резервные серверы (слейвы)

6. Описание процесса поддержки и жизненного цикла

6.1. Техническая поддержка

- Все работы по эксплуатации, доработке и поддержке проводятся штатной ИТ-командой организации.
- Контакты технической поддержки:

E-mail: info@insgt.ru

Телефон: [+7 \(926\) 000 41 04](tel:+7(926)0004104)

6.2. Процедуры обновления и устранения неисправностей

- Все обновления системы и компонентов осуществляются авторами либо официальными ответственными за сопровождение.
- Рекомендуется выполнять резервное копирование БД перед каждым обновлением.
- Установка обновлений выполняется через `git pull` (или иные процедуры), повторный запуск `yarn install && pm2 restart`.
- В случае сбоев обращаться в службу поддержки, предоставив логи ошибок из каталога `/logs`

6.3. Контроль версий исходного кода

- Исходный код, конфигурация и артефакты хранятся в системе контроля версий Git, установленной на сервере организации, находящемся на территории РФ.
- Все компоненты компилируются локально, без передачи исходных кодов за рубеж или использования зарубежных CI/CD SaaS-сервисов.

7. Технические средства хранения и компиляции

- Хранение исходного кода — частный Git-сервер (или локальный репозиторий), размещённый в офисе/дата-центре организации.
- Хранилище компиляции: защищённые серверы в РФ.
- Компиляция и сборка (`yarn build` / `npm run build`) производятся только в инфраструктуре организации. Для клиента и сервера необходим только установленный NodeJS требуемой версии.

8. Проверка и эксплуатация

- Проверить запусженность backend — `curl http(s)://localhost:8003/api/ping`
- Проверить выдачу фронтенда — открыть клиентский адрес в браузере, убедиться в доступности главной страницы.
- Проверить авторизацию и обмен данными между клиентом и сервером.
- Все дальнейшие операции проводятся только внутри организации.

9. Безопасность

- Вся инфраструктура хранится и обслуживается в РФ. Не используются облака, не сертифицированные Минцифрой.
- Все конфигурационные файлы содержат только внутренние адреса или приватные данные.