

Общество с ограниченной ответственностью «УМАРТА»

Сервис для автоматического перевода документов с использованием LLM-моделей
«УМАРТА.Переводчик»

Инструкция по установке и развертыванию ПО

Листов 16

Москва, 2025 г.

АННОТАЦИЯ

Настоящий документ представляет собой инструкцию по установке, настройке и эксплуатации программного обеспечения «Сервис для автоматического перевода документов с использованием LLM-моделей "УМАРТА.Переводчик"».

Документ охватывает следующие аспекты:

- описание компонентов системы и требований к аппаратному и программному обеспечению;
- пошаговая инструкция по установке, запуску и первичной проверке работоспособности;
- настройка LLM-провайдеров (OpenAI, DeepSeek, LM Studio, GigaChat), а также интеграция email-сервиса (в частности, VK почта);
- рекомендации по обновлению системы, резервному копированию и обеспечению безопасности, включая настройку HTTPS и подготовку к продакшен-развёртыванию;
- средства мониторинга и диагностики, просмотр логов, проверка состояния системы;
- раздел по устранению возможных неисправностей и ссылки на вспомогательные ресурсы.

Инструкция предназначена для системных администраторов и технических специалистов, осуществляющих развёртывание и поддержку Системы.

СОДЕРЖАНИЕ

1.1	ТРЕБОВАНИЯ К КВАЛИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ	4
1.2	КОМПОНЕНТЫ СИСТЕМЫ.....	4
1.3	ТРЕБОВАНИЯ К ТЕХНИЧЕСКОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	4
2.1	Подготовка к установке.....	6
2.1.1	Получение исходного кода.....	6
2.1.2	Настройка переменных окружения	6
2.2	Запуск Системы	7
2.3	Создание администратора	7
2.4	Проверка работоспособности системы	7
3.1	Настройка LLM-провайдеров	8
3.1.1	DeepSeek	8
3.1.2	OpenAI	8
3.1.3	LM Studio (локальный)	8
3.1.4	GigaChat.....	9
3.2	Настройка EMAIL-сервиса.....	9
	Система поддерживает любой SMTP сервер с шифрованием SSL/TLS.	9
4.1	Обновление Системы.....	11
4.2	Резервное копирование	11
4.2.1	База данных	11
4.2.2	Файлы пользователей.....	11
5.1	Просмотр логов Системы	12
5.2	Проверка состояния Системы	12
6.1	Рекомендации для продакшена	13
6.2	Настройка HTTPS	13
8.1	Вспомогательные ресурсы.....	15

1 ОБЩИЕ СВЕДЕНИЯ

Полное наименование: Сервис для автоматического перевода документов с использованием LLM-моделей «УМАРТА.Переводчик».

Краткое наименование: Система, УМАРТА.Переводчик.

1.1 ТРЕБОВАНИЯ К КВАЛИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ

Системный администратор должен обладать следующими навыками и знаниями:

- Работа с терминалом;
- Опыт работы с контейнеризацией и инструментами оркестрации (Docker, Docker Compose);
- Знание принципов работы веб-серверов (например, Nginx) и умение их настраивать;
- Понимание основ работы с базами данных (PostgreSQL), включая резервное копирование и восстановление;
- Навыки работы с переменными окружения и конфигурационными файлами;
- Опыт настройки и администрирования LLM-сервисов или взаимодействия с внешними API (например, OpenAI, DeepSeek);
- Базовые знания в области информационной безопасности: настройка HTTPS, управление доступом, защита конфигураций;
- Умение анализировать системные логи, проводить диагностику и устранение неисправностей;
- Желательно: опыт работы с почтовыми сервисами (SMTP), настройкой email-уведомлений и интеграций.

1.2 КОМПОНЕНТЫ СИСТЕМЫ

Система состоит из следующих основных компонентов (Таблица 1):

Таблица 1 –Компоненты Системы

СЕРВИС	КОНТЕЙНЕР	ПОРТ	НАЗНАЧЕНИЕ
nginx	nginx	80	Веб-сервер, обратный прокси
backend	backend	8000	FastAPI приложение
frontend	frontend_builder	-	Сборка React приложения
db	postgres_db	5432	База данных PostgreSQL

1.3 ТРЕБОВАНИЯ К ТЕХНИЧЕСКОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Требования к минимальным характеристикам технических средств:

- Не менее 4 ГБ свободной оперативной памяти;
- Не менее 10 ГБ свободного пространства на диске;
- Свободные порты:
 - 80 – для веб-доступа (nginx);
 - 5432 – для подключения к базе данных PostgreSQL.

Требование к программному обеспечению на пользовательском устройстве:

- Наличие установленных компонентов:
 - Docker версии 20.10 и выше;
 - Docker Compose версии 2.0 и выше.

2 УСТАНОВКА И ЗАПУСК

2.1 Подготовка к установке

2.1.1 Получение исходного кода

Перед установкой необходимо получить исходный код проекта, выполнив клонирование репозитория:

```
git clone https://github.com/<yourusername>/УМАРТА.Переводчик.git
cd УМАРТА.Переводчик
```

2.1.2 Настройка переменных окружения

Для корректной работы Системы необходимо создать конфигурационный файл `.env`, содержащий переменные окружения:

```
cp env.example .env
```

Ниже приведён список обязательных настроек, которые требуется указать в файле `.env`:

- База данных:

```
POSTGRES_USER=dbuser
POSTGRES_PASSWORD=secure_password_here
POSTGRES_DB=umarta_translator
DATABASE_URL=postgresql://dbuser:secure_password_here@db:5432/umarta_translator
```

- JWT (необходимо сгенерировать случайный ключ):

```
SECRET_KEY=<your-very-secure-secret-key-32-chars-min>
ACCESS_TOKEN_EXPIRE_MINUTES=15
REFRESH_TOKEN_EXPIRE_DAYS=7
```

- Email (для уведомлений):

```
SMTP_HOST=smtp.mail.ru
SMTP_PORT=587
SMTP_USER=<your-email-username>
SMTP_PASSWORD=<your-app-password>
EMAILS_FROM_EMAIL=<your-email@domain.name>
EMAILS_FROM_NAME=Translator
```

- LLM-провайдер (необходимо настроить как минимум одного провайдера):

```
DEEPEEK_ENABLED=true
DEEPEEK_API_KEY=<your-deepeek-api-key>
– Домены:
```

```
SERVER_DOMAIN=localhost
SERVER_HOST=http://localhost
CLIENT_HOST=http://localhost
– IOKassa (для платежей):
```

```
YOOKASSA_SHOP_ID=<your-shop-id>
YOOKASSA_SECRET_KEY=<your-secret-key>
```

```
YOOKASSA_RETURN_URL=http://localhost/payment-success
```

2.2 ЗАПУСК СИСТЕМЫ

Система запускается с использованием Docker Compose. Убедитесь, что в текущей директории находятся файлы `docker-compose.yml` и `.env`.

Запуск всех сервисов осуществляется командой `docker-compose up -d` из рабочей папки.

Для проверки текущего состояния контейнеров выполните команду:

```
docker-compose ps
```

2.3 СОЗДАНИЕ АДМИНИСТРАТОРА

Для создания администратора введите следующую команду:

```
docker exec -it backend python -m app.init_admin
```

После запуска команды вам будет предложено ввести необходимые данные для создания пользователя:

- Введите логин: `admin@admin.com`;
- Введите пароль: `admin`.

2.4 ПРОВЕРКА РАБОТОСПОСОБНОСТИ СИСТЕМЫ

Для проверки работоспособности Системы необходимо открыть в браузере ссылку <https://translate.umarta.dev/>.

При успешной загрузке должна отобразиться публичная страница пользовательского интерфейса. Подробное описание возможностей системы и инструкции по работе с ней приведены в Руководстве пользователя.

3 НАСТРОЙКА СИСТЕМЫ

3.1 НАСТРОЙКА LLM-ПРОВАЙДЕРОВ

Программное обеспечение поддерживает интеграцию с несколькими LLM-провайдерами для обработки запросов на основе искусственного интеллекта. Настройка производится через переменные окружения, указанные в конфигурационном файле `.env`.

3.1.1 DeepSeek

Поддерживаемые параметры представлены в таблице 2.

Таблица 2 – Описание переменных DeepSeek

ПЕРЕМЕННАЯ	ОПИСАНИЕ
DEEPEEK_ENABLED	Включение/отключение интеграции с DeepSeek
DEEPEEK_API_KEY	API-ключ, выданный DeepSeek
DEEPEEK_MODEL	Идентификатор модели

Пример конфигурации (`.env`):

```
DEEPEEK_ENABLED=true
DEEPEEK_API_KEY=<your-api-key>
DEEPEEK_MODEL=deepseek-chat
```

3.1.2 OpenAI

Поддерживаемые параметры представлены в таблице 3.

Таблица 3 – Описание переменных OpenAI

ПЕРЕМЕННАЯ	ОПИСАНИЕ
OPENAI_ENABLED	Включение/отключение интеграции с OpenAI
OPENAI_API_KEY	API-ключ, выданный OpenAI
OPENAI_MODEL	Идентификатор модели

Пример конфигурации (`.env`):

```
OPENAI_ENABLED=true
OPENAI_API_KEY=<your-api-key>
OPENAI_MODEL=gpt-4-turbo
```

3.1.3 LM Studio (локальный)

Поддерживаемые параметры представлены в таблице 4.

Таблица 4 – Описание переменных LM Studio

ПЕРЕМЕННАЯ	ОПИСАНИЕ
LMSTUDIO_ENABLED	Включение/отключение интеграции с LM Studio

LMSTUDIO_API_URL	Адрес, по которому Система обращается к LM Studio
LMSTUDIO_MODEL	Идентификатор модели, загруженной в LM Studio

Пример конфигурации (.env):

```
LMSTUDIO_ENABLED=true
LMSTUDIO_API_URL=http://host.docker.internal:1234/v1
LMSTUDIO_MODEL=<your-local-model>
```

3.1.4 GigaChat

Поддерживаемые параметры представлены в таблице 5.

Таблица 5 – Описание переменных LM Studio

ПЕРЕМЕННАЯ	ОПИСАНИЕ
GIGACHAT_ENABLED	Включение/отключение интеграции с GIGACHAT
GIGACHAT_API_KEY	API-ключ, выданный GIGACHAT
GIGACHAT_API_URL	Адрес, по которому Система обращается к GIGACHAT
GIGACHAT_AUTH_URL	Адрес, по которому Система обращается за временным токеном
GIGACHAT_SCOPE	Скоуп моделей
GIGACHAT_MODEL	Идентификатор модели
GIGACHAT_MAX_CONCURRENT	Максимальное количество одновременных запросов к модели
GIGACHAT_TIMEOUT	Пауза между запросами

Пример конфигурации (.env):

```
GIGACHAT_ENABLED=true
GIGACHAT_API_KEY=<your-api-key>
GIGACHAT_API_URL =https://gigachat.devices.sberbank.ru/api/v1
GIGACHAT_AUTH_URL =https://ngw.devices.sberbank.ru:9443/api/v2/oauth
GIGACHAT_SCOPE =GIGACHAT_API_Corp
GIGACHAT_MODEL =GigaChat-2
GIGACHAT_MAX_CONCURRENT=20
GIGACHAT_TIMEOUT=30
```

3.2 НАСТРОЙКА EMAIL-СЕРВИСА

Система поддерживает любой SMTP сервер с шифрованием SSL/TLS.

Например, для настройки SMTP через использование VK почты необходимо выполнить следующие шаги:

1. Сгенерировать специальный пароль для внешних приложений в настройках вашего аккаунта.
2. Созданный пароль приложения использовать в SMTP_PASSWORD.
3. Настроить переменные окружения:

```
SMTP_HOST= smtp.mail.ru
SMTP_PORT=465
SMTP_USER=<your-email-username>
SMTP_PASSWORD=<your-app-password>
```

4 ОБНОВЛЕНИЕ И ПОДДЕРЖКА

4.1 Обновление Системы

Обновление Системы включает в себя:

1. Остановку текущих сервисов `docker-compose down;`
2. Получение новой версии кода `git pull origin main;`
3. Пересборку контейнеров `docker-compose build;`
4. Запуск обновленной версии системы `docker-compose up -d;`
5. Выполнение миграций базы данных (при наличии изменений) `docker exec backend alembic upgrade head.`

Важно! Перед обновлением рекомендуется создать резервную копию данных (пункт 4.2).

4.2 РЕЗЕРВНОЕ КОПИРОВАНИЕ

Резервное копирование включает возможность сохранения базы данных и пользовательских файлов с возможностью последующего восстановления.

4.2.1 База данных

Создание резервной копии PostgreSQL:

```
docker exec postgres_db pg_dump -U dbuser umarta_translator > backup_$(date +%Y%m%d).sql
```

Восстановление базы данных:

```
docker exec -i postgres_db psql -U dbuser umarta_translator < backup_20240101.sql
```

4.2.2 Файлы пользователей

Архивирование пользовательских файлов:

```
tar -czf files_backup_$(date +%Y%m%d).tar.gz backend/files/
```

Восстановление архива:

```
tar -xzf files_backup_20240101.tar.gz
```

5 МОНИТОРИНГ И ДИАГНОСТИКА

5.1 ПРОСМОТР ЛОГОВ СИСТЕМЫ

Все логи всех сервисов: `docker-compose logs -f`

Только ошибки (последние 100 строк): `docker-compose logs --tail=100 | grep -i error`

Логи конкретного сервиса (например, backend): `docker-compose logs -f backend`

5.2 ПРОВЕРКА СОСТОЯНИЯ СИСТЕМЫ

Для диагностики и мониторинга Системы используйте следующие команды:

- Проверка состояния API:

`curl http://localhost/api/health`

- Проверка статуса контейнеров:

`docker-compose ps`

- Мониторинг использования ресурсов:

`docker stats`

6 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ СИСТЕМЫ

Ниже приведены рекомендации для защиты Системы и пользовательских данных:

6.1 РЕКОМЕНДАЦИИ ДЛЯ ПРОДАКШЕНА

1. Смените пароль администратора сразу после первого входа в Систему.
2. Используйте надёжные, уникальные пароли для всех конфиденциальных параметров:
 - база данных (PostgreSQL);
 - JWT;
 - SMTP.
3. Настройте шифрование соединения (HTTPS) с помощью reverse proxy.
4. Ограничьте внешний доступ к порту 5432 (PostgreSQL).
5. Регулярно обновляйте Систему и зависимости.

6.2 НАСТРОЙКА HTTPS

Для защиты трафика между клиентом и сервером в продуктивной среде рекомендуется использовать

- Cloudflare для SSL и CDN;
- Let's Encrypt для бесплатных сертификатов;
- Nginx Proxy Manager для управления.

7 ПОЛЕЗНЫЕ КОМАНДЫ

Данный раздел содержит полезные команды для управления контейнерами и взаимодействия с системой, а также часто используемые SQL-запросы для быстрой диагностики состояния данных в базе (Таблица 6).

Таблица 6 – Полезные команды

КОМАНДА	НАЗНАЧЕНИЕ
<code>docker-compose restart</code>	Перезапуск всех сервисов
<code>docker exec -it backend bash</code>	Доступ в терминал контейнера
<code>docker exec -it postgres_db psql -U dbuser -d umarta_translator</code>	Подключение к PostgreSQL внутри контейнера
<code>SELECT COUNT(*) FROM users</code>	Подсчёт общего количества зарегистрированных пользователей
<code>SELECT status, COUNT(*) FROM files GROUP BY status</code>	Анализ распределения файлов по статусам
<code>SELECT COUNT(*) FROM user_subscriptions WHERE status = 'active'</code>	Подсчёт активных подписок пользователей

8 УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ

В таблице 7 перечислены неисправности, которые могут возникнуть при работе Системы, и рекомендации по их устранению.

Таблица 7 – Рекомендуемые действия по устранению неисправностей

ОШИБКА	РЕКОМЕНДУЕМЫЕ ДЕЙСТВИЯ
Контейнер не запускается	<ol style="list-style-type: none">Проверить логи: <code>docker-compose logs <service_name></code>Проверить переменные окружения: <code>docker-compose config</code>
База данных недоступна	<ol style="list-style-type: none">Проверить подключение: <code>docker exec backend ping db</code>Проверить PostgreSQL: <code>docker exec postgres_db pg_isready -U dbuser</code>
Ошибки при переводе	<ol style="list-style-type: none">Проверить API-ключи LLM-провайдеров.Убедится в доступности интернета.Проверить лимиты и квоты провайдеров.

8.1 ВСПОМОГАТЕЛЬНЫЕ РЕСУРСЫ

В случае возникновения неисправностей, не охваченных настоящей документацией, либо при возникновении вопросов по эксплуатации системы, рекомендуется воспользоваться следующими ресурсами:

- Документация: [documentation/](#);
- API документация: <http://localhost/api/docs>;
- Issues: https://git01.umarta.dev/products/УМАРТА.Переводчик-/tree/master?ref_type=heads.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ