

Общество с ограниченной ответственностью «ДАТАПАКС»
(ООО «ДАТАПАКС»)

УТВЕРЖДАЮ

Генеральный директор ООО «ДАТАПАКС»



А.А. Остренин

2026г.

Программное обеспечение
Автоматизированная система «Электронный путевой лист»

Руководство по установке и настройке

ЛИСТ УТВЕРЖДЕНИЯ

1197746651706.62.01.2.00005-01 90 01-ЛУ

Руководитель проекта

А.В. Москвина

2026

Инев. № подл.	Подпись и дата	Взам. инв. №	Инев. № дубл.	Подпись и дата

УТВЕРЖДЕН

1197746651706.62.01.2.00005-01 90 01-ЛУ

**Программное обеспечение
Автоматизированная система «Электронный путевой лист»**

Руководство по установке и настройке

1197746651706.62.01.2.00005-01 90 01

Листов 21

АННОТАЦИЯ

В данном документе приведено руководство по установке и настройке программного обеспечения Автоматизированная система «Электронный путевой лист» (далее по тексту – АС «ЭПЛ»).

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ О СОСТАВЕ И АРХИТЕКТУРЕ	4
2 ТРЕБОВАНИЯ К ПРОГРАММНО-АППАРАТНЫМ СРЕДСТВАМ	5
3 УСТАНОВКА ОБЩЕСИСТЕМНОГО СЕРВЕРНОГО ПО	6
4 УСТАНОВКА И НАСТРОЙКА СЕРВЕРНОЙ ЧАСТИ.....	7
4.1 Получение исходного кода.....	7
4.2 Настройка параметров	7
4.3 Сборка и запуск	8
4.4 Состав сервисов и сетевые порты.....	8
5 УСТАНОВКА И НАСТРОЙКА ВЕБ-ИНТЕРФЕЙСА.....	10
5.1 Получение исходного кода.....	10
5.2 Настройка параметров сборки	10
5.3 Сборка и запуск	10
6 УСТАНОВКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ	11
6.1 Подготовка окружения и получение кода.....	11
6.2 Настройка адреса сервера.....	11
6.3 Сборка дистрибутива	11
7 ПРОВЕРКА РАБОТОСПОСОБНОСТИ	12
8 ПРОЦЕССЫ, ОБЕСПЕЧИВАЮЩИЕ ПОДДЕРЖАНИЕ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММЫ.....	13
9 САМОСТОЯТЕЛЬНОЕ УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ....	14
10 УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ КОМПАНИЕЙ- ПРАВООБЛАДАТЕЛЕМ	15
11 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА	16
12 СОВЕРШЕНСТВОВАНИЕ ПРОГРАММЫ	17
13 ИНФОРМАЦИЯ О НЕОБХОДИМОМ ПЕРСОНАЛЕ ДЛЯ ОБЕСПЕЧЕНИЯ РАБОТОСПОСОБНОСТИ.....	18

1 ОБЩИЕ СВЕДЕНИЯ О СОСТАВЕ И АРХИТЕКТУРЕ

АС «ЭПЛ» построена по сервис-ориентированной архитектуре и состоит из набора серверных микросервисов, веб-интерфейса и мобильного приложения. Все серверные компоненты поставляются в виде контейнеров и разворачиваются на одном сервере под управлением Docker с использованием Docker Compose. Облачные платформы и оркестраторы (Kubernetes) для эксплуатации не требуются.

Исходный код и конфигурация развёртывания поставляются в составе дистрибутива в виде архивов трёх программных компонентов:

- серверная часть (backend, API и микросервисы);
- веб-интерфейс (frontend);
- мобильное приложение (mobile).

В состав серверной части входят следующие компоненты:

- API-сервис — основной программный интерфейс, бизнес-логика, аутентификация и авторизация;
- сервисы бизнес-процессов (BPMN) и их менеджер — исполнение и управление маршрутами обработки путевых листов;
- сервис учёта транспортных средств и сервис управления устройствами;
- сервис интеграции с внешней транспортной системой;
- модуль аналитики и отчётности на базе Apache Superset;
- СУБД PostgreSQL и хранилище Redis.

Веб-интерфейс представляет собой одностраничное приложение, собираемое в статические файлы и публикуемое веб-сервером nginx. Мобильное приложение является приложением-обёрткой, отображающим веб-интерфейс системы на мобильных устройствах под управлением ОС Android (версии 8 и выше).

2 ТРЕБОВАНИЯ К ПРОГРАММНО-АППАРАТНЫМ СРЕДСТВАМ

Для развёртывания серверной части АС «ЭПЛ» требуется выделенный сервер (физический или виртуальный) со следующими характеристиками (таблица 1).

Таблица 1 — Требования к серверу

Компонент	Минимальные требования
Процессор	не менее 4 виртуальных ядер (vCPU)
Оперативная память	не менее 8 ГБ (рекомендуется 16 ГБ)
Дисковое пространство	не менее 50 ГБ (SSD), с учётом роста БД
Операционная система	Linux x86-64 (Ubuntu 22.04 LTS / Debian 12 / РЕД ОС / Astra Linux или совместимая)
Общесистемное ПО	Docker Engine 24.0+ и плагин Docker Compose v2; клиент Git; доступ по SSH
Сеть	статический внешний IP-адрес или доменное имя; открытые порты согласно таблице 2

Установка СУБД PostgreSQL и хранилища Redis отдельно не требуется: они входят в состав поставки в виде контейнеров и разворачиваются автоматически совместно с прикладными сервисами.

Для сборки мобильного приложения дополнительно требуется рабочая станция с установленным Flutter SDK (раздел 6).

3 УСТАНОВКА ОБЩЕСИСТЕМНОГО СЕРВЕРНОГО ПО

Установка (развёртывание) инфраструктуры для работы АС «ЭПЛ» производится на выделенном сервере (см. раздел 2). Она включает в себя установку общесистемного программного обеспечения:

- 1) Установить операционную систему Linux и обеспечить доступ к серверу по протоколу SSH под учётной записью администратора.
- 2) Установить Docker Engine и плагин Docker Compose. Для дистрибутивов на базе Debian/Ubuntu установка выполняется командами:

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl git
# установка Docker Engine по официальной инструкции
docs.docker.com
sudo systemctl enable --now docker
```

- 3) Убедиться в работоспособности Docker и Docker Compose:

```
docker --version
docker compose version
```

СУБД PostgreSQL и хранилище Redis устанавливать отдельно не нужно — они поднимаются как контейнеры в составе compose-окружения серверной части (раздел 4). Параметры подключения к базе данных (имя БД, пользователь, пароль) задаются в конфигурации развёртывания.

4 УСТАНОВКА И НАСТРОЙКА СЕРВЕРНОЙ ЧАСТИ

Сборка и развёртывание серверных сервисов АС «ЭПЛ» выполняются средствами Docker Compose из исходного кода, входящего в состав дистрибутива. Развёртывание производится непосредственно на целевом сервере.

4.1 ПОЛУЧЕНИЕ ИСХОДНОГО КОДА

Скопировать архив исходного кода из состава дистрибутива на сервер и распаковать его:

```
ssh epladmin@<адрес_сервера>
unzip epl_backend_source.zip -d backend
cd backend
# скопируйте архив на сервер заранее: scp epl_backend_source.zip
epladmin@<сервер>:~
```

4.2 НАСТРОЙКА ПАРАМЕТРОВ

Перед первым запуском необходимо задать параметры развёртывания в файле compose-окружения (compose-dev.yaml для среды разработки или compose.yaml для среды эксплуатации). Настраиваются следующие группы параметров:

- 1) Параметры подключения к СУБД PostgreSQL — имя базы данных, пользователь и пароль (переменные окружения DB_HOST, DB_PORT, DB_USER, DB_PASSWORD, DB_NAME). Эти же значения задаются для контейнера СУБД (POSTGRES_DB, POSTGRES_USER, POSTGRES_PASSWORD). Значения паролей рекомендуется хранить вне репозитория — в переменных окружения или в файле .env, не включаемом в систему контроля версий.
- 2) Параметры интеграции с внешней транспортной системой для сервиса trn — адрес и учётные данные внешнего источника (переменные TRN_URL, TRN_USERNAME, TRN_PASSWORD, TRN_PARKS).
- 3) Параметры доступа сервиса vehicle к бизнес-процессам (BPMN_URL, BPMN_USERNAME, BPMN_PASSWORD, BPMN_PROCESS).

Пример блока параметров подключения к СУБД в файле compose-окружения:

```
environment:
  DB_HOST: db
  DB_PORT: 5432
  DB_USER: postgres
```

```
DB_PASSWORD: <пароль_БД>  
DB_NAME: get  
MIGRATING_ON: true
```

4.3 СБОРКА И ЗАПУСК

Запустить сборку образов и развёртывание сервисов командой Docker Compose:

```
docker compose -f compose-dev.yaml up -d --build
```

Команда последовательно собирает образы микросервисов, разворачивает контейнеры СУБД PostgreSQL и Redis, выполняет миграции базы данных (при MIGRATING_ON: true) и запускает прикладные сервисы. Признаком успешного развёртывания является переход всех контейнеров в состояние «running» (Up):

```
docker compose -f compose-dev.yaml ps
```

Для повторного развёртывания после обновления кода используется типовая последовательность, оформленная в виде сценария `deploy_dev.sh`:

```
scp epl_backend_source.zip epladmin@<адрес_сервера>:~  
ssh epladmin@<адрес_сервера> -C "unzip -o epl_backend_source.zip -d  
backend && \  
    docker compose -f compose-dev.yaml up -d --build"
```

4.4 СОСТАВ СЕРВИСОВ И СЕТЕВЫЕ ПОРТЫ

После развёртывания серверная часть предоставляет следующие сервисы и сетевые порты (таблица 2).

Таблица 2 — Сервисы и сетевые порты

Сервис (контейнер)	Порт	Назначение
get-frontend	8080	Веб-интерфейс (nginx, статическая сборка Vue). Внешняя точка входа пользователей.
get-api	8085	Основной API-сервис (REST), бизнес-логика, аутентификация, профили, уведомления.
get-bpmn-manager	8086	Менеджер бизнес-процессов (BPMN), управление и миграции процессов.
get-bpmn	8087	Движок исполнения бизнес-процессов (BPMN).
get-device-manager	8088	Сервис управления устройствами (терминалами/APM).
get-bi	8090	Модуль аналитики и отчётности (Apache Superset).

Сервис (контейнер)	Порт	Назначение
get-trn	8091	Сервис интеграции с внешней транспортной системой (справочники ТС, парки).
get-vehicle	8092	Сервис учёта транспортных средств и выпуска на линию.
get-db	15432	СУБД PostgreSQL 16 (порт публикуется только для администрирования).
get-redis	7099	Кэш и брокер сообщений Redis.

Для внешних пользователей на межсетевом экране сервера достаточно открыть порт 8080 (веб-интерфейс), а также порты 8085, 8086 и 8090, к которым обращается клиентское приложение (API, BPMN-менеджер и модуль аналитики). Порты СУБД и Redis наружу не публикуются.

5 УСТАНОВКА И НАСТРОЙКА ВЕБ-ИНТЕРФЕЙСА

Веб-интерфейс собирается из исходного кода, входящего в состав дистрибутива, и публикуется веб-сервером nginx в виде контейнера. Адреса серверных сервисов задаются на этапе сборки через параметры (build args) в файле compose-окружения.

5.1 ПОЛУЧЕНИЕ ИСХОДНОГО КОДА

```
ssh epladmin@<адрес_сервера>
unzip epl_frontend_source.zip -d frontend
cd frontend
# скопируйте архив на сервер заранее: scp epl_frontend_source.zip
epladmin@<сервер>:~
```

5.2 НАСТРОЙКА ПАРАМЕТРОВ СБОРКИ

В файле compose-окружения (compose-dev.yaml) в секции args указываются доменное имя или IP-адрес сервера для обращения веб-интерфейса к серверным сервисам:

```
args:
  - VITE_APP_TITLE=Электронный путевой лист
  - VITE_APP_VERSION=1.1.0
  - VITE_BASE_API=http://<адрес_сервера>:8085
  - VITE_BPMN_API=http://<адрес_сервера>:8086
  - VITE_APP_SUPERSET_URL=http://<адрес_сервера>:8090
  - VITE_DEFAULT_APP_TYPE=epl
```

5.3 СБОРКА И ЗАПУСК

Запустить сборку и развёртывание веб-интерфейса:

```
docker compose -f compose-dev.yaml up -d --build
```

При сборке выполняется компиляция приложения (Node.js, пакетный менеджер npm) и формирование статических файлов, которые копируются в образ nginx. После запуска веб-интерфейс доступен по адресу http://<адрес_сервера>:8080. Для повторного развёртывания используется сценарий deploy_dev.sh, аналогичный серверной части.

6 УСТАНОВКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Мобильное приложение АС «ЭПЛ» реализовано на технологии Flutter и представляет собой приложение-обёртку, отображающее веб-интерфейс системы на устройствах под управлением ОС Android (версии 8 и выше). Сборка приложения выполняется на рабочей станции разработчика с установленным Flutter SDK.

6.1 ПОДГОТОВКА ОКРУЖЕНИЯ И ПОЛУЧЕНИЕ КОДА

```
unzip epl_mobile_source.zip -d mobile  
cd mobile  
flutter pub get
```

6.2 НАСТРОЙКА АДРЕСА СЕРВЕРА

В файле lib/main.dart указывается адрес веб-интерфейса (раздел 5), который будет открывать приложение. Для среды эксплуатации значение url заменяется на боевой адрес сервера:

```
body: const CustomInAppBrowser(  
  url: "http://<адрес_сервера>:8080",  
),
```

6.3 СБОРКА ДИСТРИБУТИВА

Сборка установочного пакета для платформы Android выполняется штатными средствами Flutter:

- flutter build apk --release — формируется файл APK для установки на устройства;
- flutter build appbundle --release — формируется пакет App Bundle для публикации в магазине приложений.

Готовый дистрибутив устанавливается на мобильные устройства пользователей или публикуется в магазине приложений в соответствии с принятым в организации порядком распространения/по согласованию с заказчиком.

7 ПРОВЕРКА РАБОТОСПОСОБНОСТИ

После развёртывания необходимо убедиться в работоспособности системы:

- 1) Проверить, что все контейнеры серверной части находятся в состоянии «running» (Up) — командой `docker compose ps`.
- 2) Открыть в веб-браузере адрес `http://<адрес_сервера>:8080` и убедиться, что загружается страница входа в АС «ЭПЛ».
- 3) Выполнить вход под учётной записью администратора и проверить доступность основных разделов системы.
- 4) При необходимости проверить доступность API и модуля аналитики по адресам `http://<адрес_сервера>:8085` и `http://<адрес_сервера>:8090` соответственно.

В случае выявления ошибок следует проанализировать журналы соответствующих контейнеров командой `docker compose logs <имя_сервиса>` и устранить причину (см. разделы 9–11).

8 ПРОЦЕССЫ, ОБЕСПЕЧИВАЮЩИЕ ПОДДЕРЖАНИЕ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММЫ

Поддержание жизненного цикла АС «ЭПЛ» осуществляется за счёт её сопровождения Компанией-правообладателем, включающего в себя следующие процессы:

- техническая поддержка пользователей;
- исправление ошибок и недостатков, выявленных в процессе эксплуатации;
- проведение переработки (модификации) программного обеспечения и его совершенствование;
- поддержание целостности АС «ЭПЛ» и её развитие.

9 САМОСТОЯТЕЛЬНОЕ УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ

Самостоятельное устранение неисправностей пользователем невозможно. Компания-правообладатель обязуется обеспечить консультирование пользователей в течение всего периода эксплуатации АС «ЭПЛ». Для этого Компания-правообладатель обеспечивает доступность консультаций в рамках Технической поддержки АС «ЭПЛ».

10 УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ КОМПАНИЕЙ- ПРАВООБЛАДАТЕЛЕМ

В рамках устранения неисправностей осуществляется следующее:

- выявление ошибок в функционировании;
- исправление ошибок, выявленных в функционировании;
- тестирование после исправления ошибок;
- обновление для исправления выявленных ошибок.

11 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Техническая поддержка пользователей осуществляется по следующим каналам связи:

- телефон;
- электронная почта.

В рамках технической поддержки оказываются следующие услуги:

- помощь в установке;
- помощь в настройке и администрировании;
- помощь в установке обновлений;
- пояснение функциональности, помощь в эксплуатации;
- предоставление актуальной документации по установке/настройке/функциям;
- устранение неисправностей, связанных с работой АС «ЭПЛ».

Техническая поддержка разделяется на следующие линии (уровни):

- техническая поддержка первого уровня подразумевает регистрацию обращения и консультацию, оказываемую пользователям;
- техническая поддержка второго уровня подразумевает устранение возникших неисправностей, связанных с работой АС «ЭПЛ», осуществляемое техническими специалистами Компании-правообладателя, в режиме 8х5 (восемь часов в день, пять рабочих дней в неделю).

12 СОВЕРШЕНСТВОВАНИЕ ПРОГРАММЫ

АС «ЭПЛ» регулярно и планомерно развивается: в ней появляются новые функции и возможности, проводится оптимизация работы и обновляются интерфейсы.

Компания-правообладатель осуществляет переработку (модификацию) самостоятельно или силами привлечённых третьих лиц без изменения условий об исключительном праве правообладателя.

Компания-правообладатель может осуществлять разработку и реализацию дополнительных сервисов и модулей к АС «ЭПЛ» по заявкам и заказам заказчиков — юридических лиц (далее — Заказчик). Для этого Заказчику необходимо написать электронное письмо на адрес info@dtprax.ru. Компания-правообладатель согласует с Заказчиком условия разработки дополнительных функций АС «ЭПЛ» по заявкам и заказам Заказчиков. Данные работы осуществляются на возмездной основе.

Переработка (модификация) АС «ЭПЛ» может проводиться и по инициативе самой Компании-правообладателя в связи с изменением законодательства, административных регламентов и т.п.

В результате переработки (модификации) АС «ЭПЛ» выпускаются обновлённые версии (релизы) АС «ЭПЛ», которые предоставляются пользователю, предлагаются заказчикам.

13 ИНФОРМАЦИЯ О НЕОБХОДИМОМ ПЕРСОНАЛЕ ДЛЯ ОБЕСПЕЧЕНИЯ РАБОТОСПОСОБНОСТИ

Обслуживание АС «ЭПЛ» осуществляется специалистами Компании-правообладателя. Обслуживающий персонал разделяется на следующие категории:

- системный администратор;
- администратор БД;
- специалист по техническому обслуживанию;
- специалист по технической поддержке пользователей АС «ЭПЛ».

Рекомендуемая численность персонала для эксплуатации АС «ЭПЛ»:

- системный администратор — 1 и более штатные единицы;
- администратор БД — 1 и более штатные единицы;
- специалист по техническому обслуживанию — 1 и более штатные единицы;
- специалист по технической поддержке пользователей АС «ЭПЛ» — 2 и более штатные единицы.

Основными обязанностями системного администратора являются установка, настройка и мониторинг работоспособности АС «ЭПЛ». Системный администратор должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных средств, необходимых для функционирования АС «ЭПЛ», а также должен иметь профессиональные знания и практический опыт в области системного администрирования и работы с контейнеризацией (Docker).

Основными обязанностями администратора БД являются установка, модернизация, настройка параметров СУБД; оптимизация функционирования прикладных баз данных по времени отклика и скорости доступа к данным; резервное копирование и аварийное восстановление данных; конфигурирование и настройка АС «ЭПЛ»; управление и реализация эффективной политики доступа к информации, хранящейся в прикладных базах данных. Администратор баз данных должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию используемых СУБД.

Основными обязанностями специалиста по техническому обслуживанию являются модернизация, настройка и мониторинг работоспособности

технических средств, на которых функционирует АС «ЭПЛ»; конфигурирование и настройка программно-технических средств; диагностика типовых неисправностей; настройка локальной компьютерной сети и доступа в сеть Интернет; контроль доступа к сетевым ресурсам; настройка сетевого окружения.

Основными обязанностями специалиста по технической поддержке АС «ЭПЛ» являются работа со службой Helpdesk; диагностика и устранение проблем и ошибок, возникших у пользователей при эксплуатации АС «ЭПЛ»; постановка задач системному администратору и разработчикам АС «ЭПЛ» по устранению возникших неисправностей и ошибок.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]