

ООО «ВОЙСЛИНК»

**МОДУЛЬ ВЫДАЧИ ТРАНСПОРТНЫХ РАЗРЕШЕНИЙ**

**PERMISSIONMANAGER**

**РУКОВОДСТВО АДМИНИСТРАТОРА ПО УСТАНОВКЕ И НАСТРОЙКЕ**

**Аннотация**

Настоящее руководство содержит описание практического использования Модуля выдачи транспортных разрешений PermissionManager (далее - Система). Также перечислены возможные сбои ПО и действия по устранению возникших аварийных ситуаций. В документе описана последовательность действий Администратора при работе с Системой и приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия Администратора.

## Содержание

<b>1 Назначение и функции системы .....</b>	<b>4</b>
1.1 Назначение системы .....	4
1.2 Функции системы.....	4
<b>2 Условия выполнения программы .....</b>	<b>5</b>
2.1.1 Программное обеспечение.....	5
2.1.2 Аппаратное обеспечение .....	5
<b>3 Установка и удаление ПО .....</b>	<b>7</b>
3.1 Порядок установки ПО.....	7
3.2 Установка серверного ПО .....	7
3.3 Установка клиентского ПО .....	7
<b>4 Настройка системы.....</b>	<b>8</b>
4.1 Запуск системы.....	8
4.2 Порядок проверки работоспособности .....	9
4.3 Администрирование .....	9
<b>5 Сообщения администратору .....</b>	<b>11</b>
<b>Приложение А. Инструкция по установке и настройке прикладного ПО .....</b>	<b>12</b>
<b>Перечень принятых сокращений.....</b>	<b>15</b>

## **1 Назначение и функции системы**

### **1.1 Назначение системы**

Областью применения Системы является обеспечение учета транспортных разрешений и контроль их соблюдения. Используется службой учета транспортных разрешений.

### **1.2 Функции системы**

Система предназначена для автоматизации следующих видов деятельности:

- ведение реестра транспортных средств;
- выдача и автоматизированный контроль над соблюдением транспортных разрешений;
- формирование аналитических отчетов;
- оповещение пользователя о нарушениях транспортных разрешений.

## **2 Условия выполнения программы**

### **2.1.1 Программное обеспечение**

В качестве базовых технологий и платформ Системы используются технологии и платформы Linux.

#### **2.1.1.1 Системное и базовое программное обеспечение**

Программное обеспечение Системы функционирует на базе системного и базового ПО. На сервере приложений Системы должно быть установлено следующее системное и базовое программное обеспечение:

- Сервер 1 (ядро системы): ОС Linux Ubuntu 16.04, SSH-сервер.
- Сервер 2 (Веб-сервер): ОС Linux Ubuntu 16.04, веб-сервер nginx, SSH-сервер.

На сервере баз данных Системы установлено следующее системное и базовое программное обеспечение:

- ОС Linux Ubuntu 16.04;
- СУБД PostgreSQL; - SSH-сервер.

#### **2.1.1.2 Рабочие места операторов и администраторов Системы**

На автоматизированных рабочих местах операторов и администраторов Системы установлено следующее системное и базовое программное обеспечение: - операционная система Windows; - один из следующих браузеров:

- Google Chrome версии 45 и выше;
- Яндекс Браузер версии 16 и выше.

#### **2.1.1.3 Прикладное программное обеспечение**

Прикладное программное обеспечение на серверах Системы включает:

- Сервер 1 — Ядро Системы.
- Сервер 2 — Веб сервер Системы.

Порядок установки и настройки прикладного ПО описан в приложении А «Инструкция по установке и настройке прикладного ПО».

### **2.1.2 Аппаратное обеспечение**

Аппаратное обеспечение Системы состоит из серверной инфраструктуры, включающей в себя сервера приложений, брокеров сообщений и баз данных, а также парк клиентских рабочих станций. Все программное обеспечение, развернутое в серверной инфраструктуре, поддерживает работу в среде виртуализации. Для построения сети передачи данных

используются маршрутизаторы, коммутаторы, точки беспроводного доступа, беспроводные маршрутизаторы, модемы и сетевые адаптеры.

### **2.1.2.1 Сервер приложений**

Серверы приложений Системы имеют следующие характеристики:

- Сервер 1:

–Процессор — от 4 ядер, от 2.1 ГГц;

–ОЗУ — от 8 Гб;

–ПЗУ — от 240 Гб, рекомендуется использование SSD-накопителей и резервирования;

–Сеть — от 1 Гбит/с.

- Сервер 2:

–Процессор — от 2 ядер, от 2.1 ГГц;

–ОЗУ — от 4 Гб;

–ПЗУ — от 120 Гб, рекомендуется использование SSD-накопителей и резервирования;

–Сеть — от 1 Гбит/с.

### **2.1.2.2 Сервер баз данных**

Сервер баз данных Системы имеет следующие характеристики:

–Процессор — от 8 ядер, от 2.5 ГГц;

–ОЗУ — от 16 Гб;

–ПЗУ — от 500 Гб, рекомендуется использование SSD-накопителей и резервирования;

–Сеть — от 1 Гбит/с, рекомендуется 10 Гбит/с.

### **2.1.2.3 АРМ пользователя**

Для запуска и нормального функционирования клиентского программного обеспечения Системы необходим персональный компьютер со следующими минимальными требованиями:

–компьютер с ОС не ниже Windows 7 (предпочтительно Windows 10) и не менее 2 Гб оперативной памяти или ОС Linux - коробочные или свободно распространяемые версии

–дистрибутивов Linux, рекомендуется Ubuntu (не ниже версии 16.04).

–монитор с диагональю не менее 24 дюймов (рекомендуется 2 монитора с диагональю 27 дюймов и выше);

–мышь;

–клавиатура.

### **3 Установка и удаление ПО**

#### **3.1 Порядок установки ПО**

Порядок установки программного обеспечения Системы описан в разделе 3.2 «Установка серверного ПО».

#### **3.2 Установка серверного ПО**

Установка общесистемного серверного ПО должна осуществляться в соответствии с требованиями и рекомендациями производителя общесистемного ПО, опубликованными на общедоступном информационном ресурсе «<https://ubuntu.com/download/server>».

Установка серверного ПО системы управления базами данных (далее – СУБД) должна осуществляться в соответствии с требованиями и рекомендациями производителя СУБД, опубликованными на общедоступном информационном ресурсе «<https://www.postgresql.org/>».

Установка серверного ПО системы управления базами данных (далее - СУБД) должна осуществляться в соответствии с требованиями и рекомендациями производителя СУБД, опубликованными на майте производителя «<https://www.postgresql.org/>».

Инструкция по установке и настройке прикладного ПО Системы описана в приложении А.

#### **3.3 Установка клиентского ПО**

Не требуется.

## 4 Настройка системы

### 4.1 Запуск системы

Запуск программного обеспечения на АРМ:

1. Запустить браузер на АРМ.
2. В адресной строке браузера ввести `http://<IP-адрес установки>:8100`, нажать выполнить (Enter).
3. Появится всплывающее окно для аутентификации пользователя (рисунок 1).
4. Ввести «Имя» пользователя, «пароль» и подтвердить нажатием «Войти».
5. Запустится клиентское программное обеспечение (рисунок 2).

Первый вход в систему производится под именем и паролем пользователя, заданным при установке Системы (поле «Введите e-mail» – `root@root.ru`, поле «Введите пароль» – `root`).

Рекомендации по созданию пользователей описаны в разделе 6 данного документа.

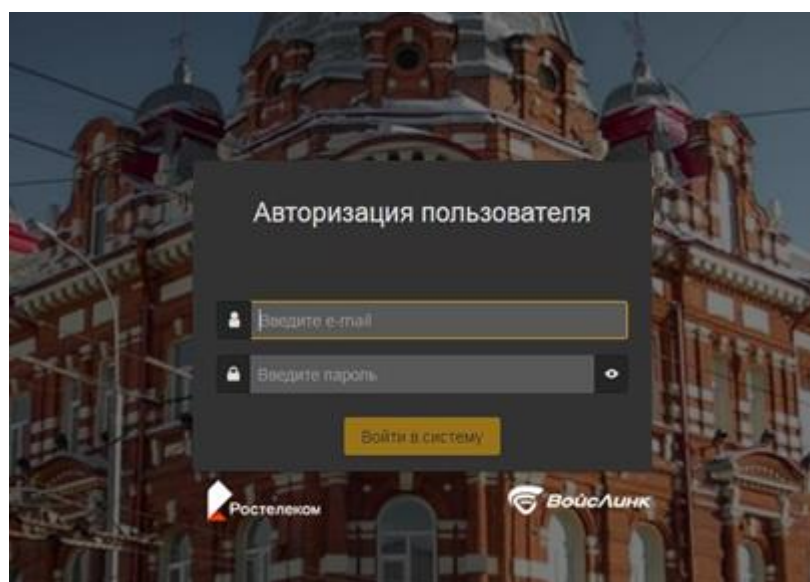


Рисунок 1 – Окно аутентификации пользователя

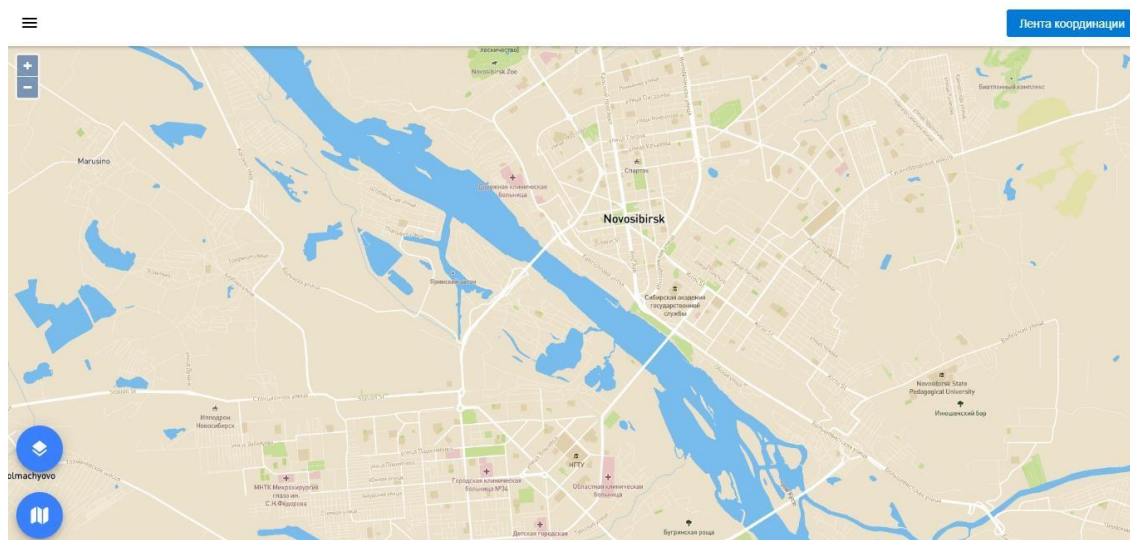



Рисунок 2 – Главное окно системы



## 4.2 Порядок проверки работоспособности

Программное обеспечение работоспособно, если в результате действий пользователя, изложенных в п. 4.1, на экране монитора отображается главное окно клиентского приложения без выдачи пользователю сообщений о сбое в работе.

## 4.3 Администрирование

Чтобы выполнить совершить действия по управлению системой нужно открыть меню конфигурирования и управления элементами системы (рисунок 2), т. е. нажать на иконку  («Меню») в левом верхнем углу основного окна клиентского приложения, и выбрать вкладку «Администрирование».

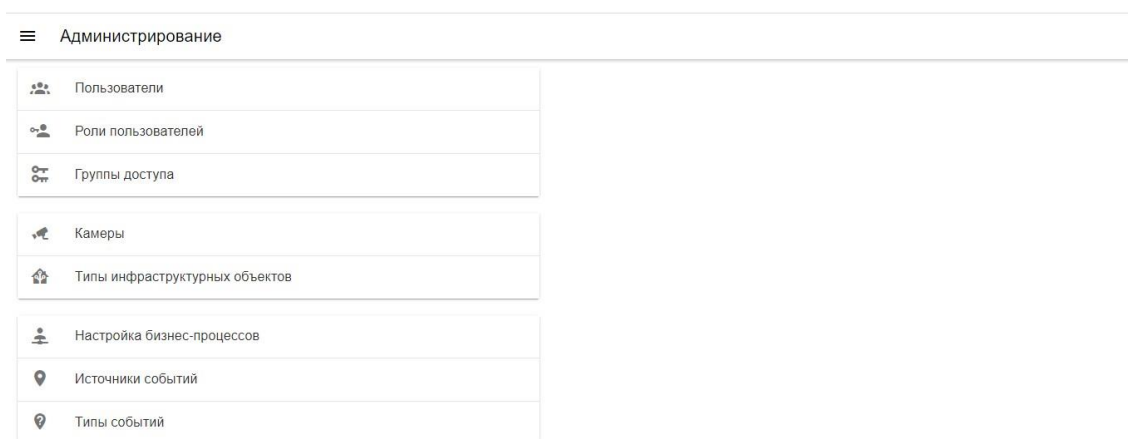


Рисунок 3 – Меню «Администрирование»

Меню «Администрирование» включает в себя следующие разделы (рисунок 3):

1. Пользователи;
2. Роли пользователей;
3. Группы доступа;
4. Камеры;
5. Типы инфраструктурных объектов;
6. Источники событий;
7. Типы событий;
8. Типы прочих средств информирования;
9. Прочие средства информирования;
10. ЕПУТС;
11. Транспорт;
12. Транспортные средства;
13. Оповещения;
14. Оповещения внешних пользователей;
15. Настройка бизнес-процессов;

16. Отчеты;
17. Правила мониторинга;
18. Теги моделей.

## 5 Сообщения администратору

Код сообщения	Содержание
Общие сообщения	
OC1	"Во время сохранения произошла ошибка, повторите сохранение. Если ошибка повторяется несколько раз - проверьте соединение с сервером."
OC2	"Сервер недоступен, проверьте соединение"
OC3	"Сохранено"
OC4	"Информация сохранена"
OC5	"Подтвердите действие"
Администрирование системы	
Администрирование пользователей и ролей	
АПР1	"Пользователь успешно сохранен"
АПР2	"Введено неверное имя пользователя"
АПР3	"Роль успешно сохранена"
АПР4	"Введено неверное имя роли"
АПР5	"Роль успешно удалена"
АПР6	"Группа доступа успешно сохранена"
АПР7	"Введено неверное имя группы"
АПР8	"Группа доступа успешно удалена"
Администрирование типов объектов	
АТО1	"Удаление успешно"
АТО2	"Группа доступа успешно сохранена"
АТО3	"Введено неверное название типа"
Администрирование источников событий	
АИС1	"Источник удален"
АИС2	"Введено неверное имя источника событий"
АИС3	"Источник событий успешно сохранен"
АИС4	"Тип событий успешно сохранен"
АИС5	"Введено неверное имя типа событий"
АИС6	"Тип событий успешно удален"

Приложение А. Инструкция по установке и настройке прикладного ПО  
**Инструкция по установке и настройке прикладного ПО Системы**

### 1. Установка глобальных зависимостей

```
$ ./glob_source/intall.sh
```

**Содержит:**

- [locale configuration](#)
- [gcc](#)
- [make](#)
- [makeself](#)
- [libssl-dev](#)
- [htop](#)
- [iotop](#)
- [postgresql-10](#)
- [pgtap](#)
- [nginx](#)

### 2. Установка окружения

```
$ ./env_source/intall.sh
```

Устанавливает environment в `${PROJECT_ROOT}/env`. Если переменная окружения `PROJECT_ROOT` не задана, то установит в корень репозитория

Внимание! Если при установке под virtualbox не удастся создать символическую ссылку, используйте:

```
$ VBoxManage setextradata VM_NAME  
VBoxInternal2/SharedFoldersEnableSymlinksCreate/SHARE_NAME 1
```

Где `VM_NAME` - имя виртуальной машины и `SHARE_NAME` имя расшаренной директории (без префикса "sf\_").

**Содержит:**

- Python-3.6.4 source code
- Python3 modules: [requirements.txt](#)
- Node-8.9.4 prebuild for linux-x64
- Node server modules: [package.json](#)
- Node client modules: [package.json](#)

### 3. Активация окружения

```
$ . _activate_env.sh  
(asudd)$
```

Добавляет в `${PATH}` python3, inv, node, npm, ava

Добавляет в `${PYTHONPATH}` `${PROJECT_ROOT}`

Добавляет в `${NODE_PATH}`

```
${PROJECT_ROOT}/app/client/node_modules:${PROJECT_ROOT}/app/server/node_modules
```

Внимание! Все команды при работе с проектом должны запускаться из активированного окружения.

(asudd)\$ - так выглядит активированное окружение

#### 4. Файл конфигурации

```
$ cp configs/default/config.yaml config.yaml
```

Модифицировать конфигурацию в config.yaml

#### 5. Invoke задачи

Для запуска задачи используется invoke (python task-runner)

Документация: <http://docs.pyinvoke.org/en/latest/>

```
(asudd)$ inv --list
```

Available tasks:

```
clear                Clear autogenerated directories
demo.up              Pull from master and rebuild project on asudd-stand
env.node-modules     Run npm install, archive node_modules to env_source, run
env_source/nodejs_modules/install.sh
env.python3-modules Download pip3 env_source/python3_modules/requirements.txt,
install python3 modules
planner.setup       asudd-planner.service to systemd
planner.start       Start asudd planner
app.client.build    Build client with webpack. With --watch start webpack
watcher
app.db.compile      Compile app/db schema
app.db.create       Create app/db database. With --reset recreate database
app.db.enumsSchema Load enums from app/db and save to docs/app/enums.md
app.db.init         run app.db.compile, then execute build
app.db.tests        Run pgtap tests, default --schema="*", --pattern="^test_"
app.server.setup    asudd-app-server.service to systemd
app.server.start    Start node js server
app.server.tests    Run unittests, --pattern - expression for matching tests
files from app/server/tests, default --pattern="*/test*.js"
app.site.docs       Generate documentation from app/schema to
app/site/docs/api
app.site.setup      asudd-app-site.conf to nginx
```

#### 6. Установка БД, переустановка БД

```
(asudd)$ inv app.db.create app.db.init
```

```
(asudd)$ inv app.db.create --reset app.db.init
```

#### 7. Запустить сервер

```
(asudd)$ inv app.server.start
```

#### 8. Добавить asudd-app-site в nginx

```
$ cp configs/default/asudd-app-site.conf asudd-app-site.conf
```

Модифицировать конфигурацию в asudd-app-site.conf

```
(asudd)$ inv app.site.setup
```

#### 9. Запустить сборку клиента

```
(asudd)$ inv app.client.build [--watch]
```

**10. Запустить тесты БД**

```
(asudd)$ inv app.db.tests
```

**11. Запустить тесты сервера**

```
(asudd)$ inv app.server.tests
```

**12. Запуск сервера через systemd**

```
$ cp configs/default/asudd-app-server.service asudd-app-server.service
```

Модифицировать конфигурацию в `asudd-app-server.service`

```
(asudd)$ inv app.server.setup
```

**Перечень принятых сокращений**

АСУДД	Автоматизированная система управления дорожным движением
АРМ	Автоматизированное рабочее место
ВС	Внештатная ситуация
ДТ	Детектор транспорта
ДТП	Дорожно-транспортное происшествие
АСУДД	Интеллектуальная транспортная система
КСОДД	Комплексная схема организации дорожного движения
ЛВС	Локальная вычислительная сеть
ОС	Операционная система
ПДД	Правила дорожного движения
ПК	Персональный компьютер
ПО	Программное обеспечение
СО	Светофорный объект
ТП	Транспортный поток
ТС	Транспортное средство
ТСОДД	Технические средства организации дорожного движения
УДС	Уличная дорожная сеть
ЧС	Чрезвычайная ситуация