**Система технического обслуживания и ремонта ТОИРУС**

**Программное обеспечение**

**Сервер системы**

**и рабочее место оператора**

**Руководство пользователя и администратора**

**ООО ИПК Технологии Энергосбережения**

**2017**

**Содержание**

[Аннотация 6](#_Toc485643921)

[Глоссарий 6](#_Toc485643922)

[1. Общие сведения о программном комплексе 7](#_Toc485643923)

[1.1 Сведения о назначении 7](#_Toc485643924)

[1.2 Основные функции системы ТОИРУС 8](#_Toc485643925)

[1.3 Структура нарядов 9](#_Toc485643926)

[1.4 Минимальный состав технических средств 10](#_Toc485643927)

[1.5 Требования к персоналу. 11](#_Toc485643928)

[2 Структура и описание программного обеспечения 12](#_Toc485643929)

[2.1 Структура данных системы. 12](#_Toc485643930)

[2.2 Сведения о структуре программного обеспечения 14](#_Toc485643931)

[2.3 Подсчет времени обслуживания 14](#_Toc485643932)

[3 Установка программного обеспечения 16](#_Toc485643933)

[3.1 Установка зависимых пакетов 16](#_Toc485643934)

[4 Запуск и настройка программного обеспечения 21](#_Toc485643935)

[5 Описание API 22](#_Toc485643936)

[5.1 Типы запросов 23](#_Toc485643937)

[5.2 Алгоритм авторизации 25](#_Toc485643938)

[6 Описание интерфейса оператора 27](#_Toc485643939)

[6.1 Главный экран интерфейса. 27](#_Toc485643940)

[6.2 Шаблоны 29](#_Toc485643941)

[7 Работа с базой данных 31](#_Toc485643942)

[7.1 Таблица типов оповещений [alerttype] 31](#_Toc485643943)

[7.2 Таблица бригады [brigade] 31](#_Toc485643944)

[7.3 Таблица пользователей бригады [brigadeuser] 32](#_Toc485643945)

[7.4 Таблица контрагентов [contragent] 32](#_Toc485643946)

[7.5 Таблица пользователей контрагентов [contragentuser] 32](#_Toc485643947)

[7.6 Таблица типов критичности [criticaltype] 33](#_Toc485643948)

[7.7 Таблица дефектов [defect] 33](#_Toc485643949)

[7.8 Таблица типов дефектов [defecttype] 34](#_Toc485643950)

[7.9 Таблица документации [documentation] 34](#_Toc485643951)

[7.10 Таблица типов документации [documentationtype] 34](#_Toc485643952)

[7.11 Таблица оборудования [equipment] 35](#_Toc485643953)

[7.12 Таблица моделей оборудований [equipmentmodel] 35](#_Toc485643954)

[7.13 Таблица статусов оборудования [equipmentstatus] 36](#_Toc485643955)

[7.14 Таблица типов оборудования [equipmenttype] 36](#_Toc485643956)

[7.15 Таблица геоположения [gpstrack] 36](#_Toc485643957)

[7.16 Таблица журнала [journal] 36](#_Toc485643958)

[7.17 Таблица измеренных значений [measuredvalue] 37](#_Toc485643959)

[7.18 Таблица типов измерений [measuretype] 37](#_Toc485643960)

[7.19 Таблица типов объектов [objects] 37](#_Toc485643961)

[7.20 Таблица типа объекта [objecttype] 38](#_Toc485643962)

[7.21 Таблица операций [operation] 38](#_Toc485643963)

[7.22 Таблица для хранения файлов после операций [operationfile] 39](#_Toc485643964)

[7.23 Таблица статуса операций [operationstatus] 39](#_Toc485643965)

[7.24 Таблица шаблона операций [operationtemplate] 39](#_Toc485643966)

[7.25 Таблица инструментов операций [operationtool] 40](#_Toc485643967)

[7.26 Таблица типов операций [operationtype] 40](#_Toc485643968)

[7.27 Таблица вердиктов операций [operationverdict] 40](#_Toc485643969)

[7.28 Таблица уровня наряда [orderlevel] 41](#_Toc485643970)

[7.29 Таблица нарядов [orders] 41](#_Toc485643971)

[7.30 Таблица статусов наряда [orderstatus] 42](#_Toc485643972)

[7.31 Таблица вердиктов наряда [orderverdict] 42](#_Toc485643973)

[7.32 Таблица обновлений параметров [referenceupdate] 42](#_Toc485643974)

[7.33 Таблица частей ремонта [repairpart] 42](#_Toc485643975)

[7.34 Таблица типов частей ремонта [repairparttype] 43](#_Toc485643976)

[7.35 Таблица этапов частей ремонта [stagerepairpart] 43](#_Toc485643977)

[7.36 Таблица задач [task] 43](#_Toc485643978)

[7.37 Таблица задач частей ремонта [taskrepairpart] 44](#_Toc485643979)

[7.38 Таблица этапов задач [taskstage] 44](#_Toc485643980)

[7.39 Таблица списка этапов задач [taskstagelist] 45](#_Toc485643981)

[7.40 Таблица списка операций этапа задач [taskstageoperationlist] 45](#_Toc485643982)

[7.41 Таблица статуса этапов задач [taskstagestatus] 46](#_Toc485643983)

[7.42 Таблица вердиктов этапов задач [taskstageverdict] 46](#_Toc485643984)

[7.43 Таблица шаблонов этапов задач [taskstagetemplate] 46](#_Toc485643985)

[7.44 Таблица типов этапов задач [taskstagetype] 47](#_Toc485643986)

[7.45 Таблица статус задач [taskstatus] 47](#_Toc485643987)

[7.46 Таблица шаблонов задач [tasktemplate] 47](#_Toc485643988)

[7.47 Таблица частей ремонта шаблонов задач [tasktemplaterepairpart] 48](#_Toc485643989)

[7.48 Таблица типа задач [tasktype] 48](#_Toc485643990)

[7.49 Таблица вердикта задач [taskverdict] 48](#_Toc485643991)

[7.50 Таблица токенов [token] 49](#_Toc485643992)

[7.51 Таблица инструментов [tool] 49](#_Toc485643993)

[7.52 Таблица типов инструментов [tooltype] 49](#_Toc485643994)

[7.53 Таблица пользователей [users] 50](#_Toc485643995)

[8 Обслуживание БД 51](#_Toc485643996)

[9 Руководство программиста 52](#_Toc485643997)

[9.1 Структура проекта 52](#_Toc485643998)

[9.2 Перечень и описание файлов сервера 52](#_Toc485643999)

[10 Возможные ошибки и их устранение 54](#_Toc485644000)

[10.1 Происходит ошибка подключения к серверу 54](#_Toc485644001)

## **Аннотация**

**В данном документе приведена информация для пользователя, администратора и системного программиста по установке, настройке и использованию программного обеспечения сервера и интерфейса конфигурирования системы технического обслуживания и ремонта, входящих в состав программного комплекса ТОИРУС.**

**В разделе «Общие сведения о программном комплексе» указаны назначение и функции программного комплекса и сведения о технических и программных средствах, обеспечивающих выполнение программных компонентов комплекса.**

**В разделе «Структура программного комплекса» приведены сведения о структуре комплекса, его составных частях.**

**В разделе «Настройка программного комплекса» приведено описание действий по настройке компонентов программного обеспечения на условия конкретного применения.**

**В разделе «Руководство программиста» приведено краткое руководство по файловой структуре приложений.**

## **Глоссарий**

**Система – весь комплекс системы технического обслуживания и ремонта ТОИРУС.**

**ПТК – программно-технический комплекс.**

**Сервер – ПТК, выполняющий функции получения/отдачи, генерации, хранения всех данных системы.**

**АРМ оператора – компьютер или другое устройство, имеющее доступ к серверу через веб-интерфейс (АРМ – автоматизированное рабочее место).**

**Веб-сервер Системы – программное обеспечение, выполняющее роль публикатора всех данных Системы.**

**МК ТОиР – мобильный клиент системы.**

# 

# **Общие сведения о программном комплексе**

**Программный комплекс верхнего уровня состоит из нескольких компонентов: сервера системы (с API внешнего доступа), базы данных, веб-сервера, клиентских АРМ и различных сервисов (аналитики, планирования, обмена с другими системами и т.д.).**

**Система ТОИРУС состоит из трех уровней: нижнего, среднего и верхнего.**

**Полевой (нижний) уровень состоит из оборудования, агрегатов, приборов учета и контроля и меток, которыми оно идентифицируется (RFID, QR, Barcode).**

**Средний уровень — это мобильный комплекс ТОИР, который может состоять как из отдельного клиентского устройства, так и нескольких устройств, разделенных по функционалу (измерение параметров (измерители с различными интерфейсами сопряжения), чтение меток (сканеры, Bluetooth и RFID считыватели)), отдельные планшеты.**

**Верхний уровень выполняет функции обмена данными со средним уровнем системы, смежными и вышестоящими системами по информационным сетям, архивирования данных, интерфейса с оператором, генерацию нарядов. К верхнему уровню системы относятся автоматизированные рабочие места диспетчеров, включая как аппаратную часть, так и программное обеспечение (общее и прикладное), коммуникационное оборудование.**

## **Сведения о назначении**

**Программное обеспечение представляет собой комплекс программно-технических средств, представляющий собой сервер и набор рабочих станций, которые позволяют операторам, пользователям и администраторам из числа обслуживающего персонала осуществлять необходимые функции управления осмотрами и сервисом оборудования системы.**

**Предоставление единой точки веб-доступа персоналу и пользователям системы ко всей необходимой информации с разграничением уровней доступа к информации.**

**Сервер системы может работать на любом оборудовании, отвечающем требованиям, заложенным в пункте 1.3 этого документа. Предпочитаемая операционная система Linux (Ubuntu, Debian).**

**Функционирование сервера и его компонентов построено на работе с предустановленной на этапе настройки базой данных.**

**Основными функциями сервера, данной системы, является хранения полученной информации, ее структуризация и анализ полученных данных. На основе этих трех признаков, формируется основная логика системы.**

**Точками входа для взаимодействия с системой, служит проработанный API и веб-интерфейс.**

**В задачи сервера входит автоматическое формирование задач-нарядов, получение его результатов, включая медиа данные.**

## **Основные функции системы ТОИРУС**

**Сервер выполняет следующие функции:**

**- генерация нарядов и связанных с ними задач для обслуживающего персонала, согласно их графику работы;**

**- передача нарядов на мобильные клиенты;**

**- передача документов, связанных с обследуемым оборудованием (руководств по эксплуатации, инструкций, паспортов и т.д.) на МК ТОиР;**

**- аутентификацию пользователя по его токену для доступа к данным и обновлениям системного программного обеспечения;**

**- прием и обработка данных по состоянию объекта/оборудования и проведенным работам;**

**- получение и отображение на АРМ оператора трека передвижения оператора;**

**- получение и отображение на АРМ оператора журнала работы оператора;**

**- прием и сохранение на сервере фотографий и другого медиаконтента;**

**- анализ принятой информации (расчеты эффективности работы персонала, выявление недоработок, статистический анализ дефектов и неисправностей);**

**- периодическое резервирование данных и очистка архивов;**

**- обновления программного обеспечения клиентских устройств.**

**Веб-сервер выполняет следующие функции:**

**- публикация данных;**

**- предоставление API.**

**АРМ оператора выполняет следующие функции:**

**- предоставление пользователям системы интерфейса для заполнения всех необходимых для работы системы данных;**

**- отображение информации об оборудовании системы, карточек операторов, справочников;**

**- отображение результатов выполнения нарядов, задач и операций вместе с прикрепленными фотографиями и другой сервисной информацией;**

**- отображение подробных журналов работы операторов МК ТОиР.**

## **Структура нарядов**

**Суть работы по обслуживанию заключается в генерации на сервере и передаче обслуживающему персоналу задач и получение результата их выполнения.**

**На верхнем (первом) уровне иерархии задач системы стоит наряд [order]. Наряд выдается оператору МК на определенный отчетный период (обычно это день). Примером типа наряда может быть «обследование проливочной линии», «плановый обход котельных» или «ремонт насосной группы».**

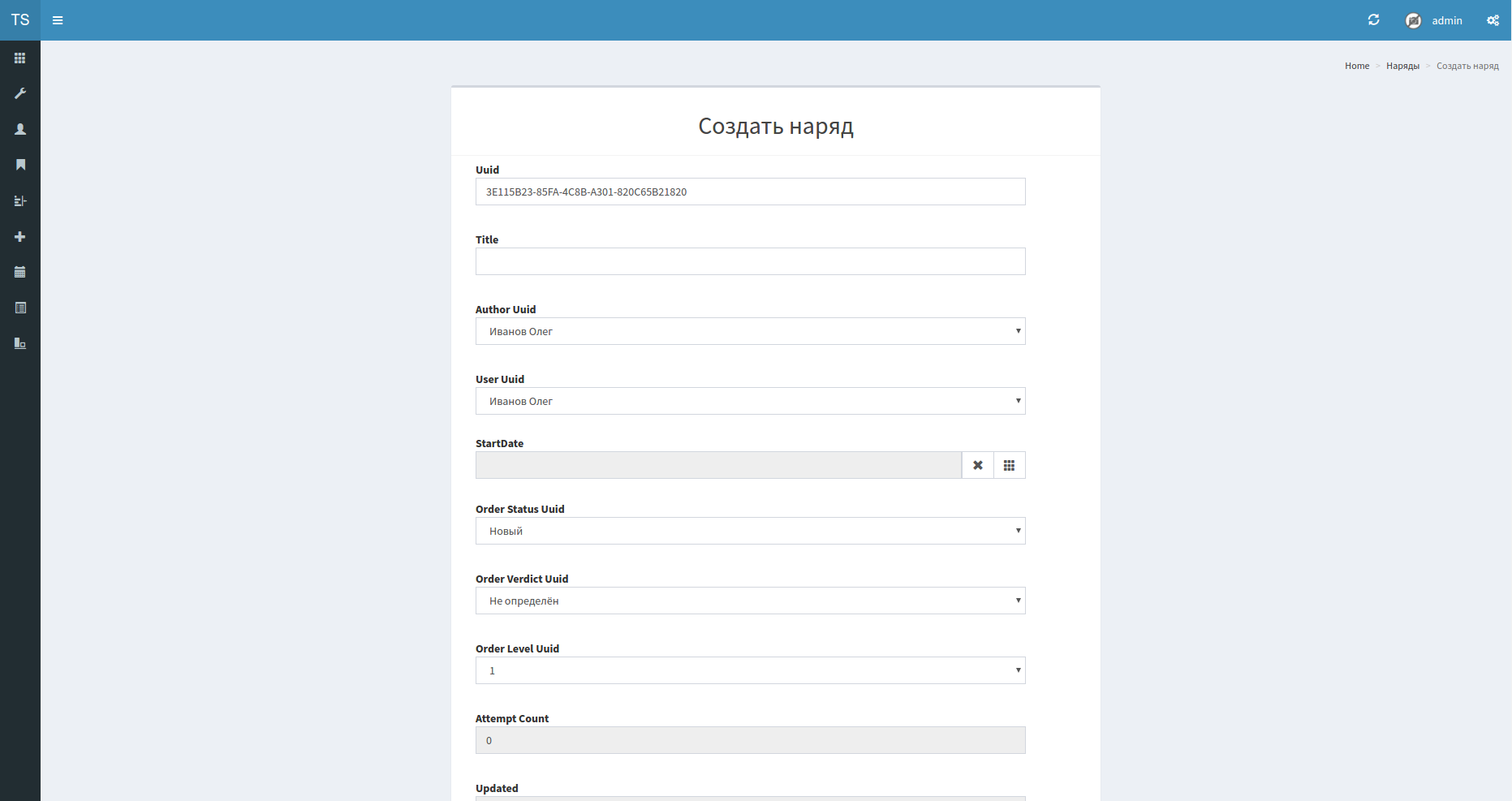


Рисунок 1.

**На втором уровне находятся задачи [task], которые относятся к данному наряду. Внутри наряда таких задач может быть несколько. Например, «проверка задвижек на входе рабочего стола», «осмотр котельной К2», «демонтаж выходных насосов». Задача уже связана с объектом или оборудованием.**

**На третьем уровне задача разбивается на этапы [task\_stage]. Каждый этап задачи содержит конкретную работу в отношении конкретного экземпляра оборудования. Например, «устранение протечки на зажимном устройстве», «осмотр теплообменника» или «снятие задвижки на входе».**

**Четвертый уровень состоит из элементарных операций [operation]. Каждый шаг операции обозначает элементарное действие над оборудованием. Например, «открутить четыре болта», «осмотреть теплообменник на предмет утечек», «слить воду с линии». Операции, представляют из себя чек-лист, к каждому пункту которого можно прикрепить комментарии, фотографии, установить статус и дефект.**

## **Минимальный состав технических средств**

**В качестве технического средства для функционирования сервера ТОИРУС должен применяться сервер с характеристиками (не менее):**

**2x Xeon E5645 6C 12M Cache 2.40 GHz • 24Gb (6x4Gb) DDR3 ECC 1333МГц HDD (2Тб, память из расчета 1Тб на тысячу RFID-меток).**

**Клиентские машины, с которых будет осуществляться доступ к серверу, могут быть с любой версией операционной системы от Windows 95 до Linux или BSD. Основное требование – наличие сетевой платы Ethernet / Wi-Fi, доступа к сети, в которой расположен сервер и наличие предустановленного браузера (Mozilla Firefox, Google Chrome).**

## **Требования к персоналу.**

**Минимальное количество персонала, требуемого для работы программного обеспечения системы, должно составлять не менее двух штатных единиц - системный администратор и конечный пользователь программы - оператор. При этом подразумевается, что операторами системы будут являться бригадиры и руководители ремонтных подразделений.**

**Системный администратор должен иметь высшее профильное образование.**

**В перечень задач, выполняемых системным администратором, должны входить:**

**- задача поддержания работоспособности технических средств;**

**- задачи установки (инсталляции) и поддержания работоспособности системных программных средств - операционной системы;**

**- задача настройки программного обеспечения;**

**- задачи периодического резервирования и восстановления ключевых элементов комплекса;**

**- задачи редактирования общих свойств системы, влияющих на ее непосредственное функционирование (интеграцию с другими системами, расписание генерации нарядов и т.п.);**

**- задачи устранения проблем, возникающих во время функционирования системы.**

# **Структура и описание программного обеспечения**

**Программный комплекс системы сбора состоит из нескольких компонентов: базы данных MySQL, сервера системы (с API внешнего доступа) (Yii2), веб-сервера (Nginx), клиентских АРМ и различных сервисов.**

**Конфигурация системы происходит с помощью интерфейса путем добавления устройств, объектов, технологических карт, пользователей, наполнения справочников. Веб-интерфейс, также позволяет просматривать результаты выполнения нарядов с прикрепленными к ним фотографиями, комментариями, зафиксированными дефектами и установленными справочниками, просматривать показания счетчиков.**

**Время хранения данных на сервере ограничено лишь объемом дискового пространства.**

## **Структура данных системы.**

**Логическая структура данных системы совпадает с структурой таблиц базы данных. Вкратце ее описание выглядит следующим образом.**

**В системе существует набор базовых справочников:**

**- Справочник уровней критичности. Необходим для распределения оборудования или производимых над ним работ по уровню критичности и градуируется от уровня 1 (низкий уровень) до уровня 5 (максимальный уровень);**

**- Справочник типов документации. Хранит типы документов, хранимых на всех уровнях системы (руководства по эксплуатации, паспорта и т.д.);**

**- Справочник статусов нарядов/задач/этапов задач/операций. Содержит варианты текущего состояния выполнения (создан, в работе, сдан и т.д.);**

**- Справочник вердиктов нарядов/задач/этапов задач/операций. Содержит варианты вердиктов окончания (завершен успешно, не завершена, оборудование отсутствует, обнаружены проблемы);**

**- Справочник типов операций над оборудованием (осмотр, ремонт и т.д.);**

**- Справочник статусов оборудования. Содержит текущие статусы оборудования (исправно, неисправно, в ремонте, отсутствует и т.д.);**

**- Справочник типов измеряемых величин (типы измеряемых параметров – температура, длина, давление и т.д.);**

**- Справочник типов оборудования. Хранит типы обслуживаемого в системе оборудования (задвижки, расходомеры и т.д.);**

**- Справочник статусов оборудования (в работе, неисправно, демонтировано);**

**- Справочник предупреждений;**

**- Справочник типов инструментов;**

**- Справочник дефектов.**

**В системе существуют базовые информационные таблицы, содержащие информацию о конкретном экземпляре системы, то есть о ее элементах:**

**- Таблица объектов содержит информацию о зданиях, сооружениях, предприятиях и других объектах, на которых располагается обслуживаемое оборудование;**

**- Таблица оборудования хранит информацию обо всех элементах системы, подлежащих обслуживанию;**

**- Таблица документации содержит все документы, хранящиеся на клиенте или сервере.**

**В качестве документации могут быть использованы файлы в следующих форматах:**

**- текстовые \*.txt;**

**- документы \*.pdf, \*.docx;**

**- изображения \*.jpg, \*.png;**

**- Таблица контрагентов хранит данные о компаниях исполнителей/подрядчиков и клиентов/заказчиков;**

**- Таблица бригад содержит данные о группах/бригадах/отделах в которые объединены пользователи/исполнители;**

**- Таблица пользователей содержит записи обо всех операторах мобильного комплекса;**

**- Таблицы нарядов, операций, задач, этапов задач;**

**- Таблица трека передвижения оператора;**

**- Журнал событий;**

**- Таблица запасных частей;**

**- Таблица инструментов.**

## **Сведения о структуре программного обеспечения**

Основным компонентами сервера, являются nginx, php, mysql и yii2, обеспечивающие работоспособность системы.

В ПО ТОИРУС идет полное следование принципам архитектурного дизайна MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер» — схема разделения данных приложения, [пользовательского интерфейса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

- Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние;

- Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели;

- Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

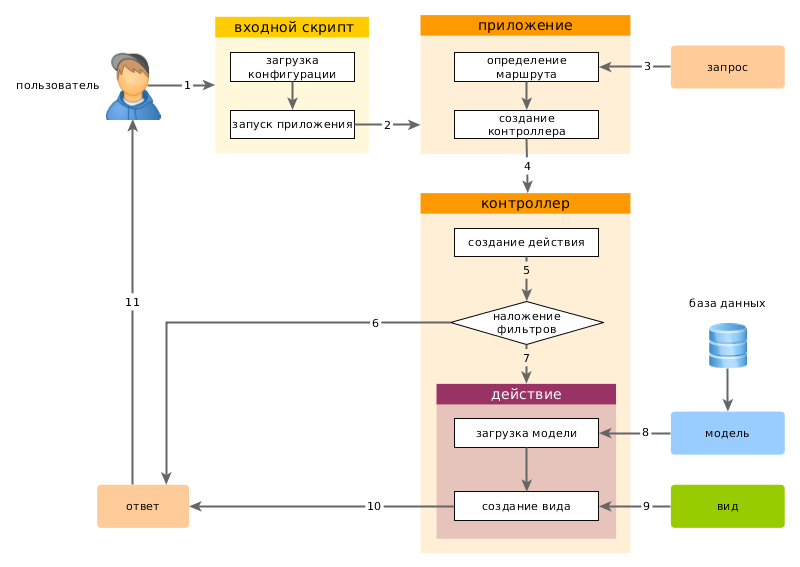
## **Подсчет времени обслуживания**

**Одной из основных задач комплекса ТОИРУС является учет времени на выполнение наряда и его составных частей. В системе минимальной единицей работы является операция над оборудованием, время которой строго регламентировано техническими документами. Сумма нормативов времен выполнения всех операций дает норматив выполнения этапа задачи, а сумма нормативов этапов задач – всей задачи в целом.**

**Нормировать выполнение всего наряда не всегда возможно, так как при его расчете необходимо учесть несколько независимых факторов, главный из которых время на перемещение между экземплярами оборудования.**

**Мобильный клиент автоматически фиксирует время начала и окончания обслуживания, может косвенно оценить время выполнения всего наряда исходя из времени сканирования персональной идентификационной карты пользователя, привязав к этому действию начало и конец работы.**

**Структурная схема программного обеспечения приведена на следующем рисунке.**

Рисунок 2.

# **Установка программного обеспечения**

**На сервере предварительно установлена операционная система с пакетами:**

**- веб-сервер Nginx (конфигурационные файлы /etc/nginx);**

**- транслятор языка php 7 (/etc/php/);**

**- база данных MySQL 5.5 и выше (/etc/mysql);**

**- система контроля версий - git;**

**- менеджер php пакетов composer.**

**Дополнительно системным администратором могут быть установлены другие пакеты и сервисные программы в соответствии с документацией на операционную систему.**

**После того, как мы удостоверились в наличии следующих пакетов, мы можем приступить к следующим шагам подготовки к развертыванию системы.**

## **Установка зависимых пакетов**

Первым делом, после установки всех базовых пакетов по обустройству минимального окружения для развертывания ПК ТОИРУС, необходимо произвести установку расширений и модулей для php.

sudo apt-get install php7.0-mbstring libapache2-mod-php7.0 php-xml php7.0-zip

Данная строка, произведет установку всех модулей от которых зависит дальнейшее развертывание системы.

Далее, нам необходимо произвести установки пакетов, которые собственно и будут в дальнейшем производить комплекс мероприятий по развертыванию ПК ТОИРУС.

sudo apt-get install composer

Как только мы установили composer, нам необходимо установить связку пакетных менеджеров, следующей командой:

sudo composer global require "fxp/composer-asset-plugin:^1.3.1"

sudo apt-get install git

После данных команд, нам необходимо клонировать с удаленного репозитория актуальную версию системы, следующей командой:

git clone <https://github.com/mikaelwasp/toir-server> имя\_сайта

Сейчас у нас есть настроенное рабочее окружение, готовое к работе с системой. Приступим к вытягиванию всех необходимых пакетов и их зависимостей по файлу зависимостей, composer.lock, следующеми командами:

cd /var/www/имя\_сайта

sudo composer install

По окончанию установки пакетов и их зависимостей, мы можем приступить к настройке.

Для того, чтобы взаимодействие нашего продукта и базы данных происходило, необходимо настроить файл main-local.php:

<?php

return [

'components' => [

'db' => [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=localhost;dbname=ИМЯ\_БД',

'username' => 'ИМЯ\_ПОЛЬЗОВАТЕЛЯ\_БД',

'password' => 'ПАРОЛЬ\_ПОЛЬЗОВАТЕЛЯ\_БД',

'charset' => 'utf8',

],

'mailer' => [

'class' => 'yii\swiftmailer\Mailer',

'viewPath' => '@common/mail',

// send all mails to a file by default. You have to set

// 'useFileTransport' to false and configure a transport

// for the mailer to send real emails.

'useFileTransport' => true,

],

],

];

Теперь можно собственно и создать базу данных к которой мы обращаемся через файл настроек yii2.

mysql -uИМЯ\_ПОЛЬЗОВАТЕЛЯ\_БД -pПАРОЛЬ\_ПОЛЬЗОВАТЕЛЯ\_БД

Как только мы вошли в командную строку mysql, необходимо выполнить:

CREATE DATABASE ИМЯ\_БД;

Данной командой мы создали базу данных к которой и будет обращаться наша система.

Также нужно проверить, есть ли в настройках по умолчанию, поле NO\_ZERO\_DATE.

Это мы можем узнать следующей командой в оболочке mysql:

select @@sql\_mode;

Вывод может быть следующий:

ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION

Как вы видите, у нас есть поле NO\_ZERO\_DATE. Его необходимо отключить. Сделать это мы можем следующим способом, нам необходимо установить в конфигурационные файлы mysql, директиву с полями, которые должны быть обработаны встроенными валидаторами mysql.

sudo nano /etc/my.cnf

Строка, которая будет загружать необходимые правила валидации mysql.

[mysqld]

sql\_mode=ONLY\_FULL\_GROUP\_BY,NO\_ZERO\_IN\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION

Данные изменения сохраняем в самом начале файла.

Также для полной уверенности, что изменения будут активированы, сюда:

sudo nano /etc/mysql/my.cnf

После изменений, перезагружаем mysql, командой:

sudo /etc/init.d/mysql restart

Теперь осталось вернуться в корень нашего сайта и запустить миграции базы данных.

sudo php yii migrate

# **Запуск и настройка программного обеспечения**

**Настройка сервера должна проводиться специалистом, соответствующего уровня. Ввод объектов, устройств и каналов и их настройка осуществляется с помощью вкладок веб-интерфейса.**

**Клиентские машины, с которых будет осуществляться доступ к данным системы сбора, могут быть с любой версией операционной системы от Windows 95 до Linux или BSD. Основное требование – наличие сетевой платы Ethernet / Wi-Fi и наличие предустановленного браузера (Mozilla Firefox, Google Chrome).**

**Доступ к интерфейсу администрирования осуществляется с помощью интерфейса, который доступен по адресу [http://admin.toir.tehnosber.ru].**

**Сервисы веб-сервера, базы данных запускаются автоматически при старте операционной системы. Если запуск сервиса по каким-то причинам не произошел, то можно или перезапустить сервер, или запустить сервис вручную.**

**Для решения нестандартных ситуаций и более детального администрирования системы возможно подключение к серверу по протоколу «ssh». Для этого необходимо осуществить подключение с помощью стандартных программ типа «putty» на порт 22 (порт может быть изменен администратором сети) с кодировкой UTF8, ввести пароль для пользователя (user) или администратора (root) при необходимости.**

**Рабочая директория сервиса находится по адресу [/var/www/toirus]. При необходимости запустить вручную сервис сбора следует произвести следующий набор операций:**

**Сервис ведет лог-файл [/var/www/toirus/vagrant/nginx/log/]. Разбор содержимого этого файла не входит в рамки компетенции данного документа и должен быть разобран отдельно.**

**База данных MySQL ведет свои лог-файлы в директории[/var/log].**

**Веб-сервер ведет свои лог-файлы в директории [/var/log/nginx].**

# **Описание API**

Доступ к данным системы осуществляется через методы взаимодействия API.

Архитектура API построена на многоуровневом(многослойном) взаимодействии, т.е. каждый слой API несет в себе функциональную нагрузку, которая упрощает работу, более глубоким слоям системы.

Основной задачей API, является поддержание безопасности хранимых данных, через создание интерфейсов с многоуровневой системой безопасности и активного взаимодействия с внешней средой – интернет.

Данную систему API, можно представить из следующих структур:

- Менеджер соединений;

- Интерфейсы запросов;

- Регуляторы целостности данных.

Менеджер соединений несет роль распределителя запросов и является первым валидатором входного запроса. В случае ошибки валидации на этапе обработки запроса менеджером соединений, последний отказывает входному запросу в транспортировке на следующий уровень обработки, тем самым разгружая интерфейсы запросов от ненужного, более глубокого анализа. В ситуации, когда первичная валидация менеджера соединений прошла успешно, менеджер определяет тип входного запроса и отправляет его на более глубокий уровень взаимодействия, соответствующий типу запроса.

Когда запрос прошел валидацию менеджера и попал на интерфейс запроса, соответствующего типа, над ним начинаются вторичная валидация и анализ входных данных. По окончанию данного взаимодействия, результат обработки может быть, как положительным, так и отрицательным, т.е. запрос удовлетворяет или не удовлетворяет заданным проверкам.

Когда входной запрос не удовлетворяет проверки интерфейса запроса, результат обработки передается на менеджера соединений, который в свою очередь информирует клиента о не успешном выполнении запроса.

Обратной стороной вышесказанного события, может быть удачная вторичная валидация и анализ входного запроса. В такой ситуации, определённый метод интерфейса запроса, передает результат своей работы сборщику данных, который в дальнейшем будет “мостом” взаимодействия регулятора целостности хранимых данных и интерфейса запроса.

По окончанию проверки регулятором, данных, которые транспортировал к нему сборщик от интерфейса запроса, регулятор выносит решения по результатам обработки. Если регулятор удостоверяется, что работа интерфейса запроса была выполнена в соответствии с заданным алгоритмом, он открывает канал соединения для интерфейса запроса, чьи данные переносил сборщик, тем самым предоставляя доступ к информации.

По окончании выборки, когда интерфейс запроса получил необходимый блок данных, регулятор целостности хранимых данных закрывает канал соединения.

Для следующей выборки, необходимых данных, интерфейсу запроса придется вновь совершить вышеуказанный алгоритм получения доступа к информации.

## Типы запросов

Запросы, которые получает API не все могут быть обработаны успешно. В архитектуре API заложена строгая логика обработки входящих запросов, которая необходима для классификации входящих обращений и перенаправлении на интерфейсы запросов соответствующего типа.

Можно выделить три основных типа запроса:

- Основной запрос;

- Специальный запрос;

- Специфический запрос.

Под основным типом запроса понимают, что клиент производит входящие обращение к API по заданному алгоритму обращения, соответствующий правам доступа клиента.

Под специальным запросом понимают, что обращение направлено от специалиста, который является администратором системы. Данные запросы направлены на получение информации о состоянии машин, приложения и т.п. (внутренняя информация).

Под специфическим запросом понимают, что запрос, который отправил клиент не тривиален и не может быть обработан или является ошибочным.

Основные запросы могут подразделяться на следующие категории:

- Запросы доступа;

- Информационный запрос;

- Справочный запрос.

Запрос доступа — это входное обращение направление на получения доступа к полноценному взаимодействию с API.

Информационный запрос — это входное обращение, направленное на получение информации по объекту запроса.

Справочный запрос — это входное обращение, направленное на получение справочной информации.

Специальные запросы могут подразделяться на:

- Системные запросы;

- Запросы к приложению;

- Запросы к окружению.

Системный запрос — это входное обращение, направленное на взаимодействия с сервером.

Запрос к приложению — это входное обращение, направленное на взаимодействие с самим приложением.

Запросы к окружению — это входное обращение, направленное на получение доступа к окружению сервера.

Специфические запросы могут подразделяться на:

- Ошибочные запросы по спецификации;

- Ошибочные запросы не по спецификации.

Ошибочные запросы по спецификации — это входные обращения, которые несут в себе некорректную информацию, которая предусмотрена документально.

Ошибочные запросы не по спецификации — это входные обращения, которые несут в себе некорректную информацию, которая не предусмотрена документально.

Данные категории также имеет подразделы, которые определяют их функциональное предназначение.

Запросы доступа могут подразделяться на:

- Запросы на авторизацию;

- Запросы на предоставление доступа к секциям повышенного уровня.

Запросы на авторизацию — это входное обращение, направление на получения токена доступа, который позволит производить расширенный список взаимодействий с системой.

Запросы на предоставление доступа к секциям повышенного уровня — это входное обращение направление на получение дополнительных привилегий при работе с системой к определенным участкам (секциям).

Справочный запрос может подразделяться на:

- По типу доступа;

- По типу справочной информации;

- По полноте.

Справочный запрос, отправленный от авторизованного пользователя имеет право к более детальной и точной информации по взаимодействию с системой.

Справочный запрос, отправленный с определённым аргументом, может быть как по команде, так и по списку всех команд.

Справочный запрос, отправленный с определенным аргументом на указание полноты или краткости ответа.

## Алгоритм авторизации

Для авторизации в системе, необходимо произвести POST запрос к API с определенным выражением и идентификационной меткой. Сервер получивший данный запрос, разберет полученной выражение из запроса и сформирует запрос к таблице пользователей. После необходимой проверки, если пользователь есть в системе и у него активный статус, ему генерируется и отправляется токен доступа, соответствующий правам доступа пользователя в базе данных.

После того, как у клиента есть токен доступа, он может производить активное взаимодействие с API. Например, формировать информационный запрос к базе данных. Структура данного обращения будет состоять из GET запроса, выражения и токена доступа. Обработав данное обращения, если оно корректное, API вернет ответ на полученный запрос.

Аналогичным образом строится и все дальнейшие обращения к API.

# **Описание интерфейса оператора**

**Интерфейс пользователя предназначен для визуализации информации по узлам учета, подключенным к серверу с помощью интернет-браузера. Интерфейс состоит из нескольких разделов, доступ к которым осуществляется после авторизации.**

Интерфейс предназначен для доступа большого числа пользователей к информации, сосредоточенной в базе данных. Доступ осуществляется через Интернет. На компьютере пользователей не устанавливаются никакие дополнительные компоненты. Достаточным условием является наличие любого графического веб-браузера.

**Интерфейс обеспечивает защиту доступа к редактированию данных. Добавление информации возможно только после предъявления логина и пароля, которые присваиваются каждому пользователю. Кроме того, обеспечивается разграничение доступа.**

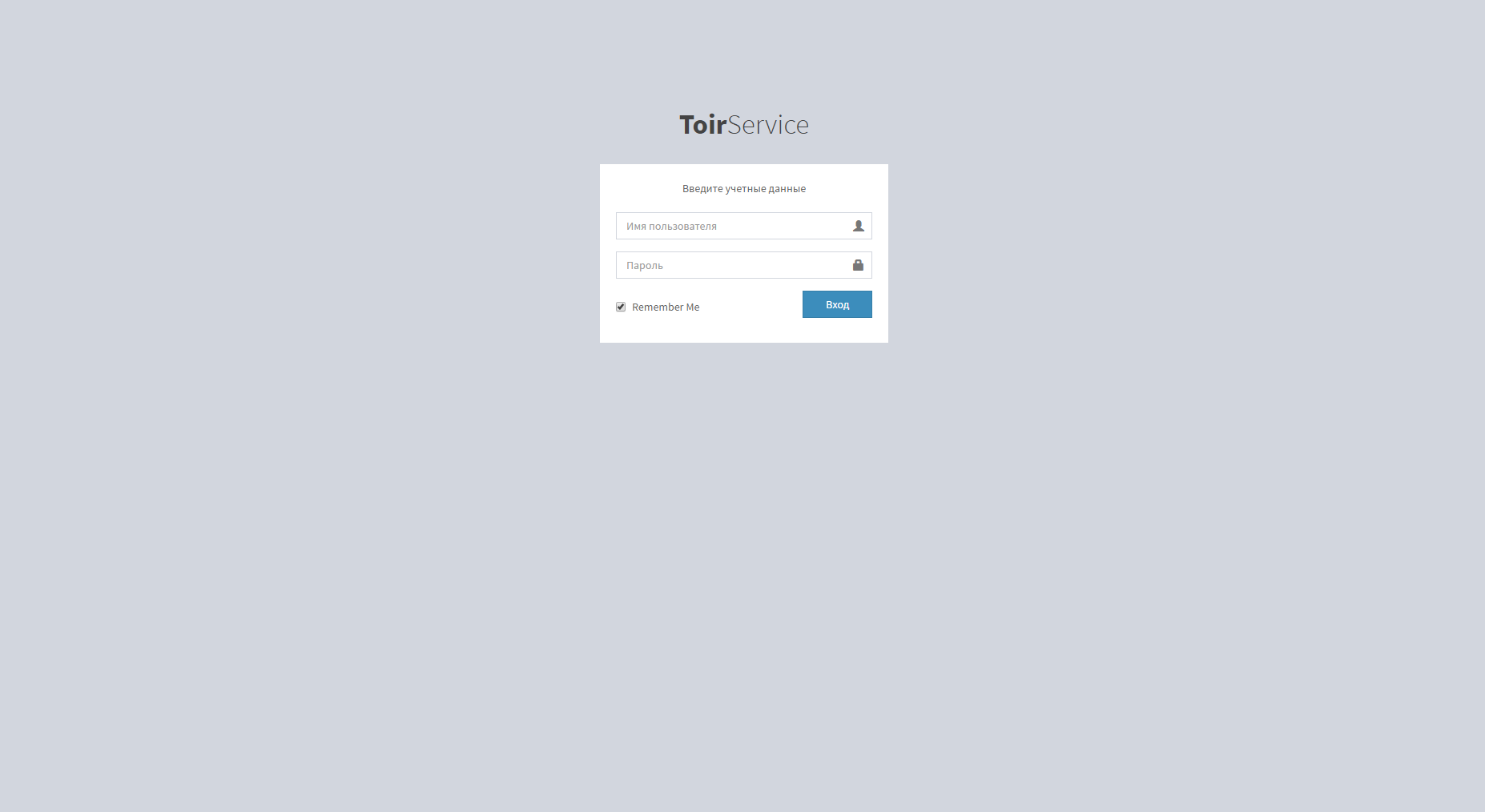


Рисунок 3.

## **Главный экран интерфейса.**

Главный экран интерфейса системы после авторизации выглядит следующим образом.

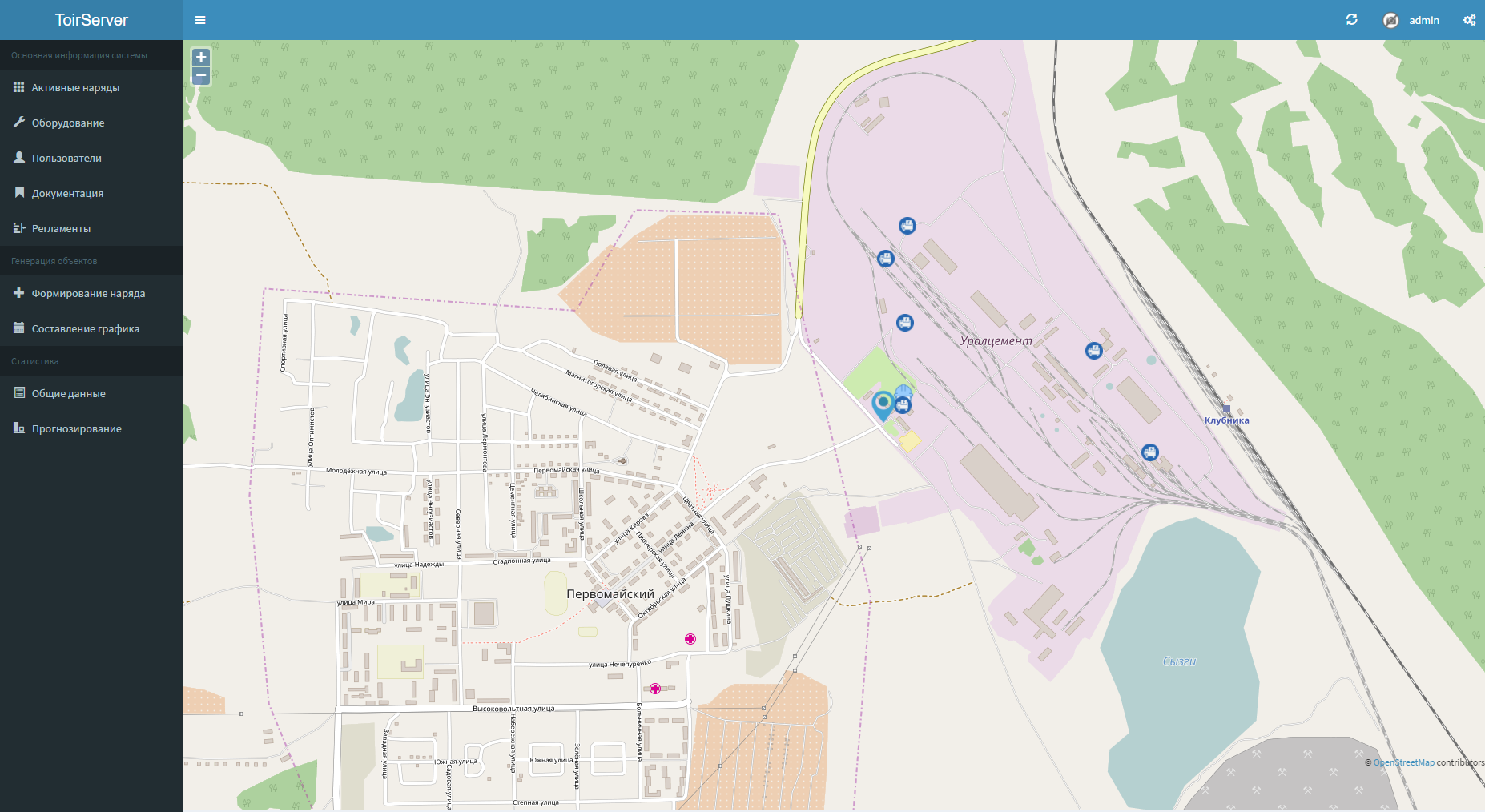


Рисунок 4.

Как видно из рисунка выше, главный экран разбит на следующие смысловые блоки:

- Карта;

- Левая панель;

- Верхняя панель.

На карте отображаются основные объекты мониторинга и учета, за которыми необходим визуальный контроль. К данным объектам можно отнести операторов, оборудование, объекты (сооружения, котельные и т.п.).

С помощью левой боковой панели, мы можем производить манипуляции с хранимыми данными, а также анализировать отображаемые данные.

Верхняя панель имеет три основные иконки. Слева направо, иконка загрузки и отображения журнала полевого оператора, иконка пользователя, которая позволяет производить настройки пользователи, а также производить выход из системы и иконка, вызывающая панель управления, через которую возможно полное взаимодействие с системой.

Следующие изображения иллюстрирует вышеописанные возможности интерфейса главной страницы.

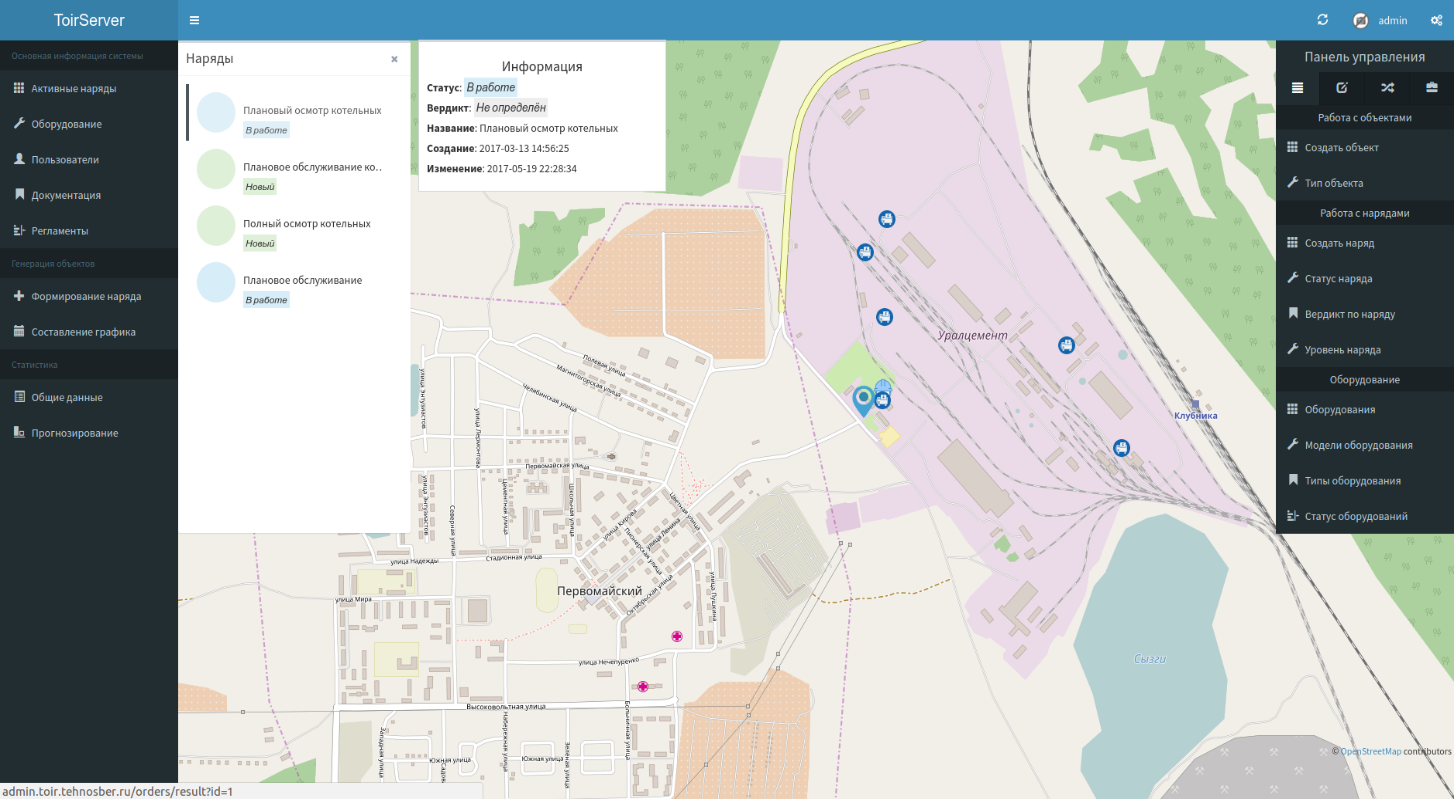


Рисунок 5.

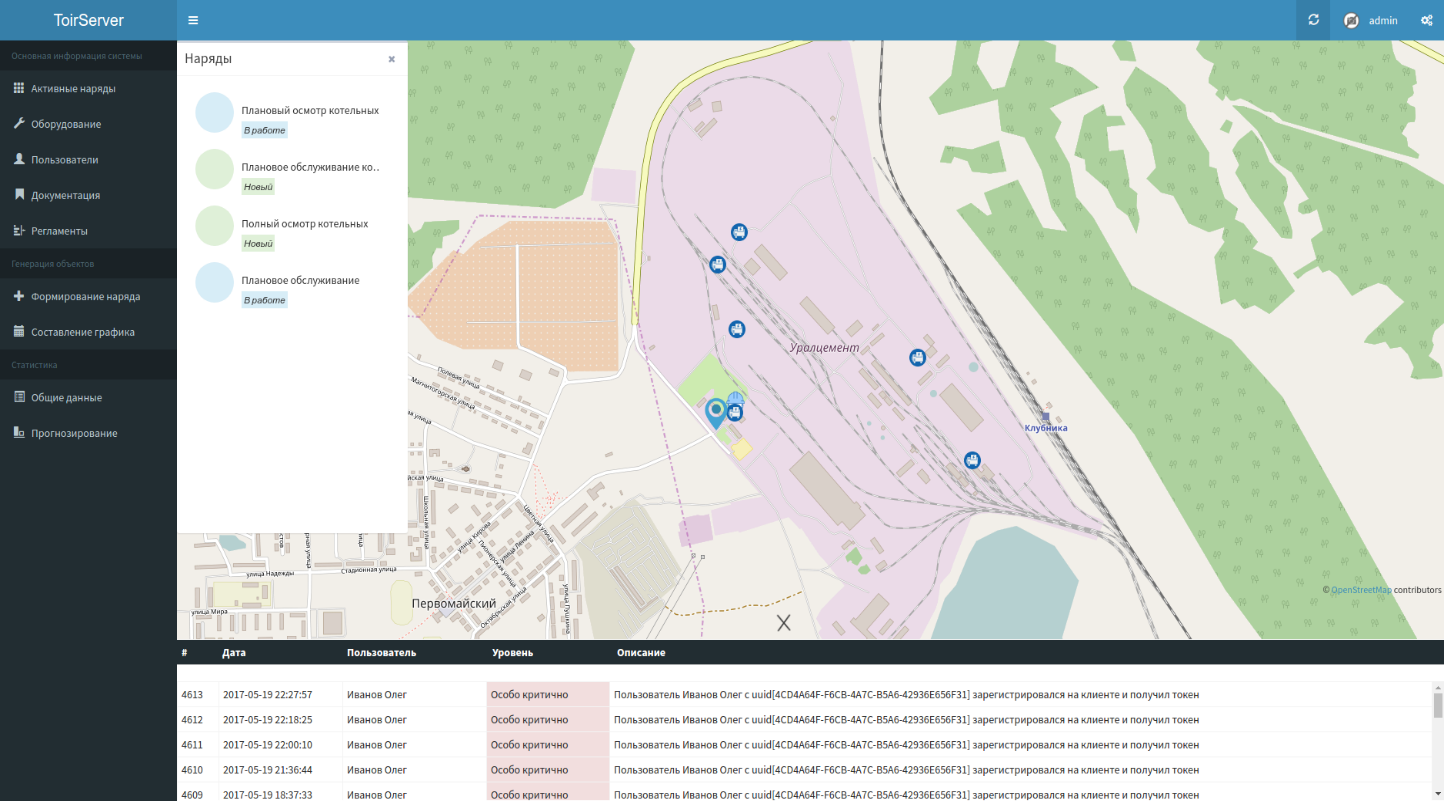


Рисунок 6.

## **Шаблоны**

**Под шаблоном задачи, этапа, операции подразумевается совокупность действий, необходимых для ее выполнения. Например, над одним типом оборудования можно совершить несколько вариантов действий-этапов задач (ремонт, замена и т.д.), каждый из которых будет иметь свой собственный шаблон, который может динамически назначаться оборудованию.**

**ПО ТОИРУС выбирает в базе данных соответствующий шаблон и ищет его это первый шаг. К каждому шагу привязываются варианты результатов выполнения, выбираемые из таблицы [operation]. Каждый вариант представлен в интерфейсе своей кнопкой, по нажатию на которую происходит переход на связанный с ним следующий шаг обслуживания. Такая цепочка выполняется до тех пор, пока шаг не будет помечен свойством «последний».**

**Каждый шаг шаблона операции соответствует одному элементарному действию типа «открутить гайку» или «осмотреть крепление».**

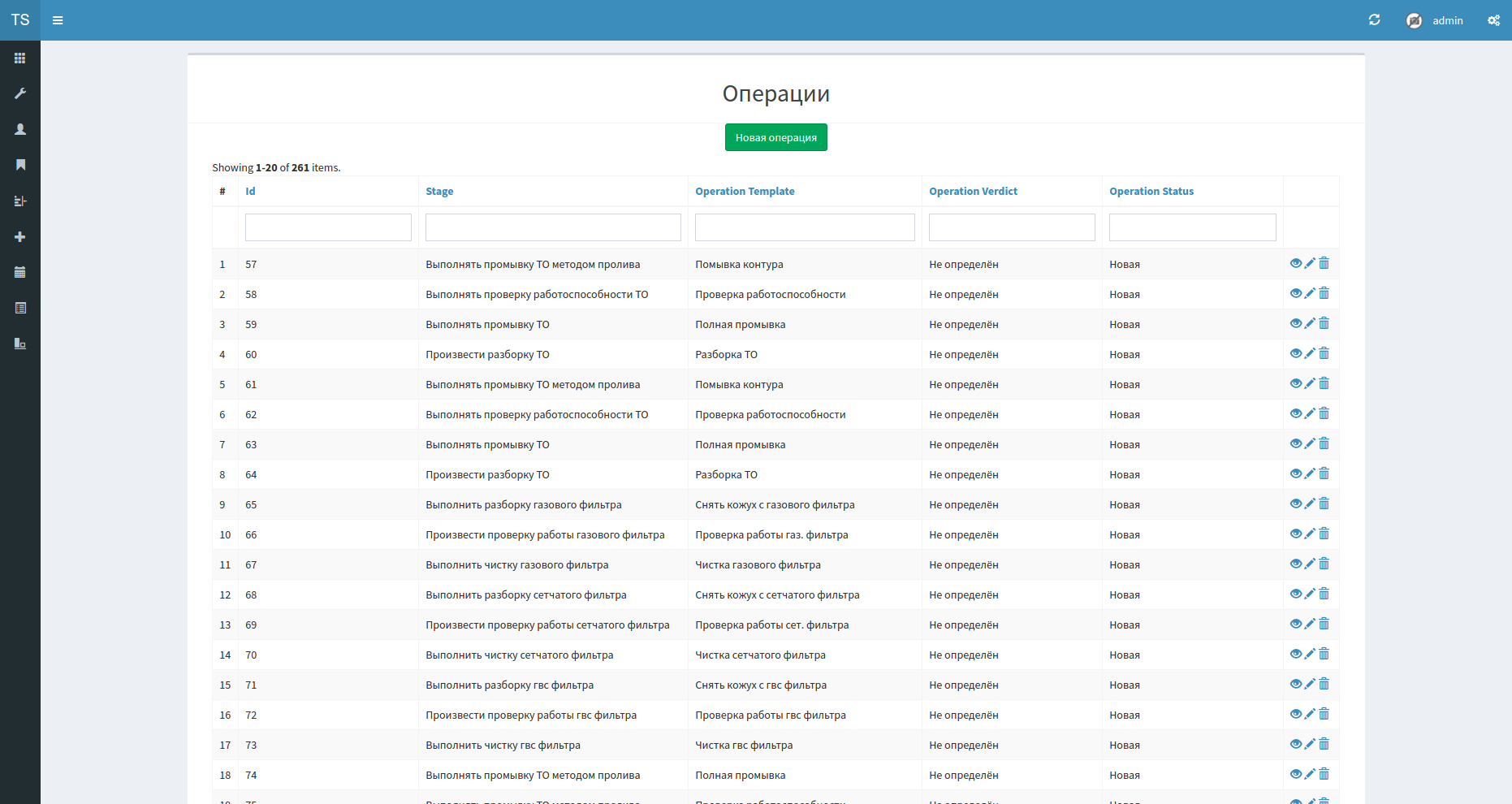


Рисунок 7.

# **Работа с базой данных**

**База данных состоит из рабочих и сервисных таблиц, справочников и журналов таблиц. Журналы необходимы для фиксации времени и типа действия над записями таблиц нарядов и операций. Все таблицы также имеют поля createdAt и changedAt, первое из которых хранит дату создания записи, а второе дату ее изменения.**

**Поля «string – uuid» хранят класический uuid в текстовой форме с количеством 50 символов.**

**Текущая версия базы данных и ее имя определяются в файле toir-base. В папке [/var/www/console/**migrations/**] хранятся все изменения, вносимые в структуру базы данных, начиная с первой версии.**

**Ниже приведена структурная схема и взаимосвязи между таблицами в базе данных.**

## **Таблица типов оповещений [**alerttype**]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица бригады [brigade]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| contragentUuid | Varchar(50) | У/н контрагента |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица пользователей бригады [brigadeuser]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| brigadeUuid | Varchar(50) | У/н бригады |
| userUuid | Varchar(50) | У/н пользователя |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица контрагентов [contragent]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| name | Varchar(150) | Имя |
| description | Text | Описание |
| phone | Varchar(150) | Телефон |
| contragentType | Varchar(50) | Тип контрагента |
| parentContragent | Varchar(50) | Старший контрагент |
| adress | Varchar(150) | Адрес |
| photo | Varchar(150) | Фотография |
| longitude | double | Широта |
| latitude | double | Долгота |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица пользователей контрагентов [contragentuser]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| contragentUuid | Varchar(50) | У/н контрагента |
| userUuid | Varchar(50) | У/н пользователя |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов критичности [criticaltype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица дефектов [defect]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| userUuid | Varchar(50) | У/н пользователя |
| date | Timestamp | Время обнаружения |
| equipmentUuid | Varchar(50) | У/н оборудования |
| defectTypeUuid | Varchar(50) | Тип дефекта |
| process | tinyint(1) | Процесс |
| comment | Varchar(100) | Комментарий |
| taskUuid | Varchar(50) | У/н задачи |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов дефектов [defecttype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| equipmentTypeUuid | Varchar(50) | Тип оборудования |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица документации [documentation]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| equipmentUuid | Varchar(50) | Оборудование |
| documentationTypeUuid | Varchar(50) | Тип документации |
| title | Varchar(100) | Название |
| path | Text | Путь |
| required | Int(10) | Включать в загрузке |
| equipmentModelUuid | Varchar(50) | Модель оборудования |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов документации [documentationtype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица оборудования [equipment]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| equipmentModelUuid | Varchar(50) | Модель оборудования |
| title | Varchar(100) | Название |
| criticalTypeUuid | Varchar(100) | Критический тип |
| startDate | Timestamp | Дата установка |
| latitude | Double | Широта |
| longitude | Double | Долгота |
| tagId | Varchar(50) | Метка |
| image | Varchar(200) | Изображение |
| equipmentStatusUuid | Varchar(50) | Статус оборудования |
| inventoryNumber | Varchar(50) | Инвентарный номер |
| location | Varchar(100) | Локация |
| parentEquipment | Varchar(50) | Старшее оборудование |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица моделей оборудований [equipmentmodel]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| equipmentTypeUuid | Varchar(100) | Тип оборудования |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица статусов оборудования [equipmentstatus]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов оборудования [equipmenttype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица геоположения [gpstrack]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| userUuid | Varchar(50) | У/н пользователя |
| latitude | Double | Широта |
| longitude | Double | Долгота |
| date | Timestamp | Время фиксации |

## **Таблица журнала [journal]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| userUuid | Varchar(50) | У/н пользователя |
| description | Text | Описание |
| date | Timestamp | Время фиксации |

## **Таблица измеренных значений [measuredvalue]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| equipmentUuid | Varchar(50) | У/н оборудования |
| operationUuid | Varchar(50) | У/н операции |
| date | Timestamp | Время обнаружения |
| value | Varchar(50) | Значение |
| measureTypeUuid | Varchar(50) | Тип измерения |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов измерений [measuretype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов объектов [objects]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| objectType | Varchar(50) | Тип объекта |
| parent | Varchar(50) | Старший объекь |
| title | Varchar(100) | Название |
| description | Text | Описание |
| photo | Varchar(150) | Фотография |
| latitude | Double | Широта |
| longitude | Double | Долгота |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типа объекта [objecttype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| description | Text | Описание |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица операций [operation]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| taskStageUuid | Varchar(50) | Этап задачи |
| operationVerdictUuid | Varchar(50) | Вердикт операции |
| operationStatusUuid | Varchar(50) | Статус операции |
| operationTemplateUuid | Varchar(50) | Шаблон операции |
| date | Timestamp | Время фиксации |
| endDate | Timestamp | Завершение операции |
| flowOrder | Int(10) | Поле сортировки |
| comment | Varchar(100) | Комментарий |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица для хранения файлов после операций [operationfile]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| operationUuid | Varchar(100) | Название |
| fileName | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица статуса операций [operationstatus]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица шаблона операций [operationtemplate]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| description | Text | Описание |
| image | Varchar(200) | Изображение |
| normative | Int(10) | Норматив |
| firstStep | Int(10) | Шаг вперед |
| lastStep | Int(10) | Шаг назад |
| operationTypeUuid | Varchar(50) | Тип операции |
| equipmentModelUuid | Varchar(50) | Модель оборудования |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица инструментов операций [operationtool]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| operationTemplateUuid | Varchar(50) | Шаблон операции |
| toolsUuid | Varchar(50) | Инструмент |
| quantity | Int(10) | Количество |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов операций [operationtype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица вердиктов операций [operationverdict]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица уровня наряда [orderlevel]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица нарядов [orders]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| authorUuid | Varchar(50) | Автор наряда |
| userUuid | Varchar(50) | Исполнитель наряда |
| recieveDate | Timestamp | Дата получения наряда |
| startDate | Timestamp | Начала выполнения |
| openDate | Timestamp | Открытое время |
| closeDate | Timestamp | Дата завершения |
| orderStatusUuid | Varchar(50) | Статус наряда |
| orderVerdictUuid | Varchar(50) | Вердикт наряда |
| attemptSendDate | Timestamp | Время отправки |
| attemptCount | int(10) | Количество отправок |
| updated | int(10) | Обновления |
| orderLevelUuid | Varchar(50) | Уровни наряда |
| customerUuid | Varchar(50) | Клиент |
| perpetratorUuid | Varchar(50) | Исполнитель |
| comment | Varchar(100) | Комментарий |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица статусов наряда [orderstatus]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица вердиктов наряда [orderverdict]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица обновлений параметров [referenceupdate]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| referenceName | Varchar(100) | Название обновления |
| updateDate | Timestamp | Дата обновления |

## **Таблица частей ремонта [repairpart]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| equipmentModelUuid | Varchar(50) | Модель оборудования |
| repairPartTypeUuid | Varchar(50) | Тип части инструмента |
| commonRepairPartFlag | int(10) | Общий флаг части ремонта |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов частей ремонта [repairparttype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| parentRepairTypeUuid | Varchar(100) | Старшая часть инструмента |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица этапов частей ремонта [stagerepairpart]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| repairPartUuid | Varchar(50) | Часть инструмента |
| stagePatternUuid | Varchar(50) | Шаблон этапа |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица задач [task]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| comment | Varchar(100) | Комментарий |
| orderUuid | Varchar(50) | У/н наряда |
| equipmentUuid | Varchar(50) | У/н оборудования |
| taskStageStatusUuid | Varchar(50) | Статус задач |
| taskStageVerdictUuid | Varchar(50) | Вердикт задачи |
| taskTemplateUuid | Varchar(50) | Шаблон задачи |
| startDate | Timestamp | Дата начала |
| endDate | Timestamp | Дата завершения |
| prevCode | Int(10) | Предыдущий код |
| nextCode | Int(10) | Следующий код |
| flowOrder | Int(10) | Фильтр отображения |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица задач частей ремонта [taskrepairpart]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| taskUuid | Varchar(50) | У/н задачи |
| repairPartUuid | Varchar(50) | Часть инструмента |
| measureTypeUuid | Varchar(50) | Тип измерения |
| quantity | Int(10) | Количество |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица этапов задач [taskstage]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| comment | Varchar(100) | Комментарий |
| taskUuid | Varchar(50) | У/н задачи |
| equipmentUuid | Varchar(50) | У/н оборудования |
| taskStageStatusUuid | Varchar(50) | Статус этапов задач |
| taskStageVerdictUuid | Varchar(50) | Вердикт этапа задачи |
| taskStageTemplateUuid | Varchar(50) | Шаблон этапа задачи |
| startDate | Timestamp | Дата начала |
| endDate | Timestamp | Дата завершения |
| flowOrder | Int(10) | Фильтр отображения |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица списка этапов задач [taskstagelist]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| taskTemplateUuid | Varchar(50) | Шаблон задачи |
| taskStageTemplateUuid | Varchar(50) | Шаблон этапа задачи |
| flowOrder | Int(10) | Фильтр отображения |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица списка операций этапа задач [taskstageoperationlist]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| taskStageTemplateUuid | Varchar(50) | Шаблон этапа задачи |
| operationTemplateUuid | Varchar(50) | Шаблон операции |
| flowOrder | Int(10) | Фильтр отображения |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица статуса этапов задач [taskstagestatus]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица вердиктов этапов задач [taskstageverdict]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица шаблонов этапов задач [taskstagetemplate]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| description | Text | Описание |
| image | Varchar(200) | Изображение |
| normative | Int(10) | Норматив |
| taskStageTypeUuid | Varchar(50) | Тип этапа задач |
| equipmentModelUuid | Varchar(50) | Модель оборудования |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов этапов задач [taskstagetype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| taskStageTypeUuid | Varchar(50) | Тип этапа задач |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица статус задач [taskstatus]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица шаблонов задач [tasktemplate]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| description | Text | Описание |
| image | Varchar(200) | Изображение |
| normative | Int(10) | Норматив |
| equipmentModelUuid | Varchar(50) | Модель оборудования |
| taskTypeUuid | Varchar(50) | Тип задач |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица частей ремонта шаблонов задач [tasktemplaterepairpart]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| taskTemplateUuid | Varchar(50) | Шаблон задачи |
| repairPartUuid | Varchar(50) | Часть инструмента |
| measureTypeUuid | Varchar(50) | Тип измерения |
| quantity | Int(10) | Количество |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типа задач [tasktype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица вердикта задач [taskverdict]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| icon | Varchar(100) | Иконка |
| taskTypeUuid | Varchar(50) | Тип задач |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица токенов [token]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| tagId | Varchar(128) | Метка |
| accessToken | Varchar(128) | Токен доступа |
| tokenType | Varchar(128) | Тип токена |
| expiresIn | Int(10) | Генерация токена |
| userName | Varchar(100) | Имя пользователя |
| issued | Varchar(128) | Выпушен |
| expires | Varchar(128) | Испекает |

## **Таблица инструментов [tool]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| toolTypeUuid | Varchar(100) | Тип инструмента |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица типов инструментов [tooltype]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| title | Varchar(100) | Название |
| parentUuid | Varchar(100) | Старший инструмент |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

## **Таблица пользователей [users]**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Название** |
| \_id | Int(10) | Идентификатор |
| uuid | Varchar(50) | Уникальный номер |
| name | Varchar(150) | Имя |
| login | Varchar(50) | Логин |
| pass | Varchar(50) | Пароль |
| type | Int(10) | Тип записи |
| tagId | Varchar(50) | Метка |
| active | Int(10) | Статус записи |
| whoIs | Varchar(50) | Должность |
| image | Varchar(200) | Изображение |
| contact | Varchar(100) | Контакт |
| connectionDate | Timestamp | Последнее подключение |
| createdAt | Timestamp | Дата создания |
| changedAt | Timestamp | Дата Изменения |

# **Обслуживание БД**

**Для обеспечения сохранности базы данных необходимо регулярно осуществлять резервное копирование на удаленный резервный компьютер. Резервирование на самом сервере не рекомендуется, так как теоретически может произойти его полный выход из строя или любое другое форс-мажорное событие.**

**Резервное копирование базы данных – единственный надежный способ предохранить данные от потери в результате поломки диска, сбоев в электропитании и т.д.**

**Основной причиной повреждения баз данных является аварийное завершение работы серверного компьютера, особенно отключение электропитания (необходимо иметь на сервере источник бесперебойного питания).**

**Резервное копирование рекомендуется осуществлять в момент отсутствия соединений с базой данных со стороны пользователей, т.е. завершить работу сервера опроса, а также отключить всех удаленных клиентов.**

**Для копирования /восстановления базы данных рекомендуется использовать MySQL Administrator. Для этого следует воспользоваться пунктами меню Backup и Restore и следовать указаниям мастера.**

**Рабочая база данных системы – toir-base. Следует регулярно (не реже раза в неделю) просматривать список таблиц на предмет их целостности и производить восстановление таблиц при необходимости.**

# Руководство программиста

В данном разделе собрана информация по системе ТОИРУС.

## **Структура проекта**

Проект состоит из трех главных приложений, которые сосредоточены в одной проектной директории. Название директории toirus. Наименование приложений следующие:

- api;

- backend;

- frontend.

## **Перечень и описание файлов сервера**

**Ниже приведен перечень и функциональное назначение исходных файлов проекта.**

**Api .// файл приложения**

**Backend // файл приложения**

Frontend **// файл приложения**

**Common // файл приложения**

**Console // файл общих моделей и конфигураций**

**Docs // файл для документации**

[Environments](https://github.com/mikaelwasp/toir-server/tree/master/environments) **// файл для окружения**

Vagrant **// файл для хранения лог файлов**

Vender **// файл для пакетов из composer.lock**

.bowerrc **// файл менеджера покетов bower**

.gitignore **// файл для исключения от отслеживания**

LICENSE.md **// файл лицензии**

README.md **// файл описания**

Vagrantfile **// файл развертывания виртуальной машины**

Codeception.yml **// файл для тестирования**

Composer.json **// файл зависимостей**

Composer.lock **// файл фиксированных зависимостей**

Init **// файл** **инициализации**

Init.bat **// файл инициализации**

Requirements.php **// файл сохраненных зависимостей**

Yii.bat **// файл работы командами yii**

Приложения api, backend и frontend, имеют одинаковую унифицированную структуру:

Controllers // Директория взаимодействия

Models // Директория работы с базой данной

Views // Директория представлений приложения

Web // Директория запуска приложения

# **Возможные ошибки и их устранение**

Бывают случаи, когда работа с системой прерывается. Ошибками к данной ситуации могут служить множество факторов, начиная от неработающего подключения от вашего провайдера вплоть до ошибки сервера. Далее будут изложены основные причины отсутствия возможности работы с системой.

## **Происходит ошибка подключения к серверу**

При возникновении проблемы с подключением, необходимо уточнить, какая ошибка отображается на экране вашего браузера. Основываясь на приведенных ниже данных, руководствоваться информацией, которая указана в таблице.

|  |  |
| --- | --- |
| Ошибка клиента | Наименование |
| 400 | Bad Request — сервер обнаружил в запросе клиента синтаксическую ошибку. Появился в HTTP/1.0. |
| 401 | Unauthorized — для доступа к запрашиваемому ресурсу требуется [аутентификация](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F). В заголовке ответ должен содержать поле WWW-Authenticate с перечнем условий аутентификации. Клиент может повторить запрос, включив в заголовок сообщения поле Authorization с требуемыми для аутентификации данными. |
| 402 | Payment Required — предполагается использовать в будущем. В настоящий момент не используется. Этот код предусмотрен для платных пользовательских сервисов, а не для [хостинговых](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) компаний. Имеется в виду, что эта ошибка не будет выдана хостинговым провайдером в случае просроченной оплаты его услуг. Зарезервирован, начиная с HTTP/1.1. |
| 403 | Forbidden — сервер понял запрос, но он отказывается его выполнять из-за ограничений в доступе для клиента к указанному ресурсу. Если для доступа к ресурсу требуется аутентификация средствами HTTP, то сервер вернёт ответ [401](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#401), или [407](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#407) при использовании прокси. В противном случае ограничения были заданы администратором сервера или разработчиком веб-приложения и могут быть любыми в зависимости от возможностей используемого [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). В любом случае клиенту следует сообщить причины отказа в обработке запроса. Наиболее вероятными причинами ограничения может послужить попытка доступа к системным ресурсам веб-сервера (например, файлам [.htaccess](https://ru.wikipedia.org/wiki/.htaccess) или [.htpasswd](https://ru.wikipedia.org/wiki/.htpasswd)) или к файлам, доступ к которым был закрыт с помощью конфигурационных файлов, требование аутентификации не средствами HTTP, например, для доступа к [системе управления содержимым](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D1%81%D0%BE%D0%B4%D0%B5%D1%80%D0%B6%D0%B8%D0%BC%D1%8B%D0%BC) или разделу для зарегистрированных пользователей либо сервер не удовлетворён [IP-адресом](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) клиента, например, при блокировках. Появился в HTTP/1.0. |
| 404 | Самая распространённая ошибка при пользовании Интернетом, основная причина — ошибка в написании адреса Web-страницы. Сервер понял запрос, но не нашёл соответствующего ресурса по указанному URL. Если серверу известно, что по этому адресу был документ, то ему желательно использовать код [410](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#410). Ответ 404 может использоваться вместо [403](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#403), если требуется тщательно скрыть от посторонних глаз определённые ресурсы. Появился в HTTP/1.0. |
| 405 | Method Not Allowed — указанный клиентом метод нельзя применить к текущему ресурсу. В ответе сервер должен указать доступные методы в заголовке Allow, разделив их запятой. Эту ошибку сервер должен возвращать, если метод ему известен, но он не применим именно к указанному в запросе ресурсу, если же указанный метод не применим на всём сервере, то клиенту нужно вернуть код [501](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#501) (Not Implemented). Появился в HTTP/1.1. |
| 406 | Not Acceptable — запрошенный URI не может удовлетворить переданным в заголовке характеристикам. Если метод был не HEAD, то сервер должен вернуть список допустимых характеристик для данного ресурса. Появился в HTTP/1.1. |
| 407 | Proxy Authentication Required — ответ аналогичен коду [401](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#401) за исключением того, что аутентификация производится для прокси-сервера. Механизм аналогичен идентификации на исходном сервере. Появился в HTTP/1.1. |
| 408 | Request Timeout — время ожидания сервером передачи от клиента истекло. Клиент может повторить аналогичный предыдущему запрос в любое время. Например, такая ситуация может возникнуть при загрузке на сервер объёмного файла методом [POST](https://ru.wikipedia.org/wiki/HTTP#POST) или [PUT](https://ru.wikipedia.org/wiki/HTTP#PUT). В какой-то момент передачи источник данных перестал отвечать, например, из-за повреждения [компакт-диска](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BA%D1%82-%D0%B4%D0%B8%D1%81%D0%BA) или потери связи с другим компьютером в локальной сети. Пока клиент ничего не передаёт, ожидая от него ответа, соединение с сервером держится. Через некоторое время сервер может закрыть соединение со своей стороны, чтобы дать возможность другим клиентам сделать запрос. Этот ответ не возвращается, когда клиент принудительно остановил передачу по команде пользователя или соединение прервалось по каким-то иным причинам, так как ответ уже послать невозможно. Появился в HTTP/1.1. |

Далее представлены ошибки со стороны сервера.

|  |  |
| --- | --- |
| Ошибка сервера | Наименование |
| 500 | Internal Server Error — любая внутренняя ошибка сервера, которая не входит в рамки остальных ошибок класса. Появился в HTTP/1.0. |
| 501 | Not Implemented — сервер не поддерживает возможностей, необходимых для обработки запроса. Типичный ответ для случаев, когда сервер не понимает указанный в запросе метод. Если же метод серверу известен, но он не применим к данному ресурсу, то нужно вернуть ответ [405](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#405). Появился в HTTP/1.0. |
| 502 | Bad Gateway — сервер, выступая в роли шлюза или прокси-сервера, получил недействительное ответное сообщение от вышестоящего сервера. Появился в HTTP/1.0. |
| 503 | Service Unavailable — сервер временно не имеет возможности обрабатывать запросы по техническим причинам (обслуживание, перегрузка и прочее). В поле Retry-After заголовка сервер может указать время, через которое клиенту рекомендуется повторить запрос. Хотя во время перегрузки очевидным кажется сразу разрывать соединение, эффективней может оказаться установка большого значения поля Retry-After для уменьшения частоты избыточных запросов. Появился в HTTP/1.0. |
| 504 | Gateway Timeout — сервер в роли шлюза или прокси-сервера не дождался ответа от вышестоящего сервера для завершения текущего запроса. Появился в HTTP/1.1. |
| 505 | HTTP Version Not Supported — сервер не поддерживает или отказывается поддерживать указанную в запросе версию протокола HTTP. Появился в HTTP/1.1. |
| 506 | Variant Also Negotiates — в результате ошибочной конфигурации выбранный вариант указывает сам на себя, из-за чего процесс связывания прерывается. Экспериментальное. Введено в [RFC 2295](https://tools.ietf.org/html/rfc2295) для дополнения протокола HTTP технологией [Transparent Content Negotiation](https://ru.wikipedia.org/w/index.php?title=Transparent_Content_Negotiation&action=edit&redlink=1). |
| 507 | Insufficient Storage — не хватает места для выполнения текущего запроса. Проблема может быть временной. Введено в WebDAV. |
| 509 | Bandwidth Limit Exceeded — используется при превышении веб-площадкой отведённого ей ограничения на потребление трафика. В данном случае владельцу площадки следует обратиться к своему хостинг-провайдеру. В настоящий момент данный код не описан ни в одном [RFC](https://ru.wikipedia.org/wiki/RFC) и используется только модулем «bw/limited», входящим в панель управления хостингом [cPanel](https://ru.wikipedia.org/wiki/CPanel), где и был введён. |
| 510 | Not Extended — на сервере отсутствует расширение, которое желает использовать клиент. Сервер может дополнительно передать информацию о доступных ему расширениях. Введено в [RFC 2774](https://tools.ietf.org/html/rfc2774) для дополнения протокола HTTP поддержкой расширений. |
| 511 | Network Authentication Required — этот ответ посылается не сервером, которому был предназначен запрос, а сервером-посредником — например, сервером провайдера — в случае, если клиент должен сначала авторизоваться в сети, например, ввести пароль для платной точки доступа к Интернету. Предполагается, что в теле ответа будет возвращена Web-форма авторизации или перенаправление на неё. Введено в черновике стандарта [RFC 6585](https://tools.ietf.org/html/rfc6585). |