|  |  |
| --- | --- |
| УТВЕРЖДАЮ: | УТВЕРЖДАЮ: |
| Руководитель учреждения  ГБУ СО «СОЦИ» | Генеральный директор  ООО «ИОН ДВ» |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия И. О. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Куминов А.С. |
| М.П. | М.П. |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г. | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г. |

Информационная система управления проектами

РУКОВОДСТВО АДМИНИСТРАТОРА

**Аннотация**

Данный документ является руководством администратора информационной системы «Реестр телекоммуникаций», реализованной на Платформе ION в соответствии с Договором № 64/18-к от 22 ноября 2017 года на выполнение работ по созданию информационной системы учета объектов телекоммуникаций в соответствии с условиями Технического задания.

Оглавление

[Аннотация 2](#_Toc494900115)

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 5](#_Toc494900116)

[1 Назначение и условия применения 6](#_Toc494900117)

[1.1 Область применения 6](#_Toc494900118)

[1.2 Объект автоматизации 6](#_Toc494900119)

[1.3 Уровень подготовки пользователей 6](#_Toc494900120)

[2 Окружение 8](#_Toc494900121)

[2.1 Серверное окружение 8](#_Toc494900122)

[2.2 Клиентское окружение 8](#_Toc494900123)

[3 Среда исполнения Node.js 9](#_Toc494900124)

[3.1 Установка среды исполнения Node.js 9](#_Toc494900125)

[3.2 Установка дополнительного окружения 10](#_Toc494900126)

[3.2.1 Установка Python 10](#_Toc494900127)

[3.3 Установка компилятора под операционную систему 11](#_Toc494900128)

[3.3.1 Установка node-gyp сборщика модулей с компиляцией Cи модулей для node.js 12](#_Toc494900129)

[4 СУБД 13](#_Toc494900130)

[4.1 MongoDB 13](#_Toc494900131)

[5 Развертывание и запуск платформы ION 16](#_Toc494900132)

[5.1 Конфигурирование системы, развертывание метаданных и инициализация данных 16](#_Toc494900133)

[5.2 Настройка параметров запуска 17](#_Toc494900134)

[5.3 Установка прав пользователя 18](#_Toc494900135)

[5.4 Запуск приложения 18](#_Toc494900136)

[5.5 Просмотр приложения 18](#_Toc494900137)

[6 Консоль администратора 20](#_Toc494900138)

[6.1 Работа с безопасностью 20](#_Toc494900139)

[6.1.1 Инициализация 20](#_Toc494900140)

[6.1.2 Управление ролями 22](#_Toc494900141)

[6.1.3 Доступ к ресурсу 28](#_Toc494900142)

[6.1.4 Управление пользователями 30](#_Toc494900143)

[6.2 Работа с профилированием 34](#_Toc494900144)

[6.3 Работа с журналом изменений 36](#_Toc494900145)

[6.4 Работа с резервированием 38](#_Toc494900146)

[7 Система заявок и сборки тестового контура 40](#_Toc494900147)

[7.1 Запуск через тестовой среды, через систему непрерывной интеграции gitlab 40](#_Toc494900148)

[7.2 Система заявок 43](#_Toc494900149)

[8 Аварийные ситуации 46](#_Toc494900150)

[8.1 Действия в случае несоблюдения условий технологического процесса, в том числе при длительных отказах технических средств 46](#_Toc494900151)

[8.2 Действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных. 46](#_Toc494900152)

[8.3 Действия в случае обнаружения несанкционированного вмешательства в данные 46](#_Toc494900153)

[8.4 Действия в других аварийных ситуациях 47](#_Toc494900154)

[9 Рекомендации по освоению и эксплуатации 48](#_Toc494900155)

[9.1 Общие рекомендации 48](#_Toc494900156)

[9.2 Контрольный пример 48](#_Toc494900157)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

|  |  |
| --- | --- |
| **Сокращение, определение** | **Описание** |
| GitLab | Сайт и система управления репозиториями кода для Git (распределенная система управления версиями) |
| MongoDB | Документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц |
| Node.js | Программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения |
| Атрибут | Элементы функциональной составляющей объекта Системы |
| Бекап | Резервное копирование. Процесс создания копии данных на носителе (жёстком диске, дискете и т. д.), предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения. |
| ВИС | Внешняя (ведомственная) информационная система |
| ИС | Информационная система |
| Метаданные | Данные о структуре информации которая управляет поведением системы |
| Объекты Системы | Данные в автоматизированной информационной системы |
| Платформа ION | Объектно-ориентированная среда исполнения прикладного уровня моделей организации данных и их предоставлений для пользователя |
| Система | Информационная система «Реестр телекоммуникаций» |
| СУБД | Система управления базами данных |

# Назначение и условия применения

## Область применения

Информационная система «Реестр телекоммуникаций» применяется для реализации учета объектов телекоммуникаций по населенным пунктам.

Место функционирования системы: виртуальная инфраструктура ведомства с использованием инструментов удаленного доступа.

## Уровень подготовки пользователей

Для обеспечения функционирования Системы необходим следующий персонал:

* администраторы.

Администраторы, работающие с Системой должны обладать следующими навыками:

* пользовательские навыки работы с ПК;
* пользовательские навыки работы с браузерами;
* администраторскими навыками работы с СУБД и целевой операционной системой;

# Окружение

Информационная система «Реестр телекоммуникаций», реализована на основе программного обеспечения ION. Платформа.

Требования к базовому развертыванию полностью эквиваленты требованиям ION. Платформа.

Информационная система «Реестр телекоммуникаций» может быть развернута на большинстве распространённых операционных систем, список ограничений по которым определяется ограничениями среды исполнения Node.js.

В данном руководстве будет описано развертывание системы в двух средах:

* под операционной системой Windows, путем установки отдельных компонентов. Должна использоваться поддерживаемая производителем версия операционной системы.
* под операционной системой Linux, путем установки компонентов в контейнерах Docker. Должна использоваться версия операционной системы, обеспечивающая установку docker и контейнеров. Например, версия CentOs 7.x и более новая.

## Серверное окружение

Для работы Системы необходимо следующее окружение:

* среда выполнения: Node.js не ниже 8.2.0;
* СУБД MongoDB не ниже 3.6;
* развернутая платформа ION в среде выполнения.

Дополнительно для улучшения функциональности и скорости работы требуется установить:

* сервер файлового хранилища NextCloud, последней версии;
* сервер кеширования Redis, последней версии.

## Клиентское окружение

Клиентом для решения на платформе ION является интернет браузер. Платформой обеспечивается совместимость со следующими интернет браузерами:

* Internet Explorer версии 11.0 и новее;
* Mozilla Firefox;
* Google Chrome;
* Safari;
* Opera.

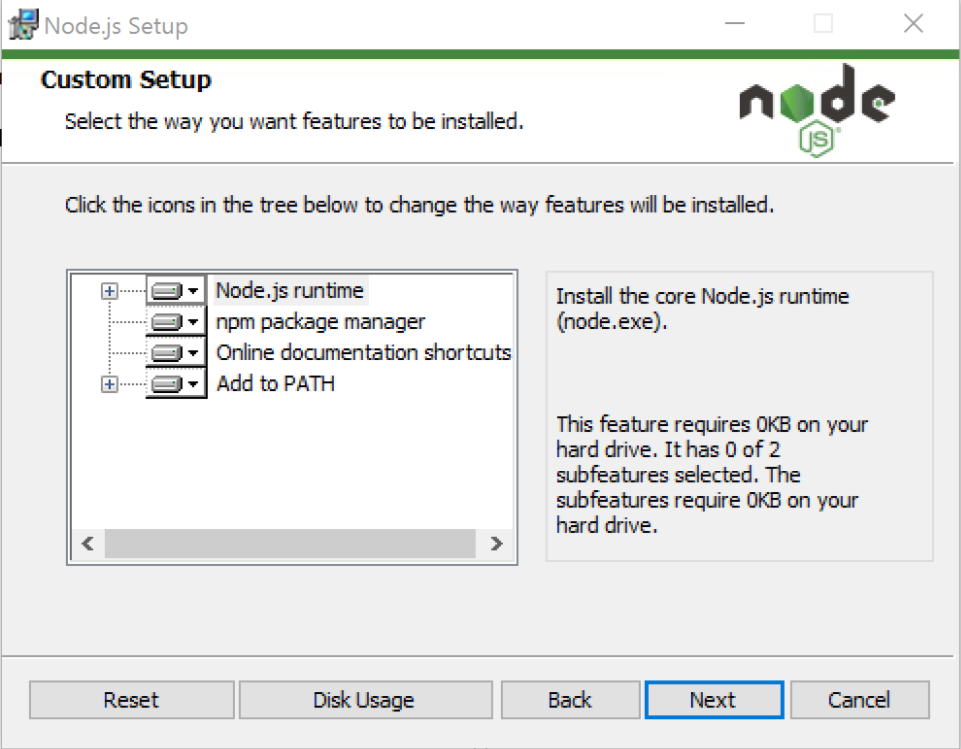
# Среда исполнения Node.js

Установка среды исполнения Node.js осуществляется под операционную среду Windows. Под операционную среду Linux обычно используются готовые контейнеры Docker содержащие и среду исполнения, и само приложение. Но в целом последовательность установки является аналогичной.

## Установка среды исполнения Node.js

Node.js - является средой, в которой осуществляется исполнение компонентов системы. Актуальную версию можно получить по адресу <https://nodejs.org>. Для работы системы необходимо установить версию старше 8.2.0, но меньше 9.0.0.

При установке нужно добавить файлы Node.js в путь – для этого в конфигурировании компонентов есть специальный пункт. Рекомендуется чтобы в путях установки Node.js отсутствовали русские буквы и пробелы.



Если node.js была установлена до этого, то необходимо выполнить в командной строке:

npm cache clean -f

Эта команда очищает кэш приложений менеджера пакетов.

## Установка дополнительного окружения

Установка окружения не является необходимой для исполнения программ на ION.платформе под Node.js собранную в виде дистрибутива. Но если предполагается самостоятельная установка зависимостей (библиотек) или их переустановка под другую операционную систему, то достаточно выполнить установку следующих компонентов:

* пакет node-gyp(<https://github.com/nodejs/node-gyp>). Для работы библиотеки под операционной системой семейства windows дополнительно необходимо установить пакет компиляции исходных кодов библиотек windows-build-tools

npm install -g node-gyp

npm install -g --production windows-build-tools

* пакет сборщика проектов Gulp (http://gulpjs.com/)

npm install -g gulp

* менджер пакетов фронтенд библиотек Bower(https://bower.io)

npm install -g bower

## Установка контейнера docker с приложением и средой исполнения

Для установки контейнера docker, нужно убедиться в доступности сервера репозитория контейнеров, например, если установка идет с локального репозитория, то сервер – docker.local.

Установка осуществляется командой:

docker pull docker.local/app/isup-pm:0-0-0-85ec5ed1

где:

docker.local/app/isup-pm:0-0-0-85ec5ed1 – адрес репозитория и имя контейнера

# СУБД

В качестве типовой СУБД в ИСУП используется MongoDB.

## MongoDB для Windows

Загрузите с официального сайта дистрибутив [MongoDB](https://www.mongodb.com/download-center#community) для Windows и установите (например, в папку C:\MongoDB). Создаем папку для логов C:\MongoDB\data\log для диагностики проблем запуска СУБД.

Для настройки MongoDB создаем файл конфигурации C:\MongoDB\mongod.conf с содержимым:

|  |
| --- |
| # mongod.conf  # for documentation of all options, see:  # http://docs.mongodb.org/manual/reference/configuration-options/  # where to write logging data.  systemLog:  destination: file  logAppend: true  path: c:\mongodb\data\log\mongo.log  # Where and how to store data.  storage:  dbPath: c:\mongodb\data\db  # journal:  # enabled: true  # engine: mmapv1  wiredTiger:  engineConfig:  cacheSizeGB: 1  # statisticsLogDelaySecs: <number>  # journalCompressor: <string>  directoryForIndexes: true  # collectionConfig:  # blockCompressor: <string>  # indexConfig:  # prefixCompression: <boolean>  # how the process runs  ##processManagement:  ## fork: true # fork and run in background  ## pidFilePath: /var/run/mongodb/mongod.pid # location of pidfile  # network interfaces  net:  port: 27017  bindIp: 127.0.0.1 # Listen to local interface only, comment to listen on all interfaces.  security:  # authorization: enabled  javascriptEnabled: false  #operationProfiling:  #replication:  #sharding:  ## Enterprise-Only Options  #auditLog:  #snmp: |

Варианты запуска MongoDB:

1. Пробный запуск вручную без сервиса

C:\MongoDB\bin\mongod.exe --config C:\MongoDB\mongod.conf

1. Установка и запуск как сервис от администратора

C:\MongoDB\bin\mongod.exe --config C:\MongoDB\mongod.conf –install

1. Остановка и удаление сервиса от администратора

C:\MongoDB\bin\mongod.exe --remove

## Контейнер MongoDB для Linux

Для установки контейнера с MongoDb, выполните команду:

docker pull mongo

После этого проверьте, что контейнер установился, а затем выполните команду получения списка установленных контейнеров:

docker images

Запуск контейнера осуществляется командой:

docker run -d --name ShareMongo -v /mongodb:/data/db -p 27017:27017 --restart=always mongo:3.6

где параметры:

--name определяет имя контейнера

-v путь монтирования папки сервера mongodb к локальной папке data/db

- p проброс порта контейнера на порт сервера

## Структура базы данных

База данных Системы организована следующим образом.

Каждому классу (сущности) системы соответствует отдельная коллекция (эквивалент таблицы в реляционных БД), которая содержит в себе объекты (эквивалент записи в реляционных БД). Запись является JSON структурой, в которой элементам сопоставляются коды атрибутов, а их значения – значениям атрибутов. Все коллекции значений начинаются с префикса наименования приложения (неймспейс).

В базе данных есть следующие специальные коллекция, являющиеся системными:

* Начинающиеся с префикса ion\_acl\_allows\_[имя объекта безопасности] – коллекции с безопасностью к каждой сущности, к которой настраивается безопасность. Объекты коллекции – являются отдельные права каждой роли или группы. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_acl\_meta – коллекция, содержащая доступы ко всем метаданным (классов).
* ion\_acl\_resources – список всех ресурсов, к которым настраивается доступ. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_acl\_roles – список всех доступов к ролям. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_acl\_users – список всех настраиваемых учетных записей пользователей. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_auth\_log – журнал событий авторизации.
* ion\_bi\_[имя отчета] – коллекция с рассчитанными значениями модуля отчетов.
* ion\_bi\_nav – коллекция с объектами навигации модуля отчетов.
* ion\_changelog – журнал изменения объектов системы.
* ion\_files – коллекция со списком файлов системы.
* ion\_geo\_layers – коллекция со слоями геомодуля.
* ion\_geo\_nav – коллекция с навигацией геомодуля.
* ion\_global\_settings – глобальные настройки приложения, созданные или обновленные в результате развертывания из файлов deploy.json всех приложений.
* ion\_meta – коллекция, содержащая все импортированные метаданные классы приложений.
* ion\_nav – коллекция, содержащая все импортированные метаданные навигации.
* ion\_notification\_recievers – коллекция, содержащая уведомления системы.
* ion\_ resources – список всех ресурсов, к которым настраивается доступ. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_ roles – список всех ролей пользователей. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_users – список всех учетных записей пользователей. Подробнее см. раздел администрирование безопасности настоящего руководства.
* ion\_user\_filters – коллекция с сохраненными пользователями фильтрами.
* ion\_view – все метаданные представлений классов.
* ion\_wf\_state – все состояния переходов бизнес-процессов, инициализированных в системе.
* ion\_workflow – все метаданные бизнес-процессов системы.
* sessions – сессии пользователей (если выбран способ хранения сессий в mongodb).

# Развертывание и запуск Системы

Конфигурирование показывается на примерах под операционную систему Windows. Разворачивание в контейнерах под операционной системой linux аналогично. Следует помнить об отличающемся разделителе папок под linux в указанных примерах.

## Конфигурирование системы, развертывание метаданных и инициализация данных

Для разворачивания самой системы, нужно подготовить папку, куда будет разархивирован архив с системой. При этом в пути к системе должны отсутствовать русские буквы, специальные символы и пробелы. Затем нужно разархивировать архив платформы в папку приложения.

Для инициализации системы нужно выполнить следующие действия, при запущенной СУБД и находясь в папке системы:

* Установка переменной окружения NODE\_PATH, в которой будет находится путь к развернутой системе – это необходимо для поиска компонентов и собственных библиотек ядра Системы. Для этого нужно запустить с правами администратора файл cmd.exe и выполнить в консоли команду:

set NODE\_PATH=%cd%

где %cd% – путь к системе.

* Инициализация метаданных системы осуществляется выполнением в командной строке и при нахождении в папке системы следующей команды

node .\bin\setup.js %CI\_PROJECT\_NAME%,

где %CI\_PROJECT\_NAME% – уточнение проекта из папки application, если мы ставим одни из них или можно запустить без параметров, чтобы инициализировать все прикладные модули. При этом уже должна быть запущена СУБД.

* Для инициализации данных нужно выполнить в командной строке команду

node .\bin\import.js --src %NODE\_PATH%\applications\%CI\_PROJECT\_NAME% --ns %CI\_PROJECT\_NAME%

где %CI\_PROJECT\_NAME% – наименование прикладного модуля.

* Для первичного входа в систему и последующей настройке прав, если права отсутствуют в инициализируемых данных, необходимо выполнить команду

node .\bin\adduser.js –name admin --pwd password

где admin – имя пользователя, а password – пароль.

Следует помнить, что NODE\_PATH всегда должна указывать на путь к ядру Системы**.** Поэтому ее лучше задать системной переменной для всех пользователей, выполнил с правами администратора команду

setx -m NODE\_PATH %NODE\_PATH%

## Настройка параметров запуска

В подпапке **config** системы нужно добавить файл **config.ini** (если он не был задан), определяющий параметры запуска (при этом удалить все примечания, идущие после знака //), либо изменить его:

|  |
| --- |
| auth.denyTop=false // - true если не нужна аутентификация на уровне платформы, и каждый модуль будет аутентифицироваться сам  db.uri=mongodb://127.0.0.1:27017/db // URI подключения к БД  db.user=username // пользователь БД – для ускорения работы, отключен по умолчанию  db.pwd=password // пароль пользователя БД  // Попытки запустить систему на этих портах  server.port[]=8888  server.port[]=8889  server.port[]=8890  module.default=registry // модуль по умолчанию  // Настройки файлового хранилища  fs.storageRoot=./files // корневая директория хранилища, куда будут сохраняться файлы  fs.urlBase=/files // база для публикации файлов в http (база URL файлов) |

## Установка прав пользователя

Для того, чтобы иметь возможность войти в систему и добавлять других пользователей – нужно завести пользователя администратора. Это осуществляется следующей командой

node %NODE\_PATH%\bin\adduser.js --name admin --pwd Admin\_Password

После этого необходимо добавить администраторские права этому пользователю.

node %NODE\_PATH%\bin\acl.js --u admin@local --role admin --p full

## Запуск приложения

Проверьте переменную NODE\_PATH командой **set**, которая должна содержать путь к папке с системой и запустите приложение командой

node bin\www

Если вы запускаете кластер, то нужно задать в переменные окружения количество инстанций и выполнить команду

node bin\cluster

Примечание: запуск контейнера docker выполняется нижеследующей командой (при этом если контейнер не установлен, он будет предварительно получен из репозитория)

docker run -d --name telecom\_production -e NODE\_PATH=/var/www -e PROD\_DOMAIN\_CLOUD="http:\/\/10.12.1.143" -e PROD\_CLOUD\_USER=admin -e PROD\_CLOUD\_PASS=admin -v /var/www/files:/var/files -p 80:8888 --add-host=pm-mongodb.local:10.12.1.141 --restart=always -e KILL\_MEM -e NODE\_ENV=production docker.local/app/telecom:0-0-0-85ec5ed1 bash /opt/production.sh

где:

--name telecom\_production – устанавливаемое имя контейнера

-e NODE\_PATH=/var/www -e PROD\_DOMAIN\_CLOUD="http:\/\/10.12.1.143" -e PROD\_CLOUD\_USER=admin -e PROD\_CLOUD\_PASS=admin – установка переменных окружения, которые определяют значения обязательной переменной окружения NODE\_PATH, место нахождения ядра Системы и переменных окружения доступа к файловому хранилищу

-v /var/www/files:/var/files – монтирование папки с файлами (хранящимися не в файловом хранилище

-p 80:8888 – проброс порта запуска приложения в контейнере 8888 на локальный порт 80

--add-host=pm-mongodb.local:10.12.1.141 – адрес хоста с базой данных

--restart=always – автоматический перезапуск докер контейнера

-e KILL\_MEM – ограничение объема памяти, при превышении которого система перезапускается

-e NODE\_ENV=production – продуктивные режим работы контейнера

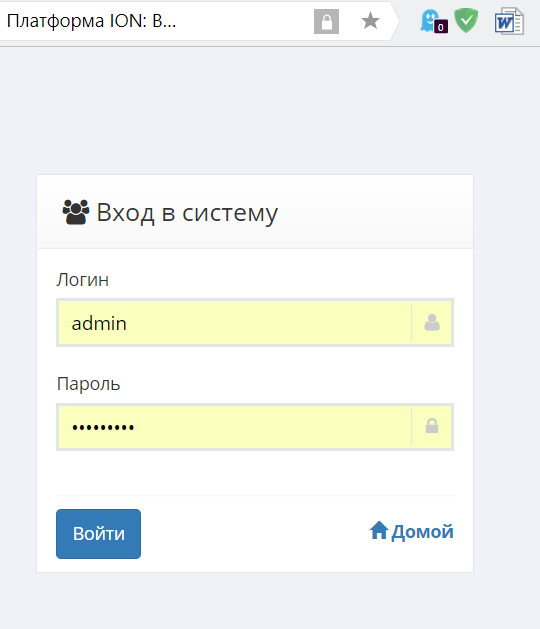
docker.local/app/telecom:0-0-0-85ec5ed1 – адрес репозитория и имя контейнера

bash /opt/production.sh – запуск приложения через скрипт

## Просмотр приложения

Для того чтобы перейти к приложению, нужно открыть страницу адрес системы, например при локальном доступе <http://localhost:8888> – в зависимости от конфигурации портов указанных в подразделе 5.2 и войти под учетными данными заданными в соответствии с подразделом 5.3.

Если вы запустили докер с пробросом портов, то открыть нужно сервер с указанным портом (зачастую порт 80). Далее везде будет использоваться локальный адрес с портом 8888.



# Консоль администратора

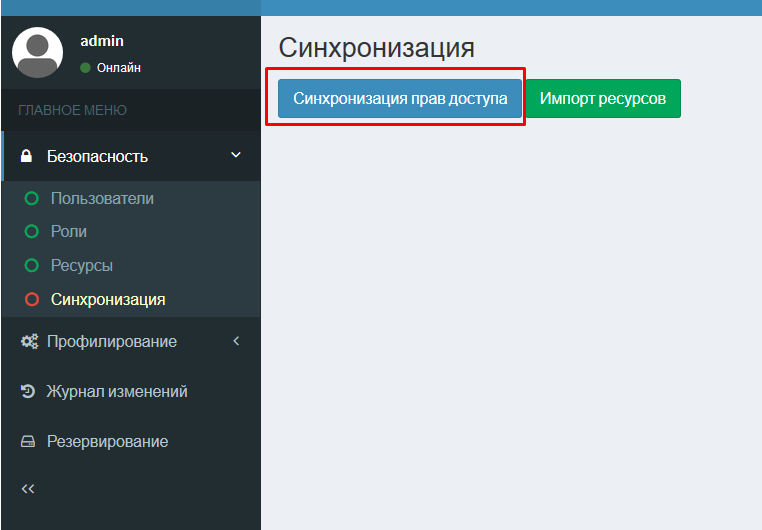
Для начала работы с консолью администратора, нужно перейти по адресу системы, добавив название модуля /ionmodules, например при локальном доступе <http://localhost:8888/ionadmin>.

## Работа с безопасностью

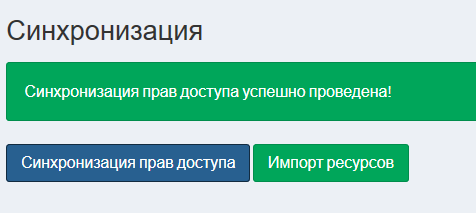
### Инициализация

При первичной настройке безопасности надо выполнить следующее:

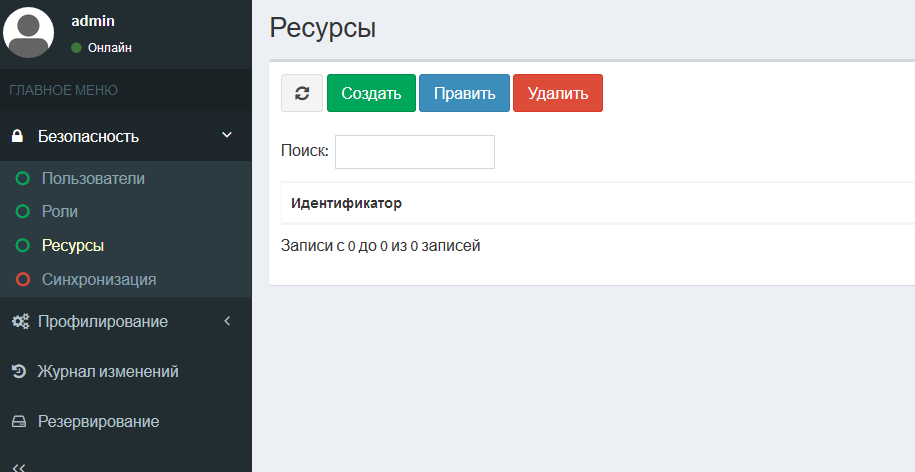
1) Синхронизировать права, чтобы права, заведенные через утилиту acl, появились в административной консоли. Для этого на странице /ionadmin/security/sync жмем кнопку "Синхронизация прав доступа".



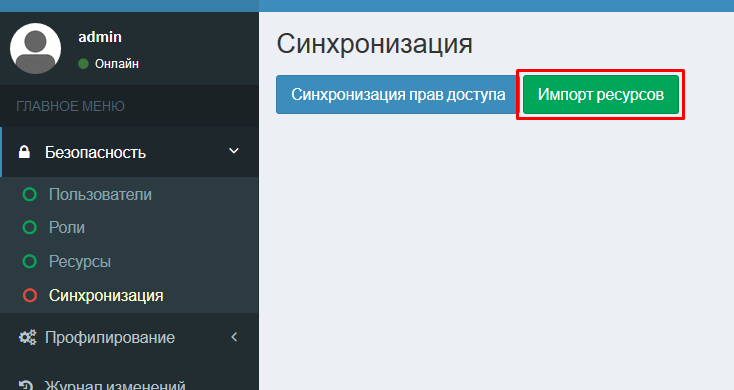
По завершении будет выведено сообщение "Синхронизация прав доступа успешно проведена!".



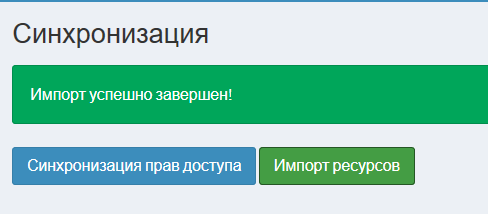
2) Сделать импорт ресурсов. Если на странице /ionadmin/security/resource нет объектов или их действительно мало, тогда надо выполнить импорт ресурсов.



Выполняется на странице /ionadmin/security/sync по кнопке "Импорт ресурсов".



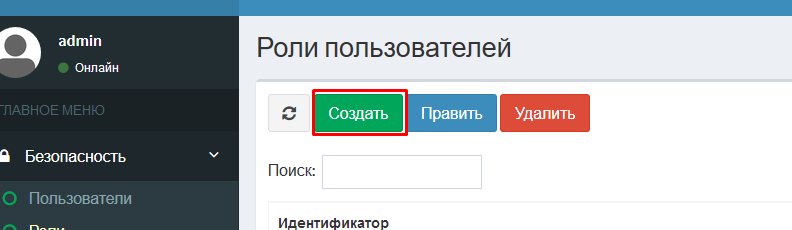
По завершении будет выведено сообщение "Импорт успешно завершен!":



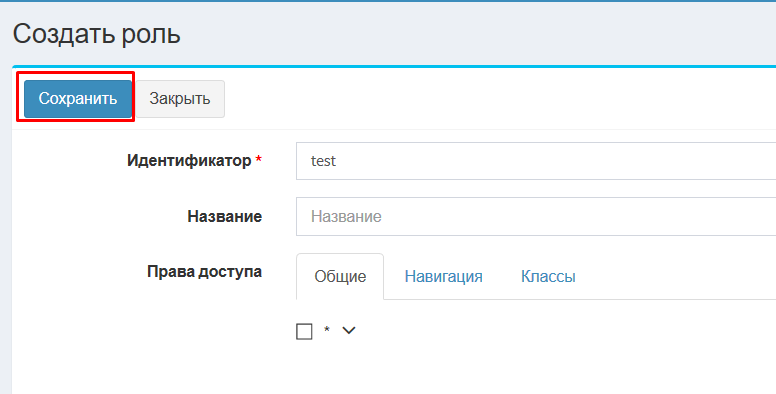
### Управление ролями

На странице /ionadmin/security/role можно создавать, редактировать или удалять роли:

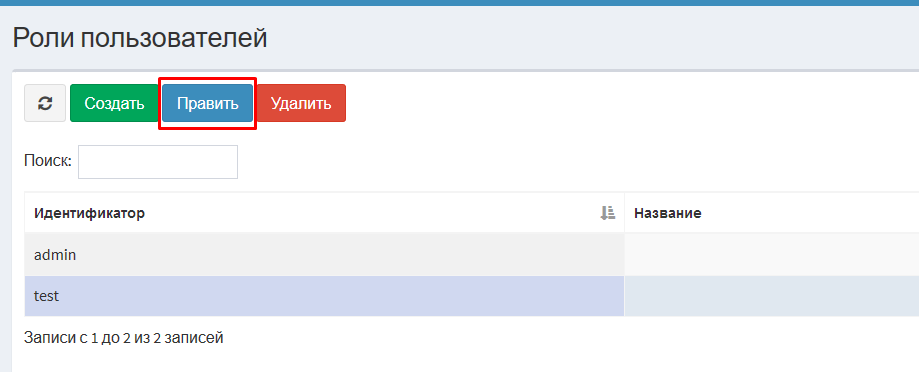
1) Создание роли. Для создания роли на странице /ionadmin/security/role нажимаем кнопку "Создать"



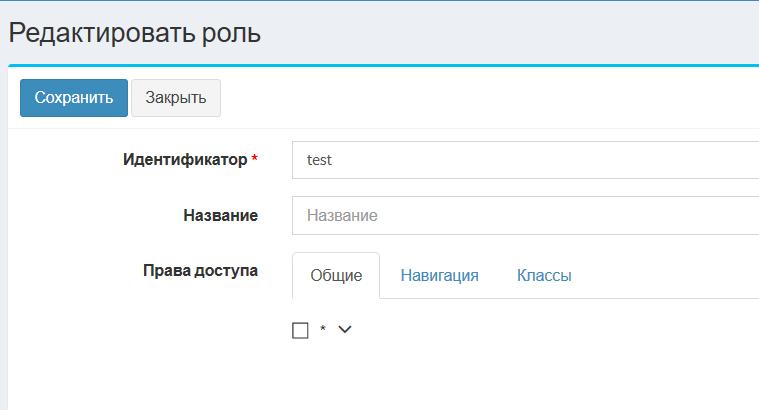
Происходит переход на новую страницу, где надо указать идентификатор роли на английском языке. Нажимаем на кнопку "Сохранить" для подтверждения создания указанной роли



2) Редактирование роли. Для редактирования роли на странице /ionadmin/security/role выбираем необходимую роль и нажимаем кнопку "Править"



Происходит переход на новую страницу, где на форме отображаются следующие поля:

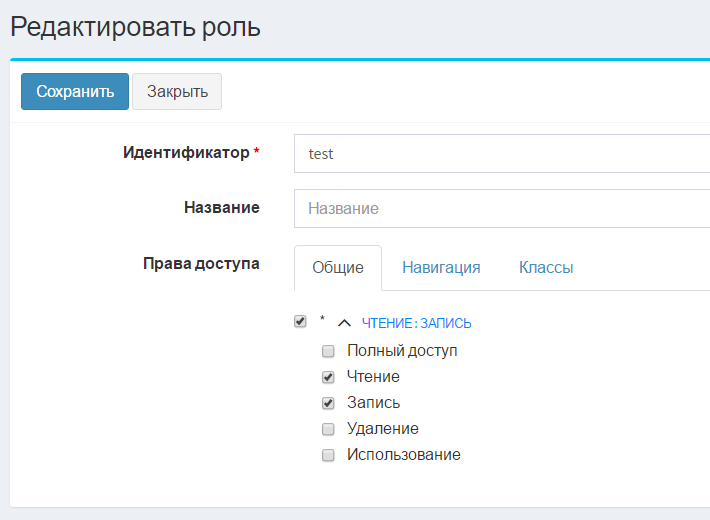


Идентификатор - выражение на английском языке для присваивания уникального имени роли. При ее изменении могут слететь права у пользователей, которые ранее привязали данную роль со старым идентификатором. В этом случае необходимо каждому пользователю привязать роль заново.

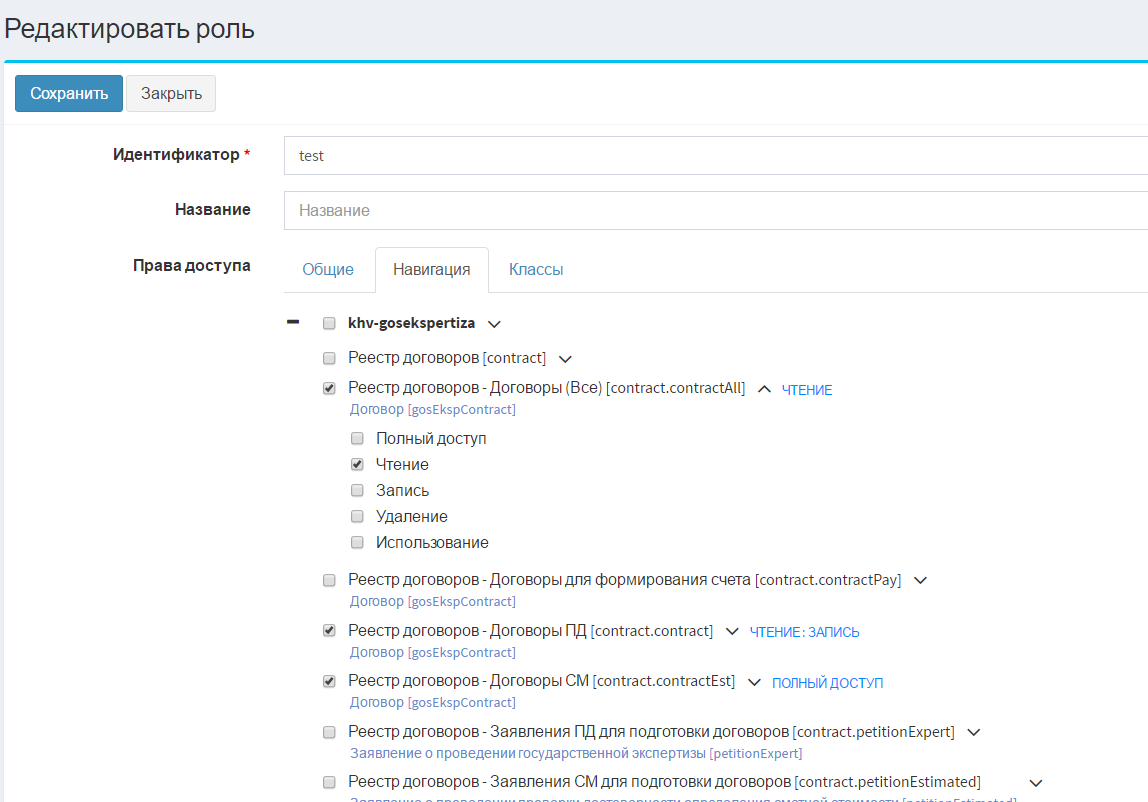
Название - осмысленное название роли, может содержать выражения на русском языке.

Права доступа - вкладки для раздачи прав:

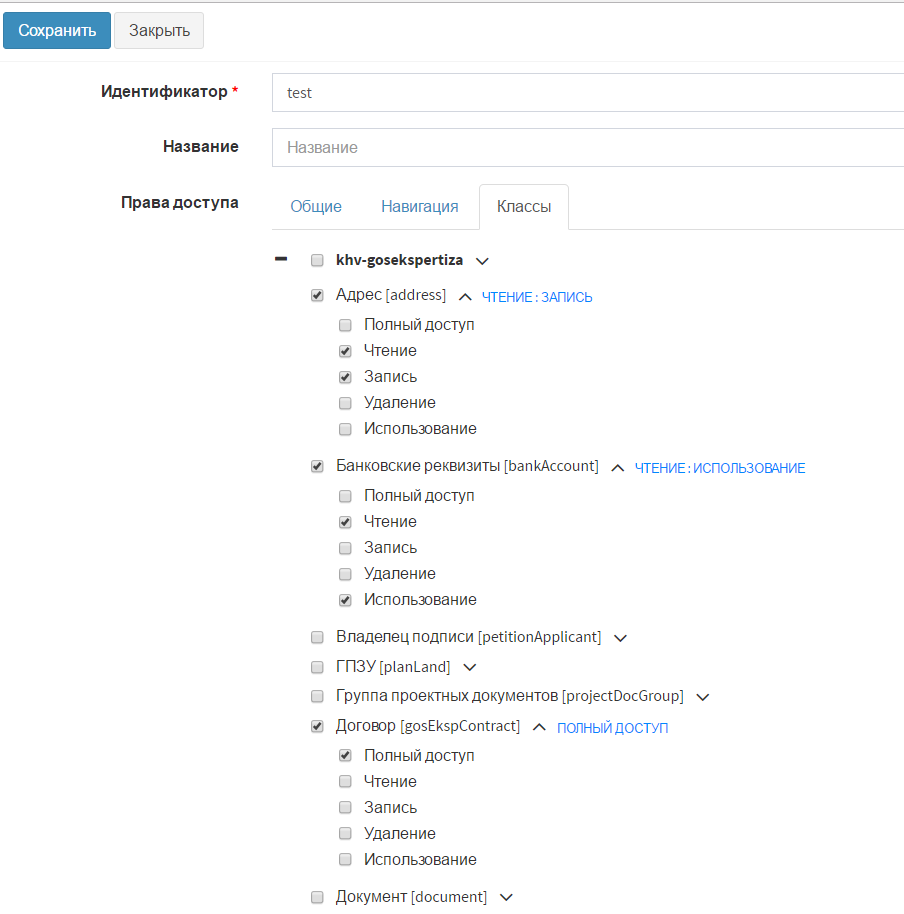
* Общие - пока используется для раздачи роли доступа ко всем ресурсам (\* - все ресурсы)



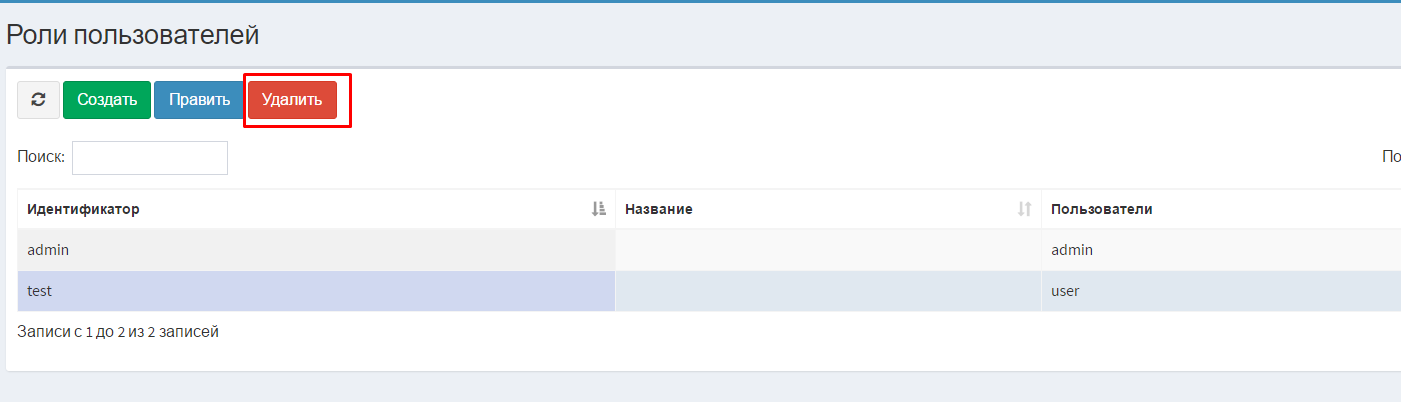
* Навигация - используется для раздачи роли доступа к меню модуля регистра. Сначала отображается системное наименование проекта, у которого есть знак плюса для отображения внутренних и импортированных ресурсов. Раздаются в данном пункте только права на чтение меню.



* Классы - используется для раздачи роли доступа к классам метаданных. Сначала отображается системное наименование проекта, у которого есть знак плюса для отображения внутренних и импортированных ресурсов. У этих ресурсов можно задать отдельные права по каждому ресурсу.



3) Удаление роли. Для удаления роли на странице /ionadmin/security/role выбираем необходимую роль и нажимаем кнопку "Удалить"



Подтверждаем удаление роли:

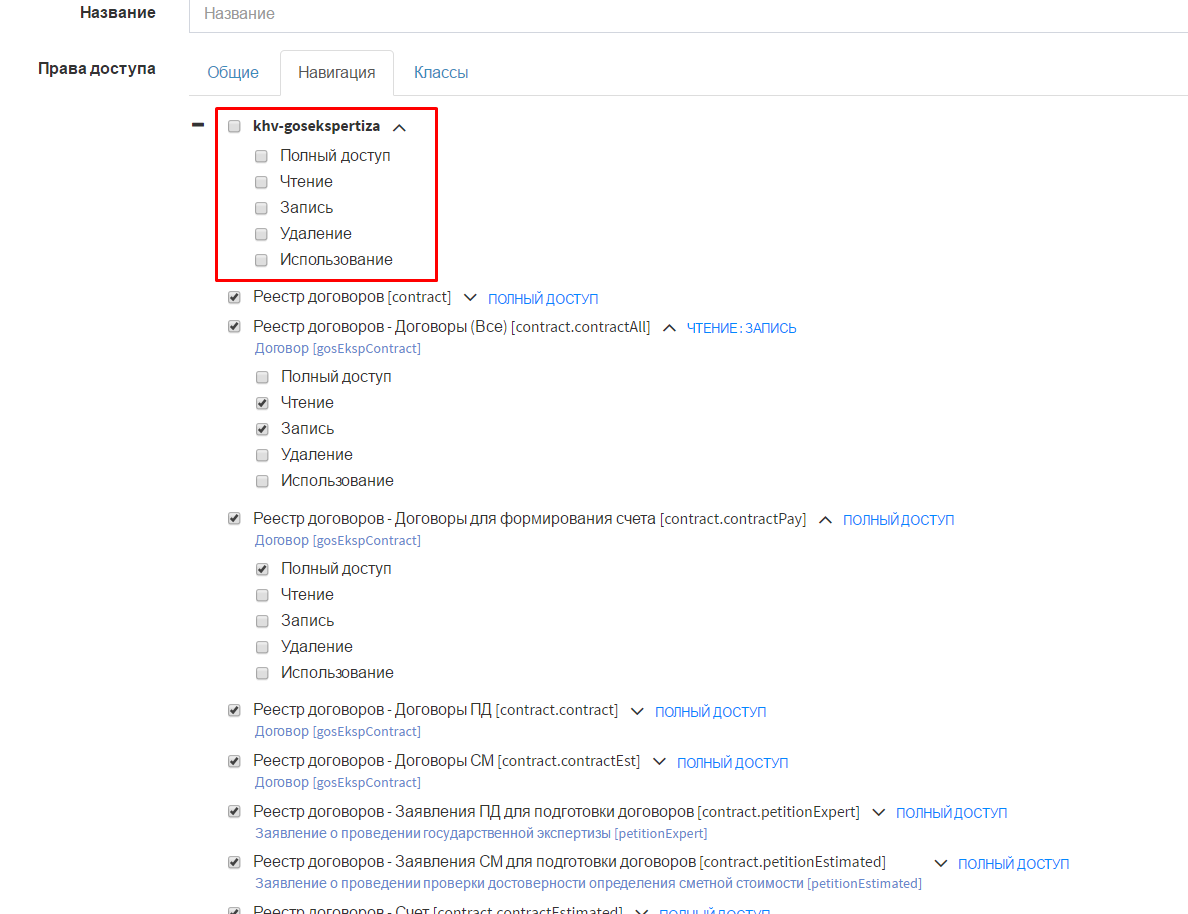


### Доступ к ресурсу

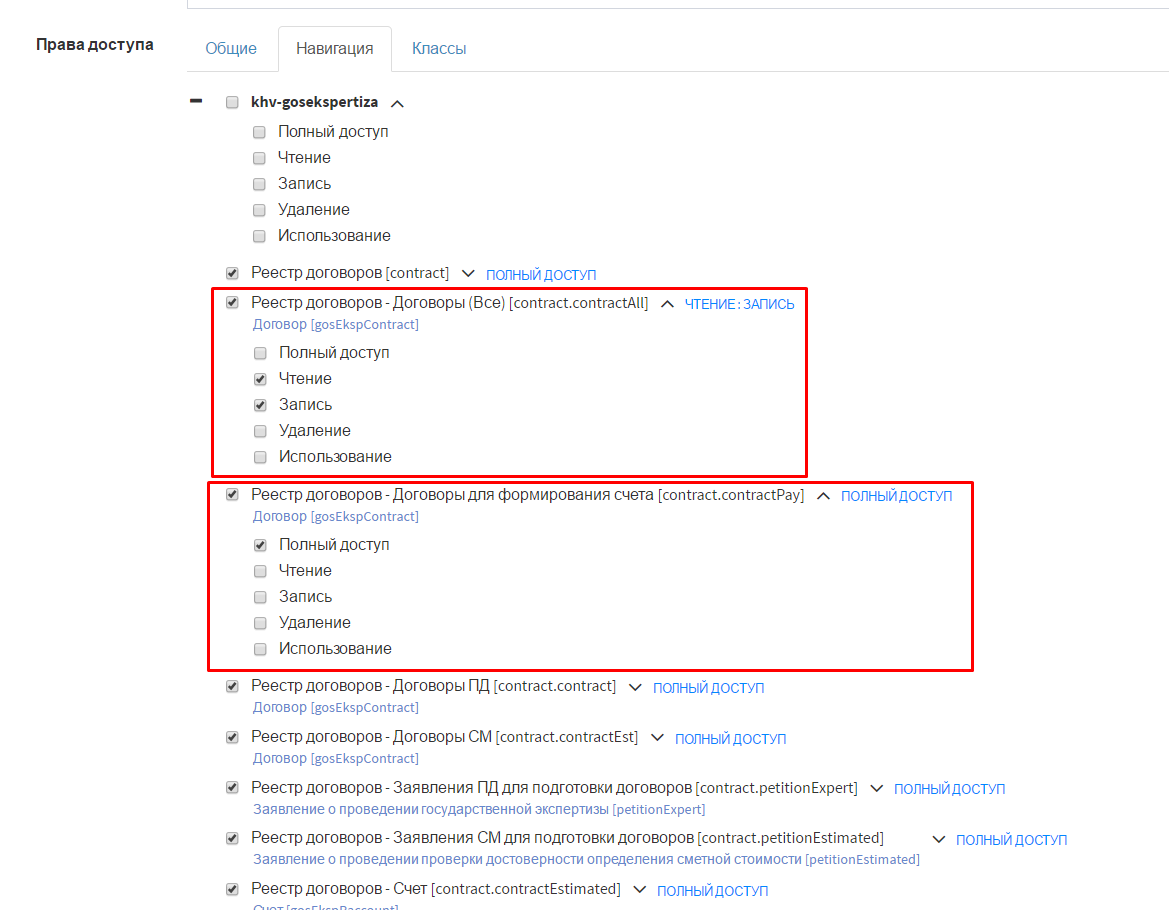
При управлении ролями в правах доступа предоставляются следующие доступы к каждому ресурсу:

|  |  |
| --- | --- |
| Доступ | Описание доступа |
| Полный доступ | Включает в себя все остальные доступы. Нельзя выбрать полный доступ и еще дополнительный доступ на чтение, запись, удаление или использование. Для навигации предоставляется доступ только на чтение ресурса. |
| Чтение | Возможность чтение объектов ресурса |
| Запись | Возможность редактирования объектов ресурса, не применяется для навигации |
| Удаление | Возможность удаления объектов ресурса, не применяется для навигации |
| Использование | Возможность создания объектов ресурса, не применяется для навигации |

Доступ может назначаться всей группе ресурсов, если их отобразить так, как показано на рисунке:



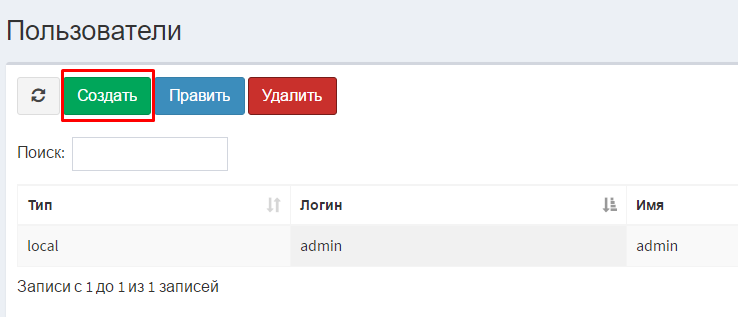
Или отдельными галками по каждому ресурсу и доступу в нем:



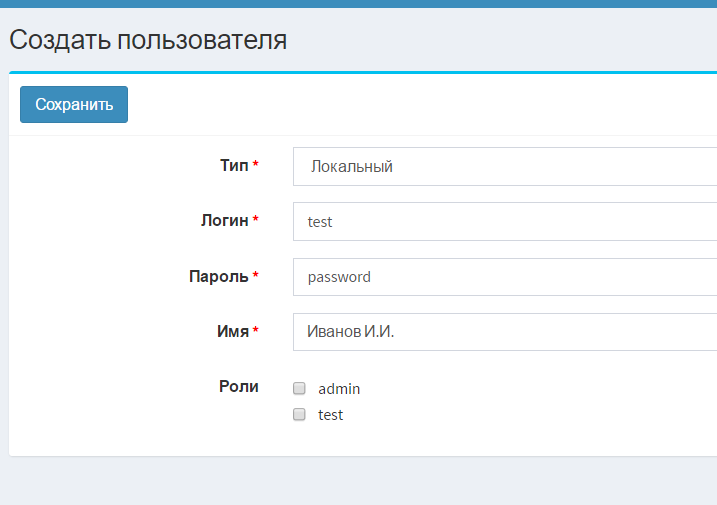
### Управление пользователями

На странице /ionadmin/security/user можно создавать, редактировать или удалять пользователей:

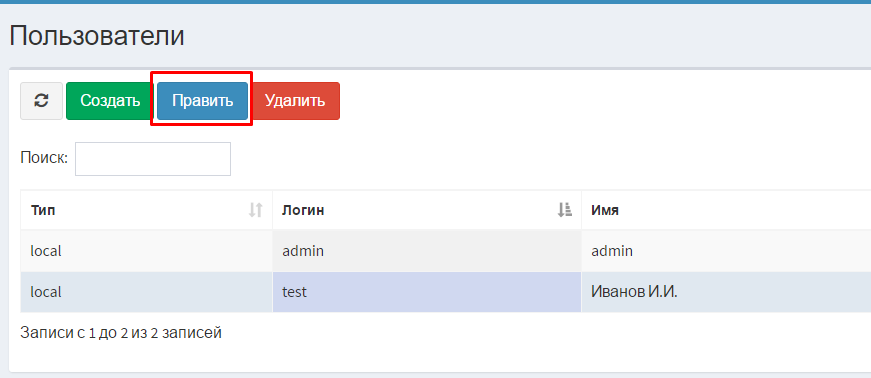
1) Создание пользователя. Для создания пользователя на странице /ionadmin/security/user нажимаем кнопку "Создать"



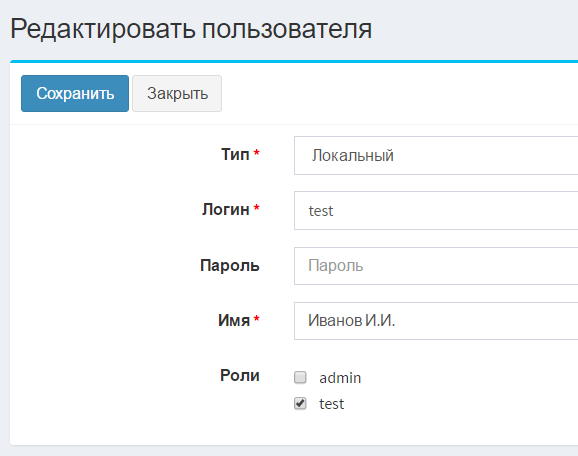
Происходит переход на новую страницу, где надо указать логин пользователя на английском языке, пароль, описание в имени. Нажимаем на кнопку "Сохранить" для подтверждения создания указанного пользователя



2) Редактирование пользователя. Для редактирования пользователя на странице /ionadmin/security/user выбираем необходимого пользователя и нажимаем кнопку "Править"



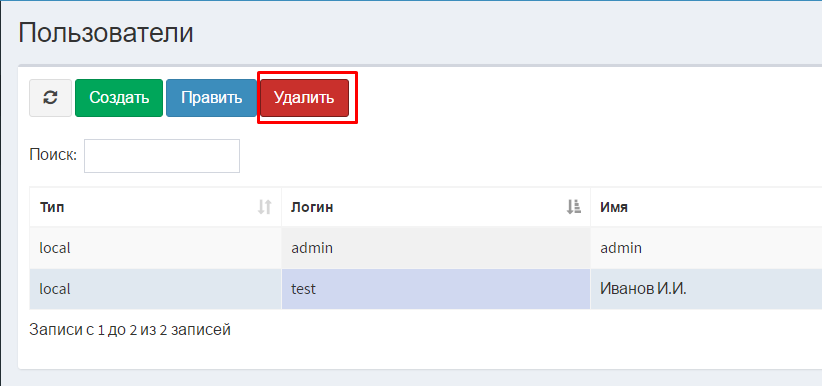
Происходит переход на новую страницу, где на форме отображаются следующие поля:



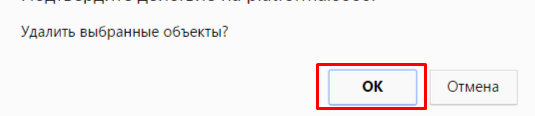
На форме указано:

* Тип - тип учетной записи пользователя, на данный момент доступен тип «локальный».
* Логин - идентификатор пользователя на английском языке.
* Пароль - пароль пользователя.
* Имя - осмысленное название пользователя, может содержать выражения на русском языке, например, Фамилия И.О.
* Роли - список ролей пользователя. Если проставлена галка у роли, то роль привязана к пользователю.

3) Удаление пользователя. Для удаления пользователя на странице /ionadmin/security/user выбираем необходимого пользователя и нажимаем кнопку "Удалить"

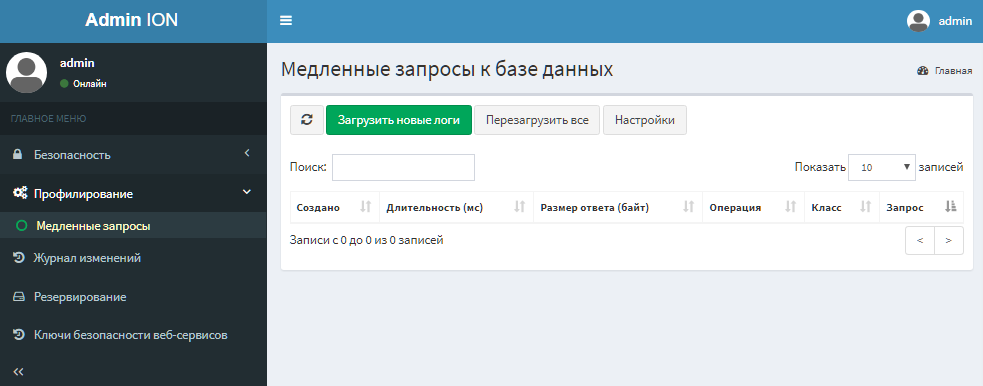


Подтверждаем удаление пользователя:



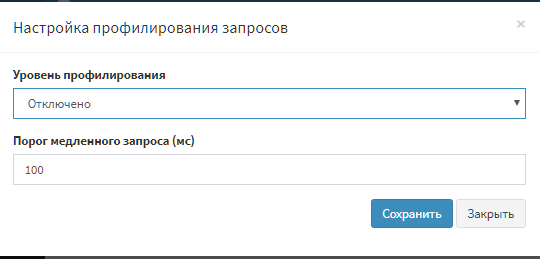
## Работа с профилированием

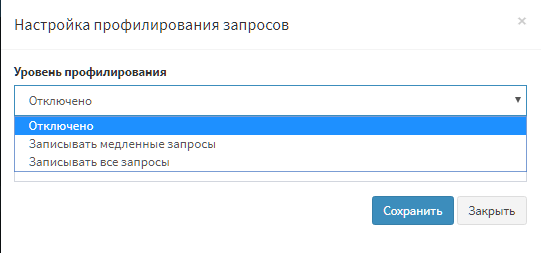
Профилирование представляет собой процесс сбора характеристик работы программы, таких как время выполнения отдельных фрагментов. В данном случае отражается информация по медленным запросам к базе данных системы:



Для просмотра информации по запросам необходимо «загрузить новые логи», после этого они отобразятся в таблице ниже.

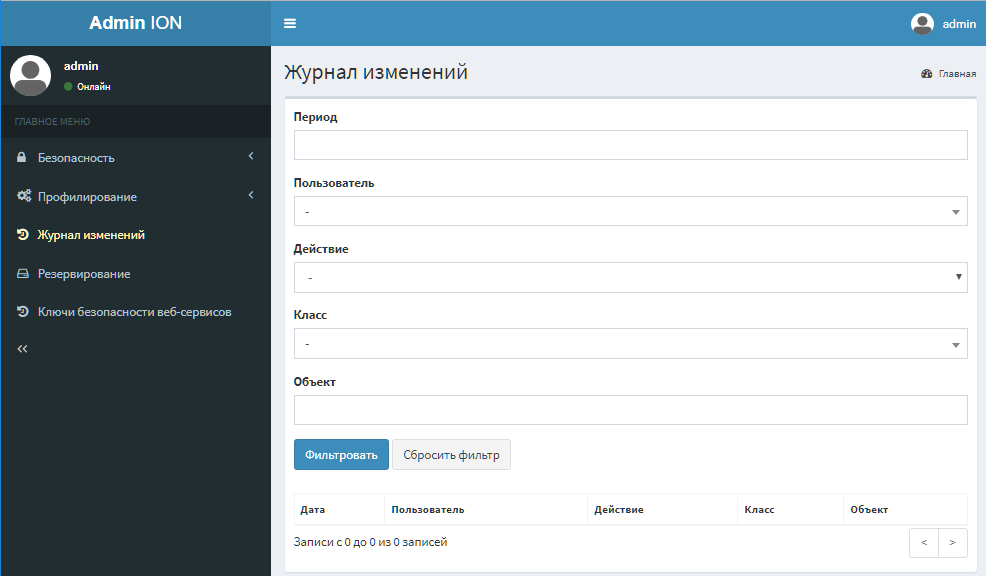
Настройка профилирования запросов выполняется при переходе по кнопке «Настройка» и выбору соответствующего уровня профилирования и величины порога медленного запроса.





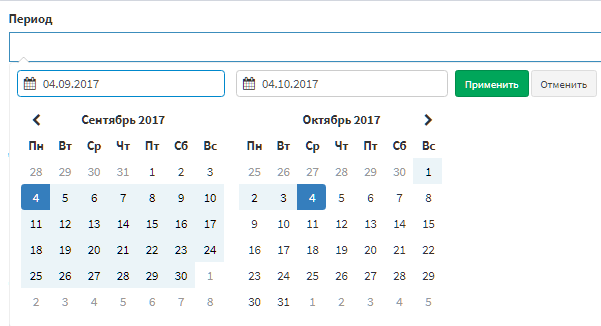
## Работа с журналом изменений

Изменения в журнал записываются в соответствии с редактированием компонентов системы. Для просмотра изменений достаточно заполнить поля журнала и выполнить действие «Фильтровать», после чего в таблице отобразятся результаты фильтра в соответствии с заполненными значениями.

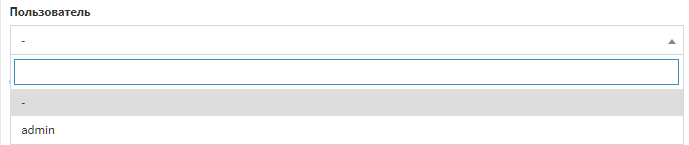


Обозначение полей фильтра:

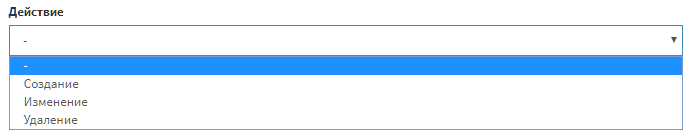
* Период – при выборе поля «Период» отобразится календарь для выбора начала и окончания периода, в рамках которого просматриваются изменения над объектами.



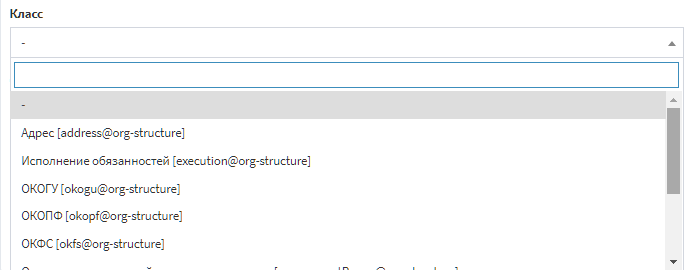
* Пользователь – выбор пользователя, которым были выполнены интересующие изменения.



* Действие – выбор действия, совершаемого в рамках просматриваемых изменений.



* Класс – наименование класса, в рамках которого были выполнены изменения.

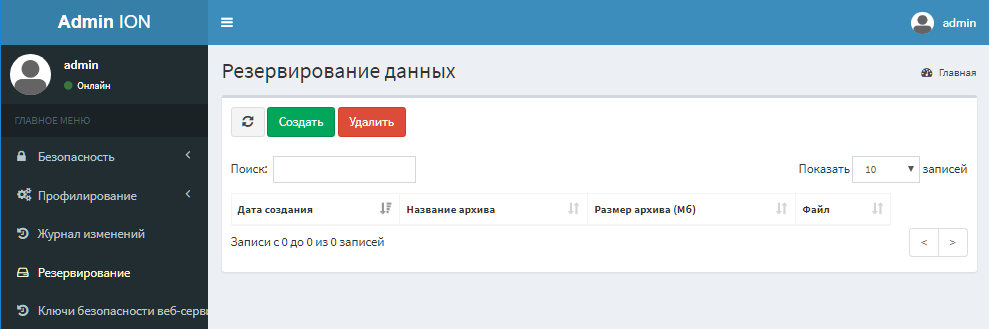


* Объект – выбор объекта, над которым были выполнены указанные выше действия.

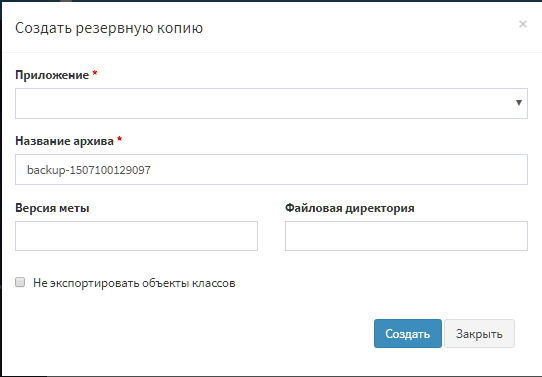
## Работа с резервированием

Резервирование данных системы представляет собой процесс создания копии объектов системы с целью обеспечения целостности данных при сбоях в работе системы, в рамках которых может произойти потеря оригинального состава данных.

Для создания резервной копии данных необходимо перейти на вкладку «Резервирование» и выбрать действие «Создать».



Далее откроется окно выбора параметров создаваемой копии данных системы.



Описание параметров создания резервной копии системы:

* Приложение – выбор наименования системы.
* Название архива – возможность изменить сетевое наименование архива на более читабельное и понятное.
* Версия меты – номер версии меты.
* Файловая директория – выбор пути для сохранения объектов резервирования.
* Экспорт объектов класса – выбор необходимости выполнять копию объектов класса системы. Объекты класса – данные, хранящиеся в системе, описывающие назначение системы в целом.

# Система заявок и сборки тестового контура

Для работы с заявками в поддержке используется система управления репозиториями кода GitLab, к которой подключен модуль непрерывной интеграции.

Ключевые возможности GitLab:

* Организация публичных и приватных репозиториев.
* Управление правами, группами.
* Импорт проектов, в том числе с GitHub.
* Вики.
* API.
* Доска идей и задач.
* Лейблы, вехи, шаблоны, поиск.
* Комментирование, объединение.
* Отслеживание изменений и прогресса.
* Отслеживание времени.
* Встроенная система непрерывной интеграции

Система поддерживается на ресурсах компании Исполнителя, или может быть развернута у Заказчика.

## Запуск через тестовую среду, через систему непрерывной интеграции gitlab

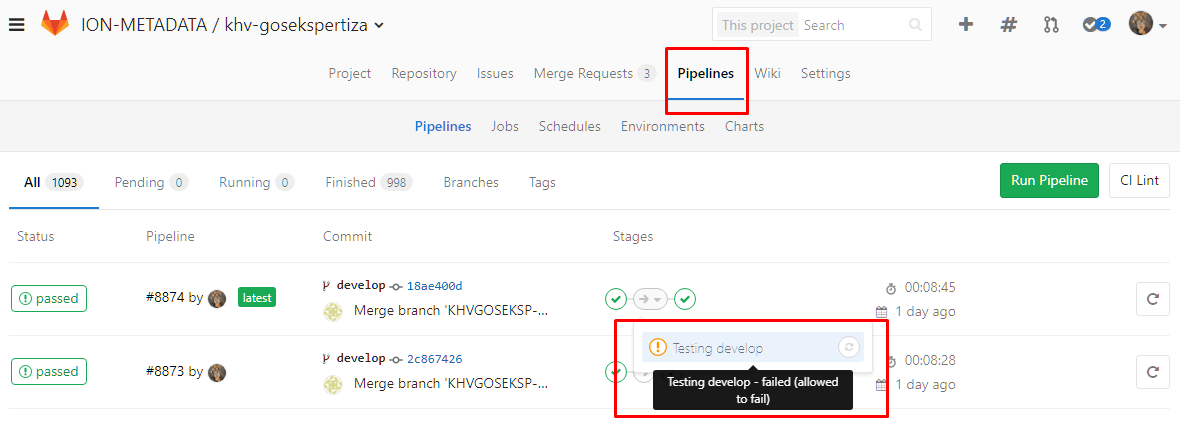
Доступ к репозиторию gitlab возможен по адресу http://git.iondv.ru. Репозитории содержат две главные ветки в хранилище (origin) с бесконечным жизненным циклом:

* master
* develop

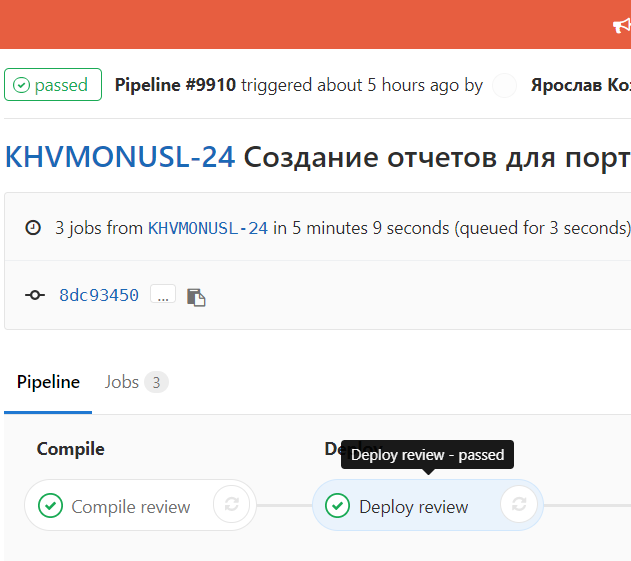
Ветвь origin/master является главной ветки, HEAD в которой, всегда указывает на production ready состояние.

Ветвь origin/develop рассматривается как главная ветка, в которой HEAD отражает состояние исходного кода, на момент последних изменений, подготовленных к следующему релизу – в рамках модели тестирования предполагается, что на этой ветке запускается проверка тестовой среды.

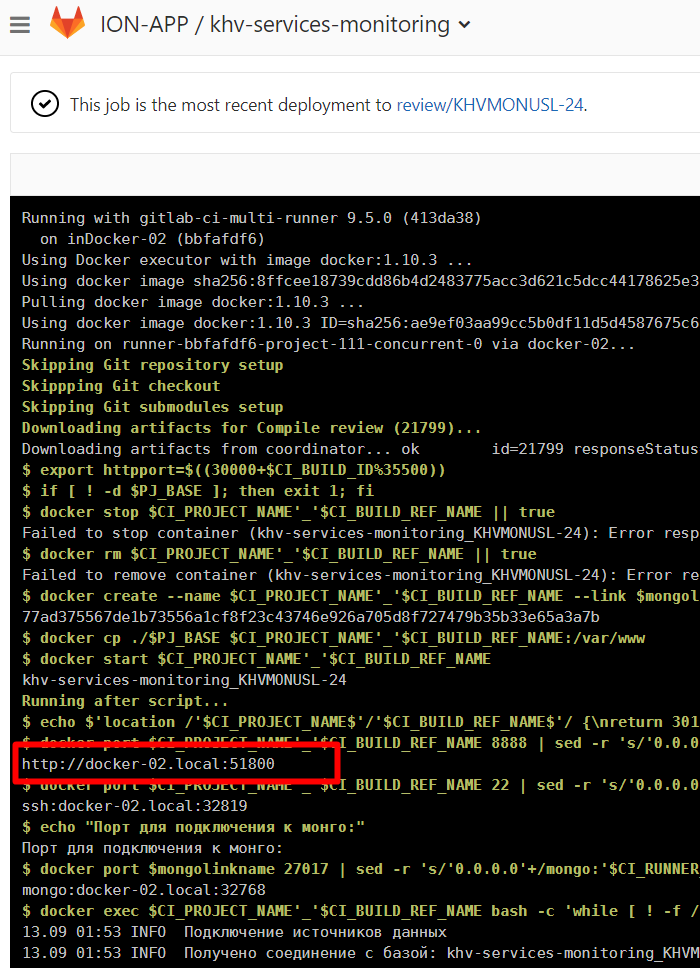
После каждого слияния веток проекта в ветки develop и master, автоматически запускается скрипт сборки проекта. Результаты сборки можно увидеть в разделе «Pipelines»:



В колонке Status можно увидеть результат сборки. Для того чтобы увидеть журналы сборки, нужно перейти в описание выполненной задачи кликнув по статусу «Deploy»

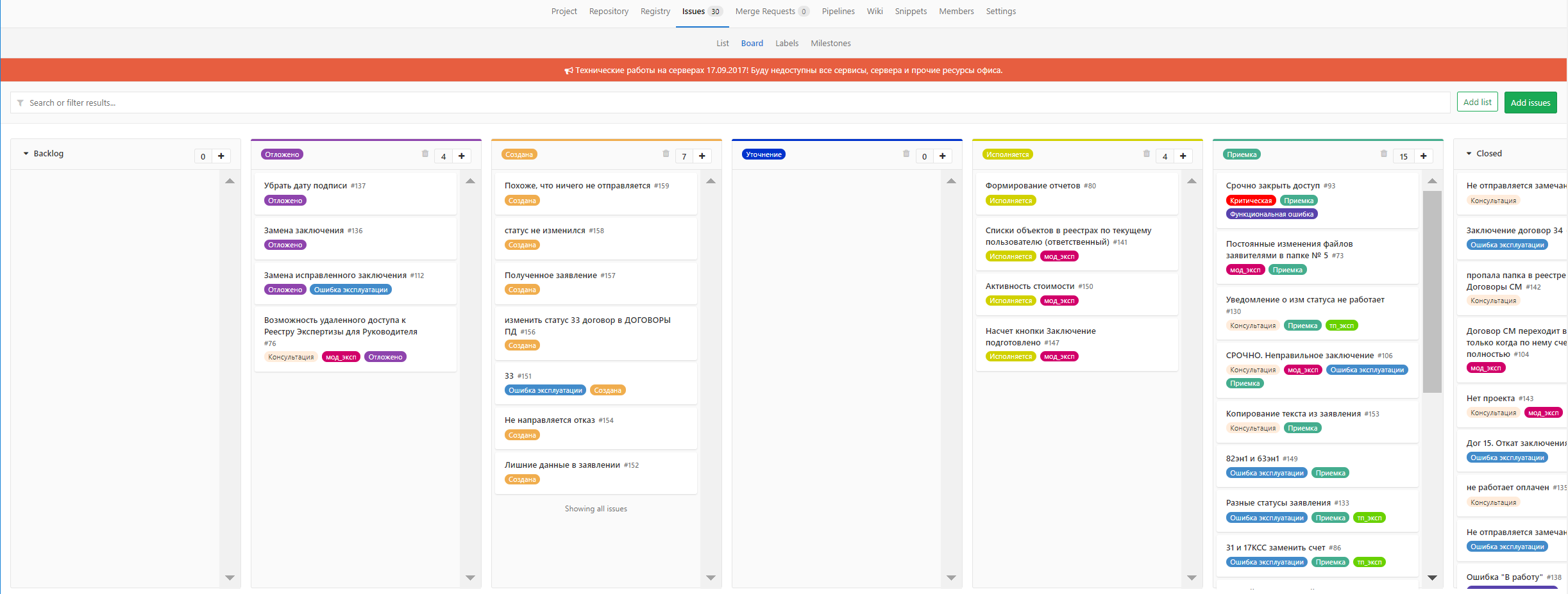


В журнале находится адрес расположения тестовой среды, на которой можно посмотреть изменения в системе. Для того чтобы посмотреть журнал, нужно нажать на статус.



## Система заявок

Для работы с заявками, необходимо перейти в пункт «Issues» проекта и далее, для удобства на доску «Board», в которой все заявки расположены по статусам.



Все заявки делятся по следующим статусам:

* backlog – только созданные заявки, ещё не сортированные или которые находятся в статусе перспективных и обсуждаются.
* Отложено – заявки, по которым требуются дополнительные действия, по которым не согласован статус необходимости взять в работу.
* Создана – задачи которые приняты как необходимые для выполнения, но ещё не взятые в работу.
* Уточнение – заявки, по которым требуется информация от заказчика и которые не могут быть без этого выполнены.
* Консультация - Заявка, которая требует консультации, разъяснения специалиста ТП, не связана с ошибками и не требует создания задачи.
* Ошибка - Проблема, препятствующая функционированию продукта, требует немедленного реагирования.
* Исполняется – заявки, по которым проводится активная работа.
* Приемка – выполненные заявки, по которым нужна проверка заказчика.
* Closed – выполненные или закрытые заявки, работы по ним больше не ведутся.

# Аварийные ситуации

## Действия в случае несоблюдения условий технологического процесса, в том числе при длительных отказах технических средств

При несоблюдении условий технологического процесса, в том числе при длительных отказах технических средств необходимо проверить работоспособность системы. В случае если система не работоспособна, общесистемному администратору необходимо выявить в каком звене происходит отказ общей системы. Ошибка может быть на уровне приложения, ошибка в целостности данных на сервере базы данных, сетевых настроек или работоспособности технических средств.

## Действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных.

Для надежной работы системы необходимо производить регулярное копирование системы на другие носители информации. Так же необходимо производить резервное копирование (бекап) базы данных и файлов, сохраненных в системе. Хранение резервных копий всех данных системы может осуществляться, как на магнитных носителях технического средства, непосредственно на котором установлена система, так и на других носителях, в том числе внешних файловых хранилищах. Если резервной копии данных нет в наличии, необходимо обратиться к общесистемному администратору Системы.

## Действия в случае обнаружения несанкционированного вмешательства в данные

При обнаружении несанкционированного вмешательства в данные необходимо проверить данные на наличие ошибок. Если при проверке данные системы не повреждены злоумышленником, то необходимо выявить шаг, на котором было произведено несанкционированное вмешательство в данные, и устранить возможность будущего проникновения. Общесистемному администратору рекомендуется тщательно проверить настройки подсистемы безопасности. В случае если данные повреждены, необходимо их восстановить из резервной копии системы.

## Действия в других аварийных ситуациях

Если вы получили какое-либо сообщение о другом сбое системы, обратитесь к общесистемному администратору.

# Рекомендации по освоению и эксплуатации

## Общие рекомендации

Данных, приведенных в настоящем Руководстве, достаточно для эксплуатации и запуска Системы. Освоение Системы осуществляется естественным способом в процессе ее эксплуатации.

Персоналу, ответственному за эксплуатацию, поддержку и сопровождение Системы следует также ознакомиться с функциональным назначением, характеристиками и условиями эксплуатации, приведенными в руководствах производителей указанных устройств и программного обеспечения.

## Контрольный пример

Для проверки работоспособности Системы рекомендуется выполнить следующую последовательность действий:

* создать учетную запись пользователя;
* добавить пользователю роль;
* настроить роль пользователя на просмотр объектов;
* осуществить вход в Систему под учетной записью созданного пользователя и протестировать результаты действий.