

УПРАВЛЕНИЕ ПОДКЛЮЧЕНИЕМ УСЛУГ

Руководство по эксплуатации

Версия 1.5.0

Настоящая документация может быть использована только для поддержки работоспособности продуктов, установленных на основании договора с АО «Нэксайн». Документация может быть передана на основании договора, по которому производится (производилась или будет производиться) установка продуктов, или явно выраженного согласия АО «Нэксайн» на использование данной документации. Если данный экземпляр документации попал к вам каким-либо иным образом, пожалуйста, сообщите об этом в АО «Нэксайн» по адресу, приведенному ниже.

Все примеры, приведенные в документации (в том числе примеры отчетов и экранных форм), составлены на основании тестовой базы АО «Нэксайн». Любое совпадение имен, фамилий, названий компаний, банковских реквизитов и другой информации с реальными данными является случайным.

Все встречающиеся в тексте торговые знаки и зарегистрированные торговые знаки являются собственностью их владельцев и использованы исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили в документации ошибки или опечатки или предполагаете их наличие, пожалуйста, сообщите об этом в АО «Нэксайн».

Все имущественные авторские права сохраняются за АО «Нэксайн» в соответствии с действующим законодательством.

© АО «Нэксайн», 1992–2023

АО «Нэксайн»

Россия, 199155, Санкт-Петербург, ул. Уральская, д.4 лит.Б, помещение 22Н

Тел.: + 7 (812) 326-12-99; факс: + 7 (812) 326-12-98.

office@nexign.com; www.nexign.com

Содержание

1. Общие сведения	4
2. Системные требования	5
3. Архитектура продукта	6
4. Настройка	7
4.1. Настраиваемые параметры SAM	7
4.2. Параметры в файле application.yml	7
5. Особенности эксплуатации	8
5.1. Аргументы	8
5.2. Предикаты	9
5.3. Действия	11
5.4. Команды	12
5.4.1. Создание команды	13
5.4.2. Настройка алгоритма	14
5.4.3. Запуск отладки команды	16
5.4.4. Версии команд	17
5.5. Оборудование	19
5.6. Пулы	23
5.7. Данные аутентификации	25
5.8. Конфигурации	26
5.9. Коллекции	27
5.9.1. Создание новой коллекции	28
5.9.2. Связывание сущности с коллекцией	29
5.9.3. Экспорт и импорт коллекций	30
5.9.4. Удаление коллекции	31
5.10. Группы пользователей	31
5.11. Режим эмуляции	33
6. Журналирование	34
7. Мониторинг	36

1. Общие сведения

Продукт «Управление подключением услуг» (SAM) предназначен для управления телекоммуникационным оборудованием оператора связи, поддерживающим работу по протоколу HTTP/HTTPS, в соответствии с бизнес-логикой предоставления абонентам телекоммуникационных услуг.

В документе используются следующие термины:

Коллекция

Набор сущностей SAM (команд, предикатов, действий, аргументов, пулов и др.), который используется для резервного копирования и переноса на другой экземпляр SAM.

Команда

Последовательность действий на телекоммуникационном оборудовании и базе данных, описываемая алгоритмом управления услугой связи.

Конфигурация

Описание определенного телекоммуникационного оборудования и соответствующего ему набора сущностей.

Предикат

Условие, применяемое в алгоритме обработки команды и определяющее ход обработки команды в зависимости:

- от результата выполнения действия;
- конкретного экземпляра оборудования, на котором выполняется действие;
- переданных входных аргументов.

2. Системные требования

Для корректного функционирования SAM необходимо следующее программное обеспечение:

- операционная система (одна из перечисленных):
 - Red Hat Enterprise Linux Server release 7 или 8;
 - РЕД ОС 7 или 8;
- SQL-база данных PostgreSQL 14;
- Java 11;
- для клиентской web-части управления и конфигурирования SAM – интернет-браузер Google Chrome.

Численность персонала, которому разрешено работать с SAM, определяется Оператором связи.

Режим работы персонала определяется нормативными документами Оператора связи.

Администрирование SAM предполагает следующие служебные обязанности:

- построение алгоритмов управления услугами, которые предоставляются абонентам;
- конфигурирование системы запросов к телекоммуникационному оборудованию;
- конфигурирование параметров доступа к телекоммуникационному оборудованию;
- диагностика работы продукта, локализация и устранение сбоев;
- мониторинг производительности, оптимизация и настройка продукта.

Квалификационные требования:

- знание языков: groovy, XML;
- знание протоколов: HTTP, TELNET, TCP/IP (при необходимости – SOAP);
- знание системы команд для используемого оборудования (коммутаторов, VM и т.п.).



Примечание.

Перечисленные выше обязанности могут быть возложены на администраторов биллинговой системы.

3. Архитектура продукта

Продукт SAM включает в себя следующие части:

- клиентская часть (сервер SAM) – Java standalone-приложение, которое в том числе содержит средства обеспечения пользовательского интерфейса;
- схема данных.

Сервер SAM принимает запросы на выполнение команд от внешних приложений либо по протоколам HTTP, либо по AMQP из очередей RabbitMQ.

При получении запроса на выполнение команд сервер SAM формирует запросы к оборудованию или СУБД в соответствии с правилами, определенными пользователем, и параметрами входного запроса. Последовательность действий и выбор конкретного экземпляра оборудования для отправки каждого запроса определяется алгоритмом, задаваемым пользователем при конфигурировании SAM-команды. Команды и аргументы команд создаются пользователем при помощи административного интерфейса. Вызов команд SAM осуществляется посредством вызова метода выполнения команды по протоколу HTTP или обработки очередей RabbitMQ. При обращении к оборудованию, использующему специфические протоколы, могут применяться специальные адаптеры.

Запросы на управление услугами могут обрабатываться группой экземпляров SAM, что обеспечивает гибкое масштабирование, «горячее» резервирование и равномерное распределение нагрузки на всю группу экземпляров SAM.

Настройка системы управления оборудованием выполняется с помощью многофункционального визуального административного интерфейса SAM. Интерфейс предоставляет развитые средства алгоритмизации управляющих команд и позволяет создавать комплексные команды, включающие в себя управление множеством телекоммуникационных услуг на различном оборудовании (кроссплатформенные услуги). Упростить администрирование системы помогает использование конфигураций и коллекций. Каждая конфигурация представляет собой совокупность описания определенного телекоммуникационного оборудования и соответствующего ему набора команд, действий и предикатов.

4. Настройка

Для настройки SAM:

- установите требуемые значения настроечных параметров SAM (см. раздел «[Настроечные параметры SAM](#)»); эти параметры также могут быть заданы в инсталляторе продукта (см. Руководство по установке [SAM-DOC_INSTALL]);
- задайте параметры в конфигурационном файле /config/application.yml (см. раздел «[Параметры в файле application.yml](#)»);
- задайте параметры журналирования в файле logback.xml (см. главу «[Журналирование](#)»);
- установите значения переменных окружения через setenv.sh.

4.1. Настроечные параметры SAM

Настроечные параметры SAM хранятся в локальной конфигурации. Локальная конфигурация – это YML-документ с уникальным именем, который находится в узле ZooKeeper /ps/config/apps/sam/localConfigs/.

Допускается создание нескольких локальных конфигураций SAM.

4.2. Параметры в файле application.yml

Задайте в конфигурационном файле /config/application.yml имя конфигурации параметров продукта и строку соединения с ZooKeeper.

5. Особенности эксплуатации

В главе приводится описание особенностей реализации и использования продукта.

5.1. Аргументы

Аргументы представлены отдельным справочником, который открывается выбором в главном меню пункта **Аргументы**. Справочник содержит список аргументов, значение по умолчанию которых может быть другим аргументом или значением. Для аргументов типа «строка» допускается использовать макроподстановки, вместо которых будут подставлены значения аргументов.

Синтаксис макроподстановки:

```
${argName1}
```

где `argName1` – аргумент с именем `argName1`.

При создании аргумента обязательными для заполнения являются поля **Имя** и **Тип данных**. Дополнительно можно задать значение по умолчанию и параметры преобразования (глобальный трансформер).

Преобразование (глобальный трансформер) задается в виде регулярного выражения.

Для работы с аргументами выберите в web-интерфейсе пункт меню **Аргументы** – откроется форма «Справочник аргументов» (см. [Рис. 1](#)).

Имя	Описание	Значение по умолчанию	Преобразование	Чувств...	Тип данных	Тэги	Встроенный
EQUIPMENT_RESPONSE_MESSAGE	Value of equipment respon...			Нет	string		Да
EQUIPMENT_RESPONSE_CODE	Value of equipment respon...			Нет	long		Да
arg2		Нет		Нет	bool		Нет
PREDICATE_RESULT	Value of predicate result			Нет	string		Да
arg1		Да		Нет	bool		Нет
RESPONSE	Value of command/action r...			Нет	string		Да
argString		test		Нет	string		Нет

Рис. 1. Справочник аргументов

Для создания нового аргумента:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание аргумента» (см. [Рис. 2](#)):
 - введите имя и описание аргумента;
 - выберите из раскрывающегося списка тип данных аргумента;

- установите значение аргумента по умолчанию; значение может быть задано ссылкой на другой аргумент или значением типа данных аргумента;
- при необходимости установите флаг «Преобразование» и введите регулярные выражения и выражения для замены, задающие правила преобразования значения аргумента;
- задайте тэги;
- нажмите на кнопку **Сохранить**.

Справочник аргументов

Имя	Описание	Значение по умолчанию	Преобразование	Чувствительный
ARG_SET_IN_GROOVY_ACTION				Нет
ARGWITHTRANSFORMER			<code>[{"inputArgument":"ARG WITHTRANSFORMER","pipe":{"regex":{"<aCode>[a-z](1,1)?<dCode>[0-9](2,2)","replacement":"\${dCode}\${aCode}"},"regex":{"<rCode>[a-z](1,1)\\s[0-9](2,2)","replacement":"-\${rCode}-"}}}]</code>	Нет
CHOOSE_FIRST_BRANCH		Да		Нет
COMMAND_RESULT	Value of command result			Нет
DB_RESULT				Нет
DB_TEXT		some value		Нет
EQUIPMENT_RESPONSE_CODE	Value of equipment response code			Нет

Создание аргумента

* Имя:

Описание:

* Тип данных:

Чувствительный

Преобразование

* Входной аргумент:

Регулярные выражения

* Регулярное выражение:

* Замена:

Тэги:

Рис. 2. Создание аргумента

5.2. Предикаты

SAM использует следующие типы предикатов:

- RegExp – предикаты в виде регулярных выражений;
- XSLT – вычисляются с помощью XSLT-обработчика и предназначены для определения результата выполнения шага алгоритма, возвращающего сложный XML-документ;
- Groovy – алгоритм на языке Groovy.

Для предиката можно установить флаг «Инvertировать результат». Этот флаг действует как логическое отрицание: если обработчик предиката возвратил «Ложь», то предикат считается имеющим значение «Истина», и наоборот – «Истина» инvertируется в «Ложь».

Каждый предикат должен относиться к единственной категории, которая определяет его назначение, например, «Определение результата выполнения запроса», «Выбор очередного шага алгоритма», «Определение кода ошибки оборудования» и пр.

Для работы с предикатами выберите в web-интерфейсе пункт меню **Предикаты** – откроется форма «Справочник предикатов» (см. [Рис. 3](#)).

Наименование	Текст	Тип	Категория	Инвертироват...
action_repeat_predicate	return true	Groovy	ACTION_REPEAT	Нет
equipment_selection_predicate	return context.get("arg1") == true	Groovy	EQUIPMENT_SELECTION	Нет
equipment_selection_predicate 2	return context.get("arg1") == false	Groovy	EQUIPMENT_SELECTION	Нет
false_pred	return false;	Groovy	ACTION_RESULT	Нет
newPredicate	return true	Groovy	ACTION_RESULT	Нет
predicate_depend_argString	sleep(500); if (context.get("argString").equals("test")) { return true; } else if(context.get("argString").equals("test_false")) { return false; } else { throw new Exception("error") }	Groovy	ACTION_RESULT	Нет
true_pred	sleep(900); return true;	Groovy	ACTION_RESULT	Нет

Рис. 3. Справочник предикатов

Для создания нового предиката:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание предиката» (см. [Рис. 4](#)):
 - введите наименование предиката;
 - выберите из раскрывающегося списка категорию предиката;
 - выберите из раскрывающегося списка тип предиката;
 - введите текст предиката;
 - при создании предиката для условия с отрицанием установите флаг «Инвертировать результат» – будет получен результат, обратный возвращенному представленным выражением;
 - нажмите на кнопку **Сохранить**.

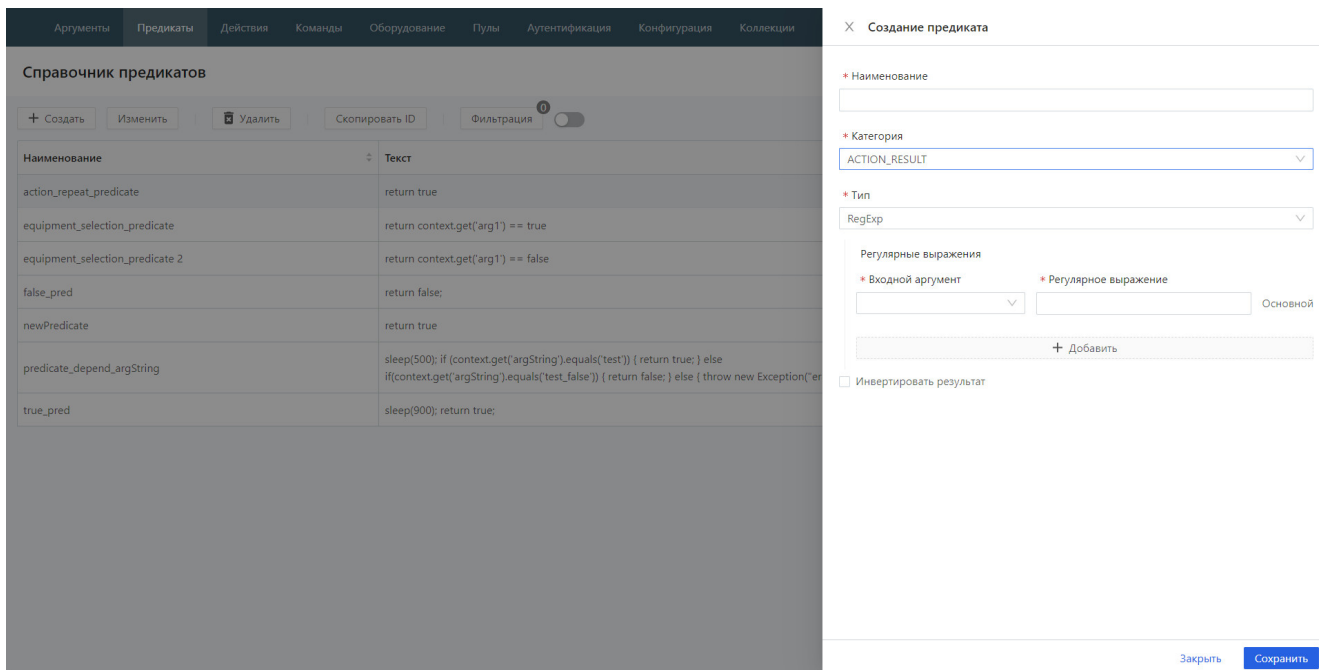


Рис. 4. Создание предиката

5.3. Действия

Данные о действиях содержатся в справочнике (см. [Рис. 5](#)), который открывается с помощью пункта **Действия** главного меню.

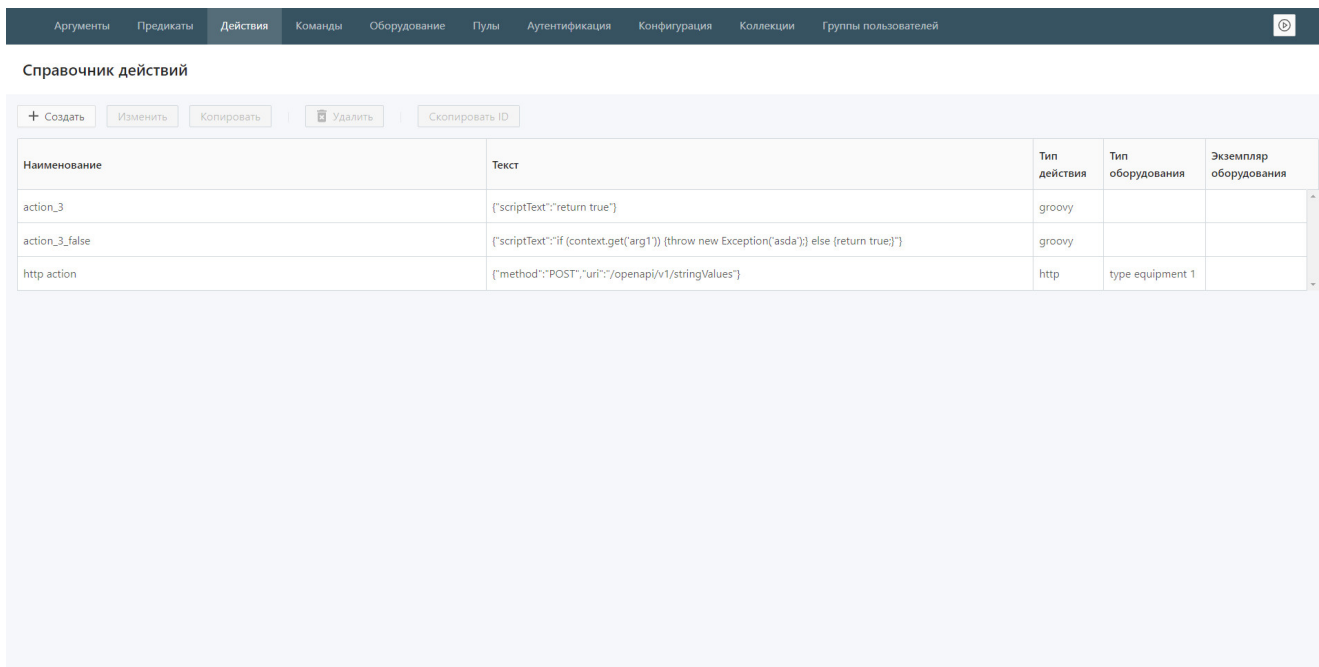


Рис. 5. Справочник действий

Для создания нового действия:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание действия» (см. [Рис. 6](#)):

- введите имя и описание действия;
- выберите в раскрывающемся списке в поле **Тип** тип действия (http, groovy, database);
- после выбора типа действия заполните свойства создаваемого действия в отобразившихся полях (состав полей зависит от выбранного типа);
- выберите в раскрывающемся списке в поле **Тип оборудования** тип оборудования, которому адресовано создаваемое действие;
- при необходимости выберите экземпляр оборудования, с которым требуется явно связать действие (в этом случае не будут выполняться предикаты маршрутизации), из раскрывающегося списка в поле **Экземпляр оборудования**;
- нажмите на кнопку **Сохранить**.

Наименование	Текст
action_3	["scriptText":"return true"]
action_3_false	["scriptText":"if (context.get('arg1')) {throw new Exception('asda')} else {return true}"]
http action	["method":"POST","uri":"/openapi/v1/stringValues"]

Рис. 6. Создание действия

5.4. Команды

Каждая команда SAM – это ориентированный ациклический граф атомарных с точки зрения SAM шагов. SAM при выполнении команды обходит дерево, начиная с начальной вершины – пустого шага, отмечающего точку входа в алгоритм. Следующий выполняемый шаг выбирается с помощью предикатов выбора очередного шага: находясь в некоторой вершине, SAM вычисляет все предикаты выбора очередного шага её дочерних узлов.

Только один предикат должен иметь значение «Истина». Если таких предикатов несколько, то это является прочей ошибкой выполнения команды.

Поддерживаются безусловные переходы и переходы по умолчанию. Если для вершины задан безусловный переход, то для нее не может быть задано других переходов.

Для вершины может быть задан один переход по умолчанию. Если он задан, то этот переход будет выбран в случае, если не сработали предикаты выбора шага условных переходов.

Может быть задан единственный переход – переход по умолчанию. В этом случае он работает как безусловный переход.

Если все предикаты выбора шага не сработали и не задан переход по умолчанию, то это трактуется как прочая ошибка выполнения команды.

Результат выполнения шага может быть сохранен в аргумент или выведен в ответе с заданным

тэгом. В шаге может быть вызвано действие или другая команда.

Для возможности ветвления алгоритма в зависимости от результата последнего шага и команды используются встроенные аргументы `LAST_STEP_RESULT` (удаляется из контекста перед началом выполнения каждого шага) и `COMMAND_RESULT` (удаляется из контекста перед началом выполнения команды). Значения данных аргументов вставляются (переопределяются, если уже есть) в контекст после выполнения каждого шага.

Данные о командах содержатся в справочнике (см. [Рис. 7](#)), который открывается с помощью пункта **Команды** главного меню.

Код	Наименование	Описание	Использовать постоянно...	Внутренняя	Активна
cmd1	cmd1		Нет	Нет	Да
cmd2	cmd2		Нет	Нет	Да
cmd3	cmd3		Нет	Нет	Да
cmd4	fail command		Нет	Нет	Да
executeGroovyAction	Выполнение действия groovy с выставлением аргумента arg_set_in_groovy_action = "argument value"		Нет	Нет	Да
groovyOrEquipmentBranch	Команда с двумя ветвями: команда с groovy или http-действие с выбором экземпляра		Нет	Нет	Да
alwaysFalse	Команда с ошибкой		Нет	Нет	Да
argWithTransformer	Преобразование аргумента с трансформером		Нет	Нет	Да

Рис. 7. Справочник команд

5.4.1. Создание команды

Для создания новой команды:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание команды» (см. [Рис. 8](#)):
 - введите код, имя и описание команды;
 - при необходимости установите флаг «Использовать постоянное соединение»;
 - установите флаг «Внутренняя», если нужно запретить вызов команды извне;
 - нажмите на кнопку **Настроить алгоритм** для настройки алгоритма команды (см. подраздел «[Настройка алгоритма](#)»);
 - нажмите на кнопку **Сохранить**.

Код	Имя	Описание
Cactus	CactusName	
nexignCode	nexign	тестовая команда

Рис. 8. Создание команды

5.4.2. Настройка алгоритма

Для настройки алгоритма команды нажмите на кнопку **Настроить алгоритм** на форме создания или редактирования команды.

На странице «Настройка алгоритма» содержится только начальный шаг алгоритма. Используя представленные в окне инструменты, необходимо создать полный алгоритм обработки новой команды. Доступны следующие действия:

- [создание шага](#);
- изменение шага;
- удаление шага;
- [создание перехода](#);
- изменение перехода;
- удаление перехода.

Создание шага

Для создания шага:

1. На форме «Настройка алгоритма» нажмите на кнопку **Создать шаг**.
2. В открывшемся окне «Создание шага» (см. [Рис. 9](#)):
 - введите имя шага;
 - установите переключатель в поле **Действие** (если на создаваемом шаге требуется выполнить действие) или **Команда (подпрограмма)** (если на создаваемом шаге требуется выполнить команду);
 - выберите из списка соответствующее действие или команду;
 - в блоке «Обработка результата» установите переключатель для одного из вариантов:
 - **Предикат успешности** – выберите в раскрывающемся списке;
 - **Безусловно** – выберите безусловный результат шага («Успех»/«Неуспех»);
 - введите название переменной в поле **Сохранить результат в переменной** (необязательно);

- введите название атрибута в поле **Имя атрибута**, если требуется вернуть результат выполнения шага в атрибуте (необязательно);
- при необходимости установите флаги:
 - «Критичный шаг» – флаг указывает, что если шаг завершился с ошибкой, то результат выполнения команды – ошибка, независимо от того, продолжается выполнение команды далее или нет;
 - «Финальный шаг» – флаг указывает, что если шаг успешен, то последующие шаги не влияют на результат выполнения команды;
 - «Игнорировать ошибки» – при установленном флаге выполнение команды будет продолжено, даже если результат выполнения шага – ошибка;
- нажмите на кнопку **Сохранить**.

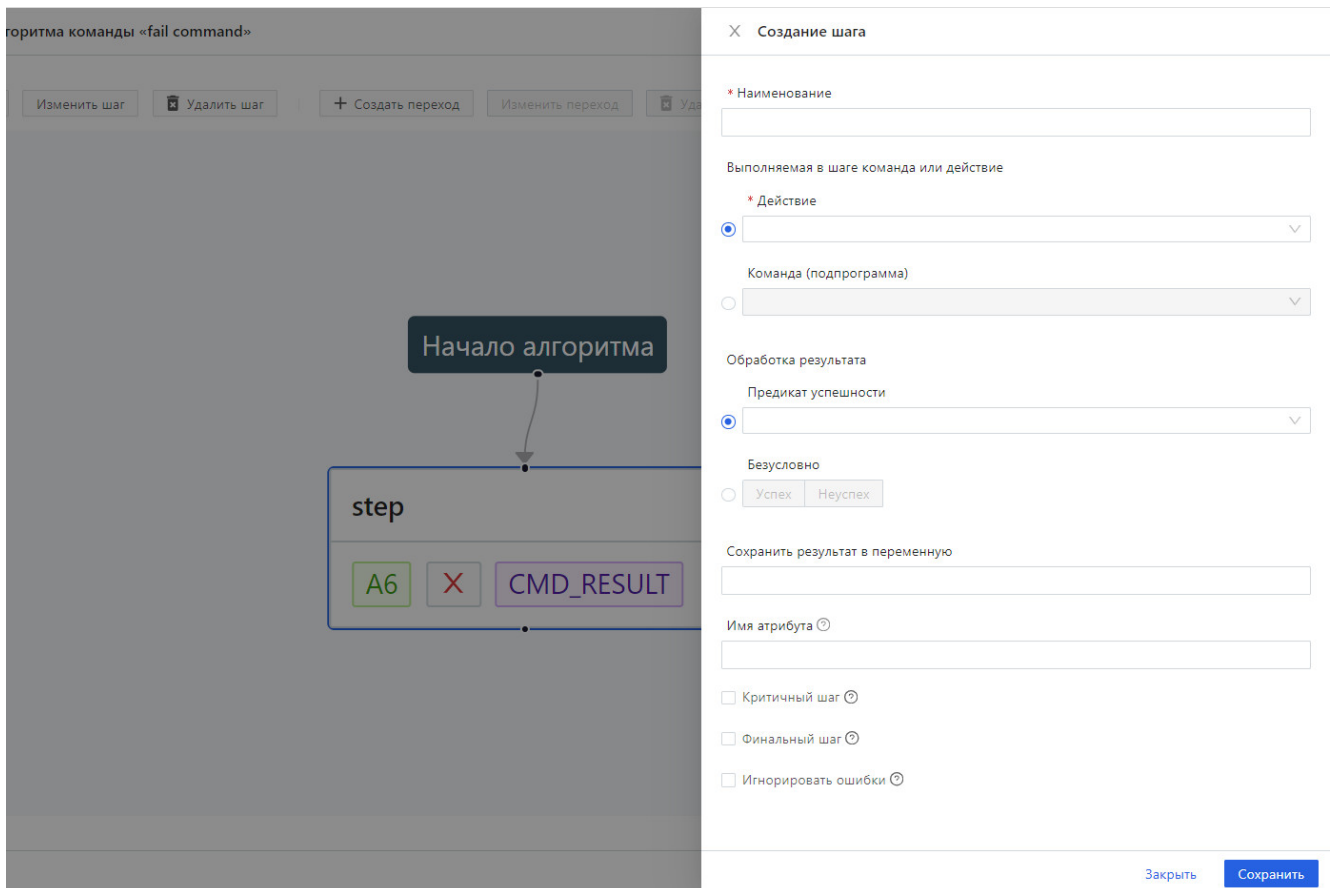


Рис. 9. Создание шага

Создание перехода

Для создания перехода:

1. На форме «Настройка алгоритма» нажмите на кнопку **Создать переход**.
2. В открывшемся окне «Создание перехода» (см. [Рис. 10](#)):
 - выберите из раскрывающегося списка шаги, являющиеся началом и концом перехода;
 - установите переключатель в поле **Тип перехода**;
 - нажмите на кнопку **Сохранить**.

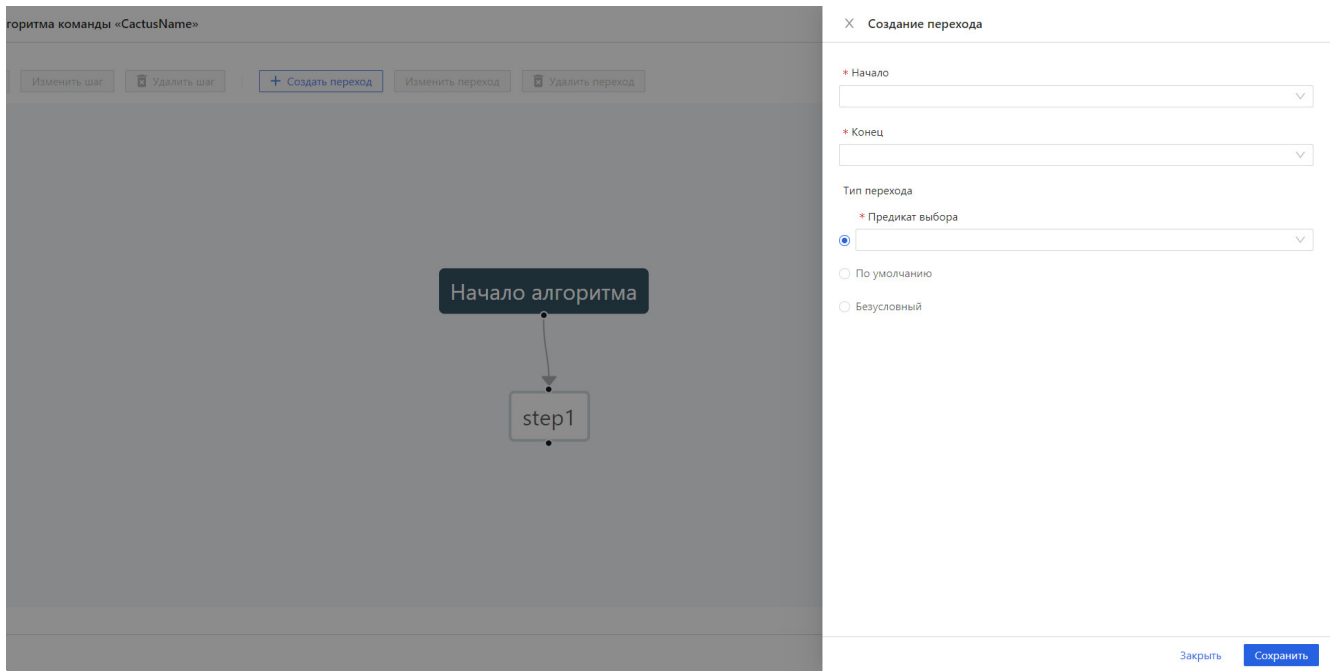


Рис. 10. Создание перехода

5.4.3. Запуск отладки команды

Команда может быть запущена в режиме отладки. Для этого выберите нужную команду и нажмите на кнопку **Запустить отладку** (см. [Рис. 11](#)).

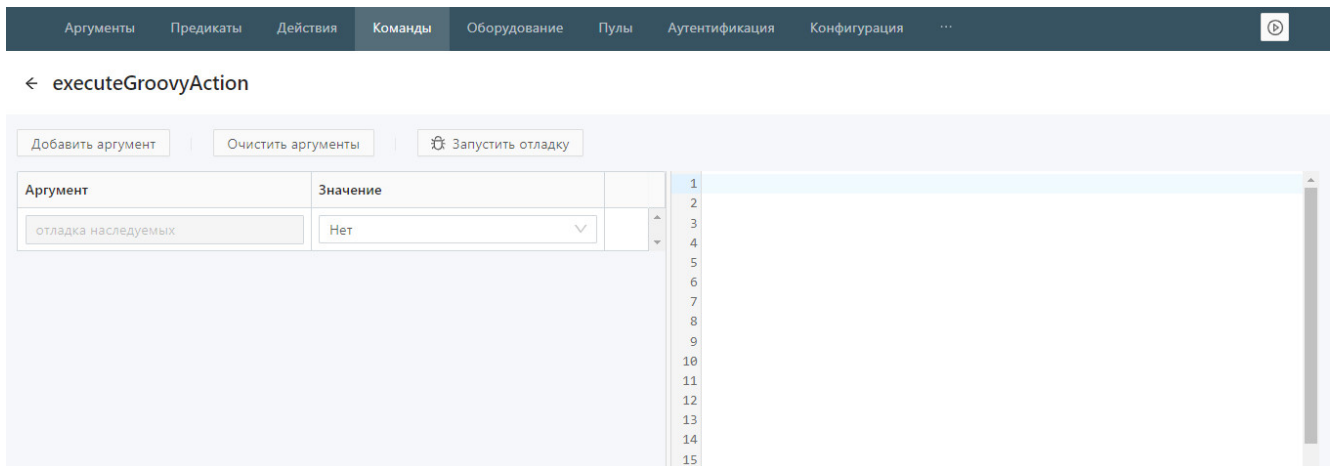


Рис. 11. Запуск отладки

В открывшемся окне добавьте требуемые аргументы, укажите их значения и нажмите на кнопку **Запустить отладку**.

Значение **Да** в поле **Отладка наследуемых** задает необходимость отладки команд или действий, вызываемых из данной команды, и не распространяется на действия, указанные в шагах основной команды.

После окончания работы команды в правой части экрана отобразится информация о выполнении команды (см. [Рис. 12](#)).

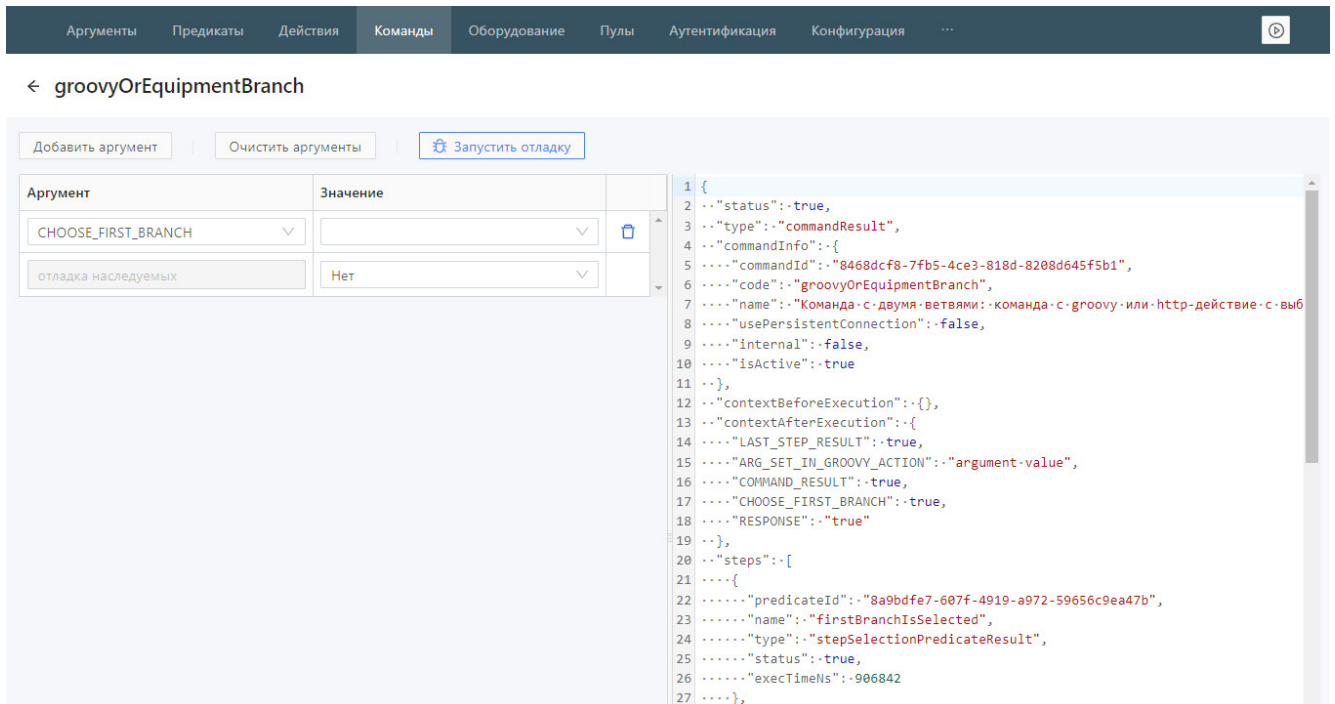


Рис. 12. Результат запуска отладки

Результат содержит:

- данные обо всех этапах выполнения команды, включая предикаты;
- состояние контекста до и после выполнения команды, а также изменения контекста на конкретном элементе, включая предикаты.

Каждый элемент, включая саму команду, содержит время выполнения. Значения чувствительных аргументов маскируются.

5.4.4. Версии команд

Для каждой команды ведется учет версий. Для просмотра версий выберите команду и нажмите на кнопку **История** (см. [Рис. 13](#)).

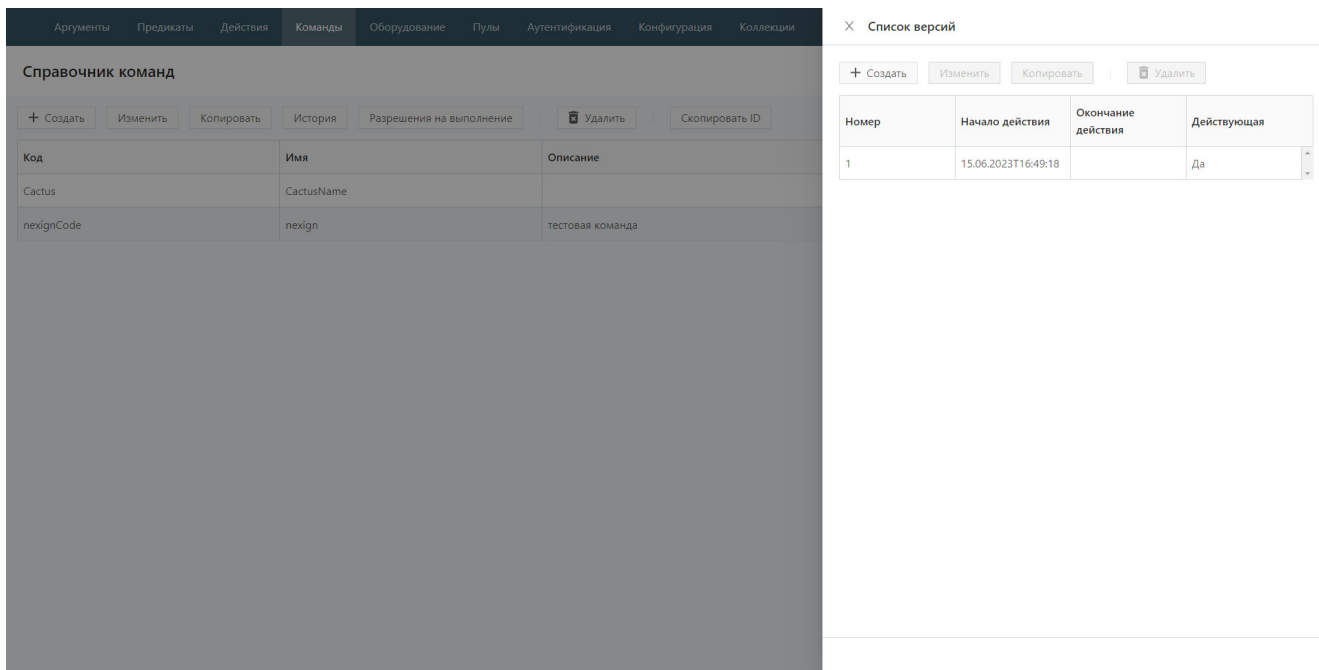


Рис. 13. Версии команды

Для создания новой версии:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание версии» (см. [Рис. 14](#)):
 - задайте даты начала и окончания действия версии команды;
 - настройте алгоритм команды (см. раздел «[Настройка алгоритма](#)»);
 - нажмите на кнопку **Сохранить**.

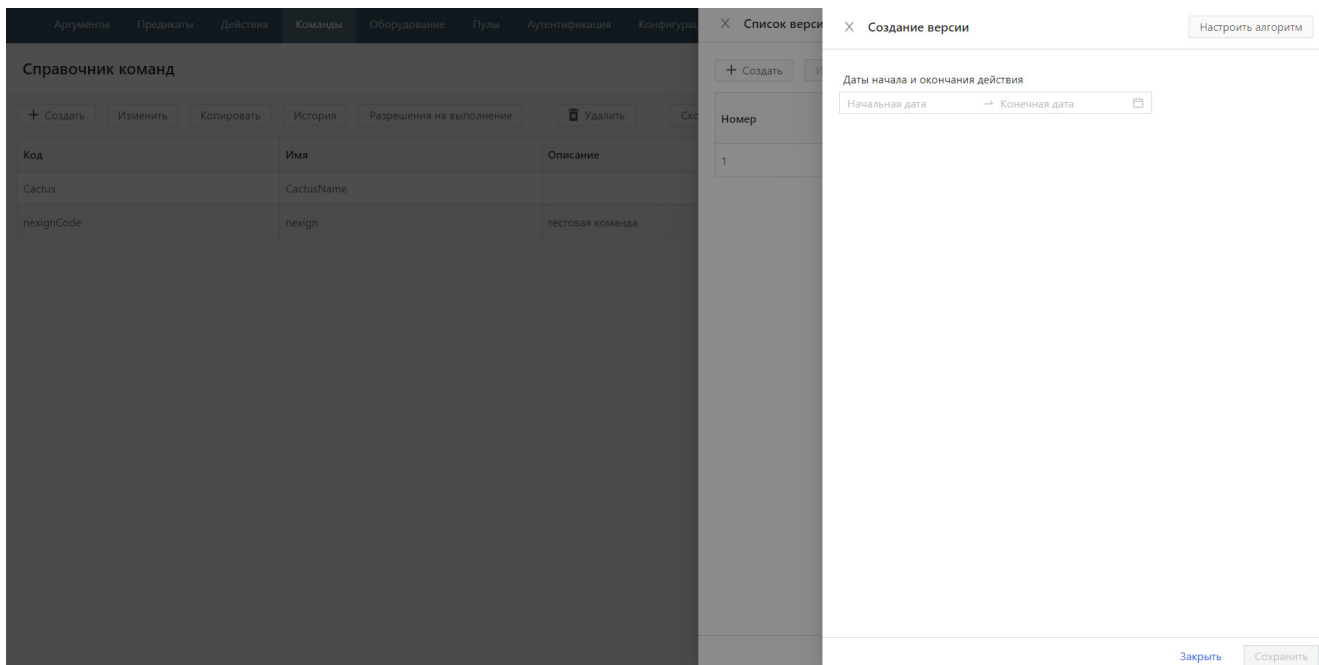


Рис. 14. Создание версии команды

Для редактирования версии команды:

1. Выберите версию, в которую нужно внести изменения.

2. Нажмите на кнопку **Изменить**.
3. В открывшемся окне «Редактирование версии» измените значения полей.
4. Нажмите на кнопку **Сохранить** для сохранения изменений.

5.5. Оборудование

Данные о телекоммуникационном оборудовании, с которым взаимодействует SAM, а также средства для работы с оборудованием содержатся в справочнике, который открывается с помощью пункта **Оборудование** главного меню.

Справочник содержит три списка (см. [Рис. 15](#)):

- «Классы оборудования» – полный список классов оборудования;
- «Типы оборудования» – список типов оборудования выбранного класса;
- «Экземпляры оборудования» – список экземпляров оборудования выбранного типа.

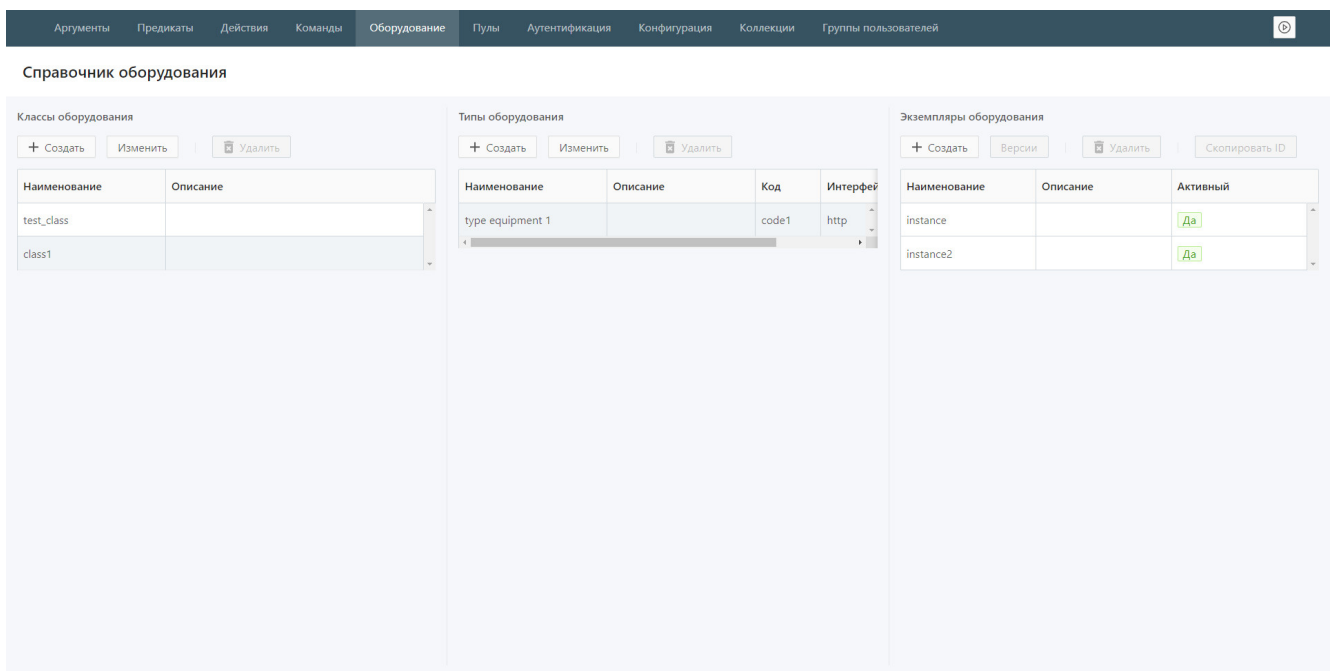


Рис. 15. Справочник оборудования

Для создания нового класса оборудования:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание класса оборудования» (см. [Рис. 16](#)):
 - введите имя и описание класса оборудования;
 - нажмите на кнопку **Сохранить**.

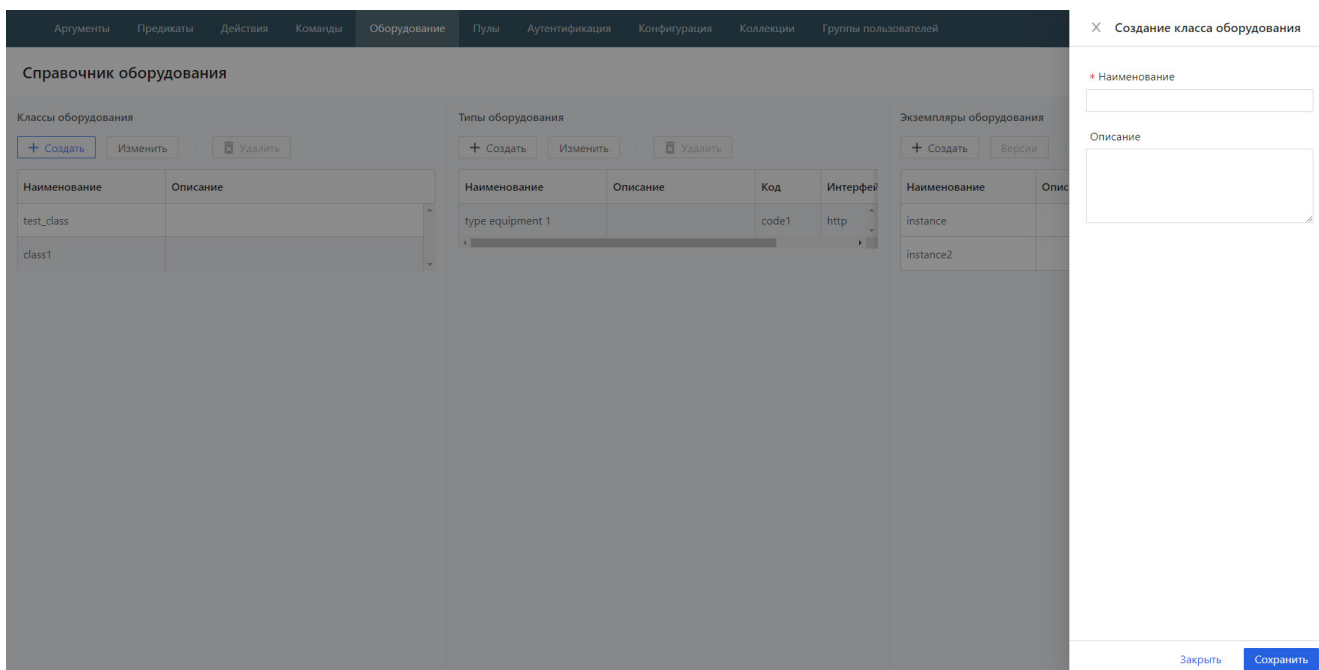


Рис. 16. Создание класса оборудования

Для создания нового типа оборудования:

1. Выберите класс оборудования, для которого требуется создать новый тип.
2. Нажмите на кнопку **Создать**.
3. В открывшемся окне «Создание типа оборудования» (см. [Рис. 17](#)):
 - введите имя, описание и код типа оборудования;
 - выберите в раскрывающемся списке в поле **Интерфейс** один или несколько интерфейсов, поддерживаемых данным типом оборудования;
 - нажмите на кнопку **Сохранить**.

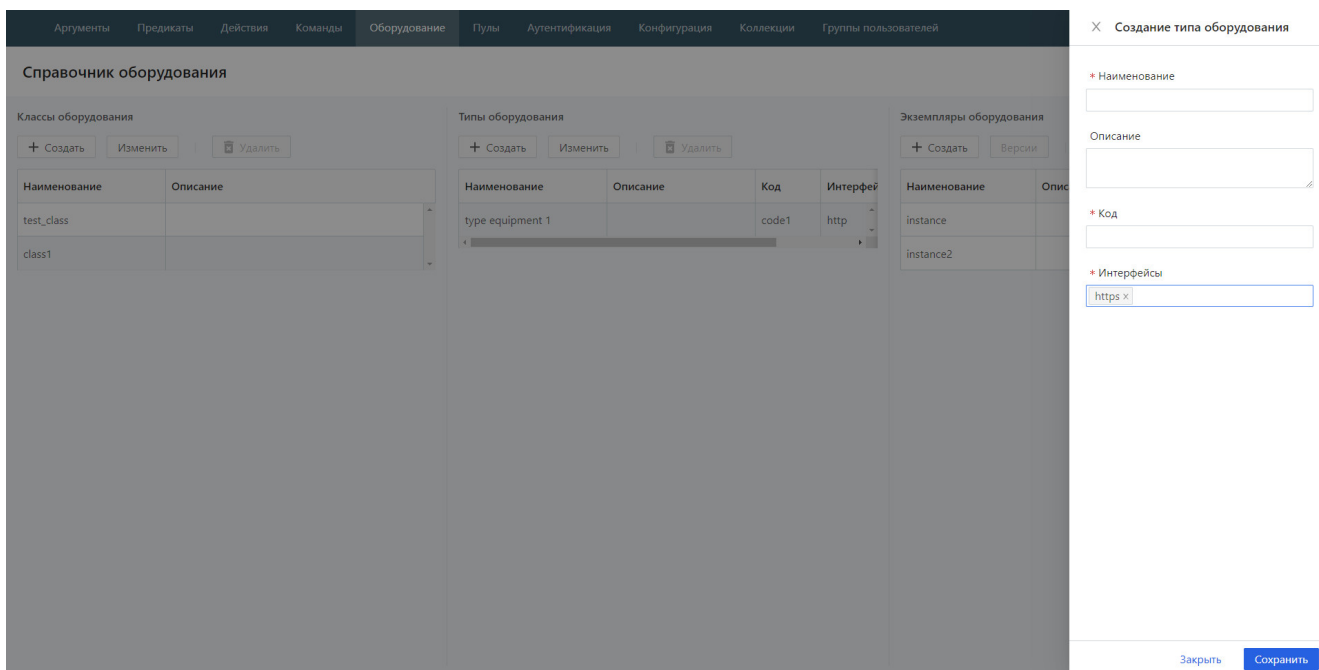


Рис. 17. Создание типа оборудования

Для создания нового экземпляра оборудования:

1. Выберите тип оборудования, для которого требуется создать экземпляр.
2. Нажмите на кнопку **Создать**.
3. В открывшемся окне «Создание экземпляра оборудования» (см. [Рис. 18](#)):
 - введите наименование и описание экземпляра оборудования;
 - выберите в раскрывающемся списке предикаты выбора оборудования (необязательно); список содержит предикаты категории `EQUIPMENT_SELECTION`;
 - выберите в раскрывающемся списке предикаты ошибок оборудования (необязательно); список содержит предикаты категории `EQUIPMENT_ERROR`;
 - задайте свойства интерфейсов: выберите пул в поле **Пул оборудования** и введите в поле **Тайм-аут запроса в мс** значение тайм-аута для запросов;
 - при необходимости обработки возможных ошибок установите флаг «Обработка ошибок» и задайте параметры обработки:
 - в поле **Предикаты повтора действий** выберите предикат; список содержит предикаты категории `ACTION_REPEAT`;
 - укажите количество повторов и интервал между ними;
 - введите значение ограничения количества запросов к экземпляру в секунду (необязательно); по умолчанию производительность экземпляра не ограничена;
 - введите период действия записи об экземпляре оборудования в полях **Дата начала действия** и **Дата окончания действия** (необязательно);
 - нажмите на кнопку **Сохранить**.

Справочник оборудования

Классы оборудования

Наименование	Описание
httpClass	
DB	
HTTP	
HTTP_ERROR	

Типы оборудования

Наименование	Описание
httpType	

* Наименование

Описание

Предикаты выбора оборудования

Предикаты ошибок оборудования

Свойства интерфейсов

* Пул оборудования \odot * Тайм-аут запроса в мс

http

Обработка ошибок

Ограничение количества запросов в секунду

Дата начала действия

Выберите дату

Дата окончания действия

Выберите дату

Закрыть Сохранить

Рис. 18. Создание экземпляра оборудования

Для каждого экземпляра оборудования ведется учет версий. Для просмотра версий выберите экземпляр оборудования и нажмите на кнопку **Версии** (см. [Рис. 19](#)).

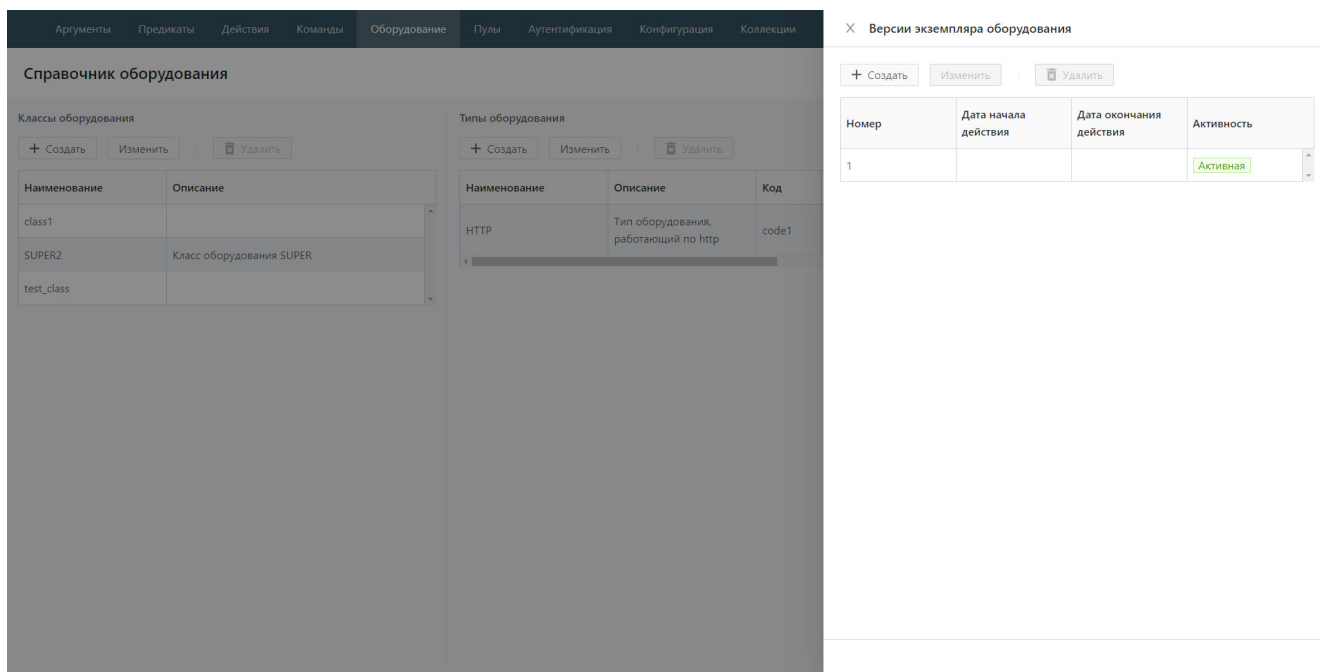


Рис. 19. Версии экземпляра оборудования

Для создания новой версии:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание версии экземпляра оборудования» (см. [Рис. 20](#)):
 - введите имя и описание экземпляра оборудования;
 - выберите в раскрывающемся списке предикаты выбора оборудования (необязательно); список содержит предикаты категории `EQUIPMENT_SELECTION`;
 - выберите в раскрывающемся списке предикаты ошибок оборудования (необязательно); список содержит предикаты категории `EQUIPMENT_ERROR`;
 - задайте свойства интерфейсов: выберите пул в поле **Пул оборудования** и введите в поле **Тайм-аут запроса в мс** значение тайм-аута для запросов;
 - при необходимости обработки возможных ошибок установите флаг «Обработка ошибок» и задайте параметры обработки:
 - в поле **Предикаты повтора действий** выберите предикат; список содержит предикаты категории `ACTION_REPEAT`;
 - укажите количество повторов и интервал между ними;
 - введите период действия записи об экземпляре оборудования в полях **Дата начала действия** и **Дата окончания действия** (необязательно);
 - нажмите на кнопку **Сохранить**.

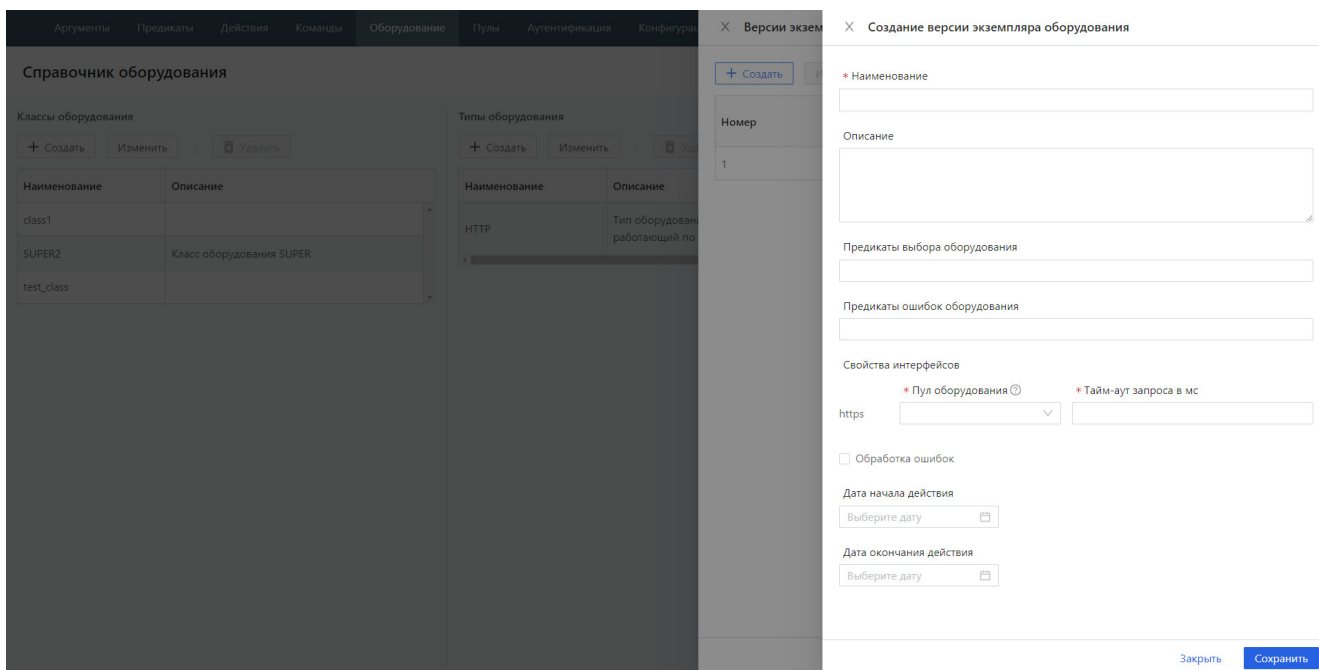


Рис. 20. Создание версии экземпляра оборудования

Для редактирования версии:

1. Выберите версию, в которую нужно внести изменения.
2. Нажмите на кнопку **Изменить**.
3. В открывшемся окне «Редактирование версии экземпляра оборудования» измените значения полей.
4. Нажмите на кнопку **Сохранить** для сохранения изменений.



Примечание.

Версии экземпляра оборудования не должны пересекаться во времени.
Версии в прошлом изменять запрещено.
Версии в будущем изменять разрешено.
При изменении текущей версии она закрывается и создается новая текущая версия.

5.6. Пулы

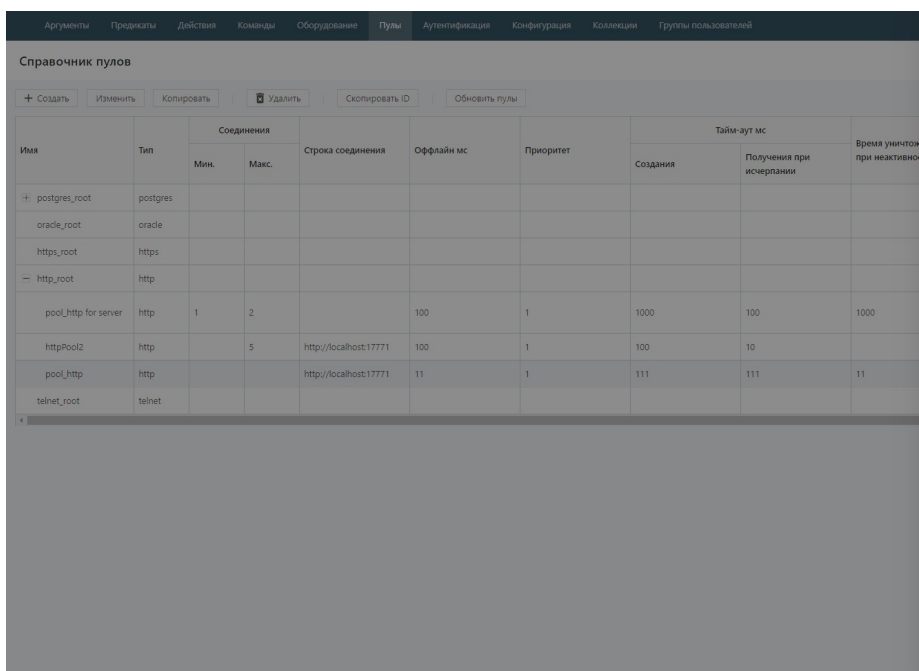
Данные о пулах оборудования содержатся в справочнике (см. [Рис. 21](#)), который открывается с помощью пункта **Пулы** главного меню.

Имя	Тип	Соединения		Строка соединения	Офлайн мс	Приоритет	Тайм-аут мс		Время уничтожения при неактивности мс	Команда
		Мин.	Макс.				Создания	Получения при исчерпании		
postgres_root	postgres									
oracle_root	oracle									
https_root	https									
http_root	http									
pool_http for server	http	1	2		100	1	1000	100	1000	
httpPool2	http		5	http://localhost:17771	100	1	100	10		100
pool_http	http			http://localhost:17771	11	1	111	111	11	
telnet_root	telnet									

Рис. 21. Справочник пулов

Для создания нового пула:

1. Выберите родительский пул, для которого требуется создать дочерний пул.
2. Нажмите на кнопку **Создать**.
3. В открывшемся окне «Создание пула» (см. [Рис. 22](#)):
 - введите имя пула;
 - заполните свойства создаваемого пула; по умолчанию значения свойств дочерних пулов наследуются от свойств родительского пула; при необходимости переопределения значений введите их вручную;
 - нажмите на кнопку **Сохранить**.



Создание пула

* Имя

Ping

Мин. соединений: Макс. соединений:
После применения конфигурации пулов будет взято значение 0 После применения конфигурации пулов будет взято значение 1000

Строка соединения
Будет унаследовано значение http://localhost:17771

Приоритет
Будет унаследовано значение 1

Тайм-аут создания мс
Будет унаследовано значение 111

Тайм-аут получения при исчерпании мс
Будет унаследовано значение 111

Время уничтожения при неактивности мс
Будет унаследовано значение 11

Офлайн мс
Будет унаследовано значение 11

Аутентификация

Рис. 22. Создание пула

5.7. Данные аутентификации

Данные о пользователях с возможностью аутентификации содержатся в справочнике (см. [Рис. 23](#)), который открывается с помощью пункта **Аутентификации** главного меню.

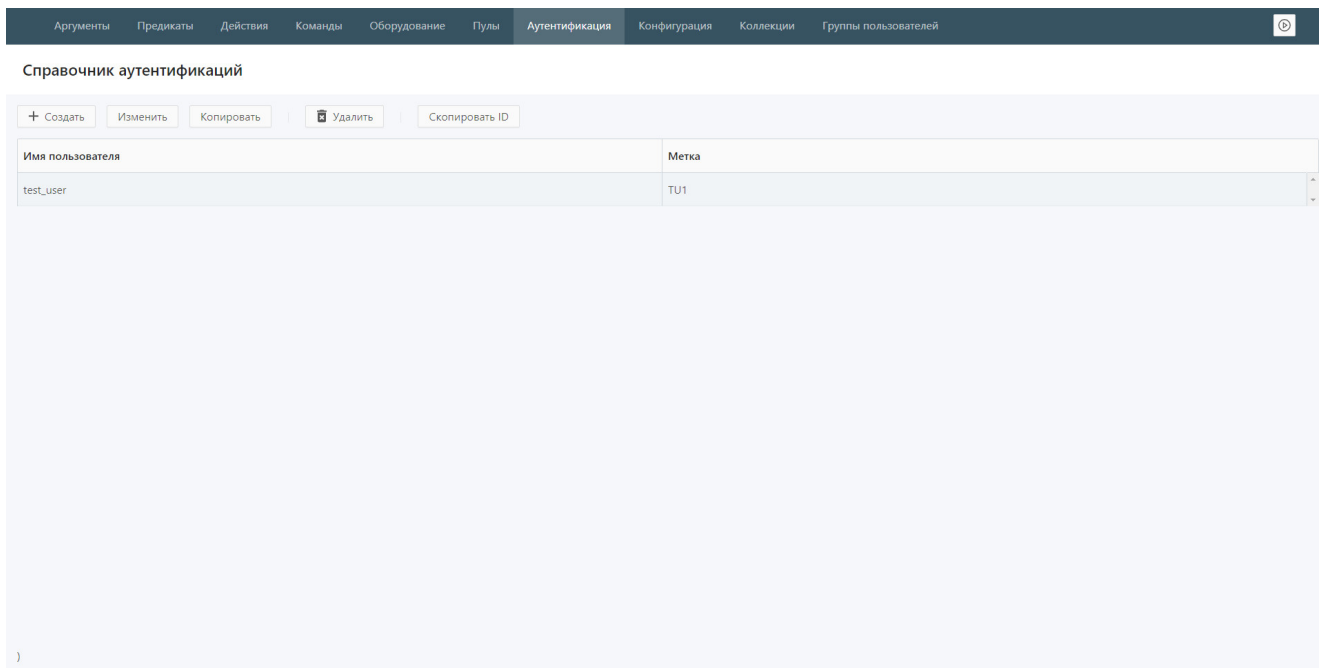


Рис. 23. Справочник аутентификаций

Для создания новой аутентификации:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание аутентификации» (см. [Рис. 24](#)):
 - введите значения обязательных полей **Имя пользователя** и **Пароль**;
 - заполните необязательное поле **Метка**;
 - нажмите на кнопку **Сохранить**.

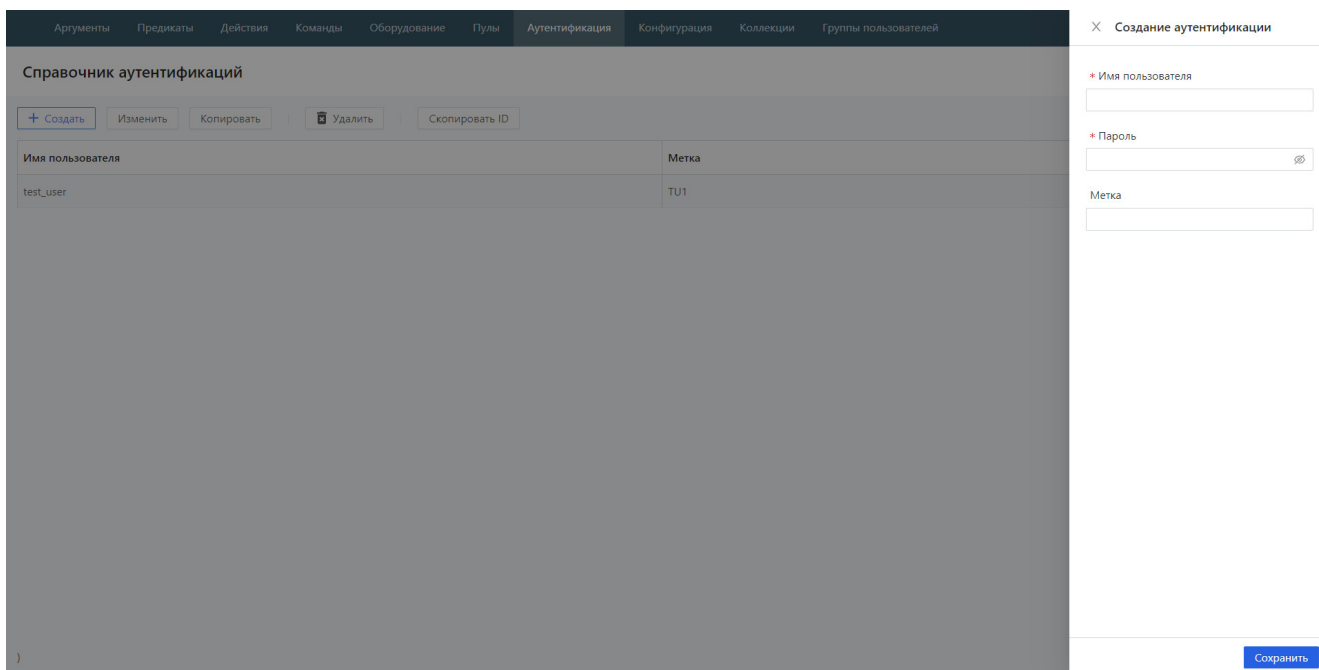


Рис. 24. Создание аутентификации

5.8. Конфигурации

Конфигурационная информация SAM (или «конфигурация») представляет собой совокупность оборудования, набора команд, алгоритмов, запросов, коллекций, предикатов и аргументов, то есть данных всех таблиц схемы базы данных SAM, за исключением таблиц журналов.

Конфигурацию SAM можно импортировать или экспортировать.

Сохранение конфигураций SAM выполняется в виде файла JSON.

Для импорта конфигурации:

1. Выберите пункт **Конфигурация** главного меню.
2. Нажмите на кнопку **Импорт** – откроется окно импорта конфигурации (см. [Рис. 25](#)).
3. Выберите файл JSON, содержащий конфигурацию, и нажмите на кнопку **Импортировать**.

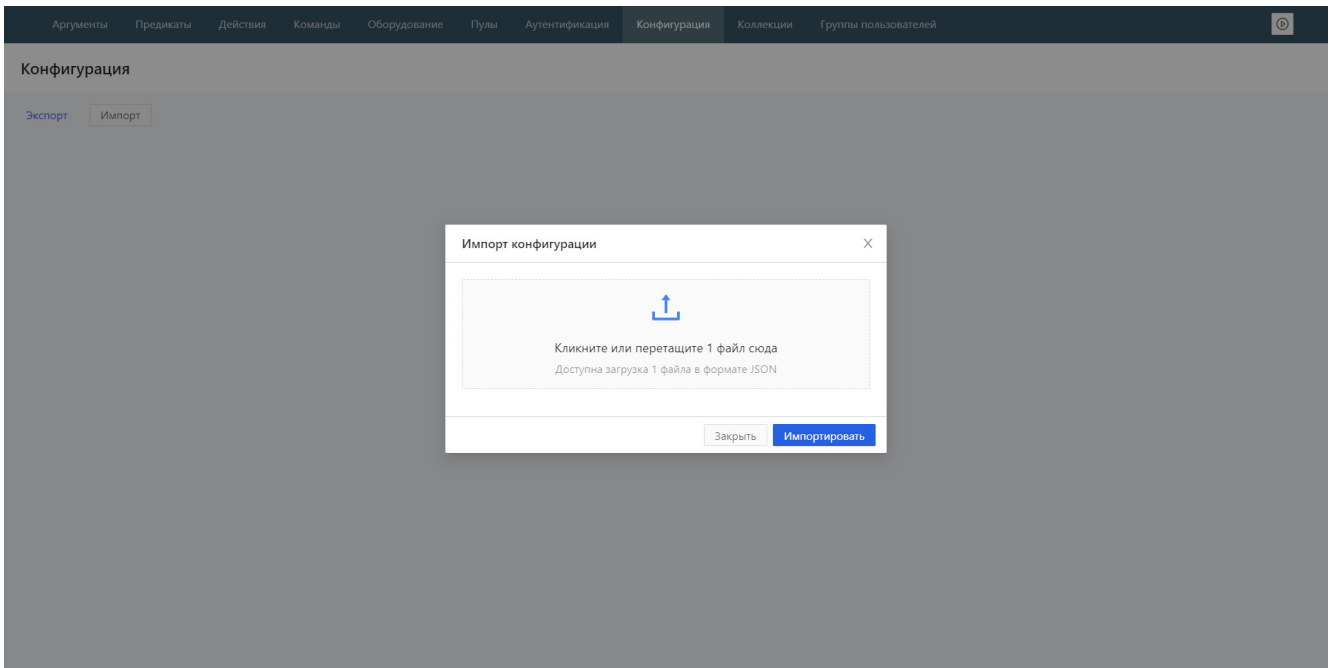


Рис. 25. Импорт конфигурации

5.9. Коллекции

Коллекция представляет собой набор сущностей SAM, который можно в любой момент загрузить в открытую конфигурацию или удалить из нее, тем самым быстро модифицировав текущую конфигурацию SAM.

Под сущностью SAM понимаются следующие объекты:

- команды;
- действия;
- предикаты;
- аргументы;
- данные аутентификации;
- классы, типы, экземпляры оборудования;
- пулы;
- группы пользователей;
- права на выполнение команд.

Перечисленные сущности могут быть явно, через web-интерфейс, включены в коллекцию (связаны с коллекцией) или удалены из коллекции. Каждая сущность может принадлежать только одной коллекции или быть свободной, то есть не принадлежать ни одной коллекции. Под удалением сущности из коллекции подразумевается удаление связи между сущностью и коллекцией.

Коллекции предназначены:

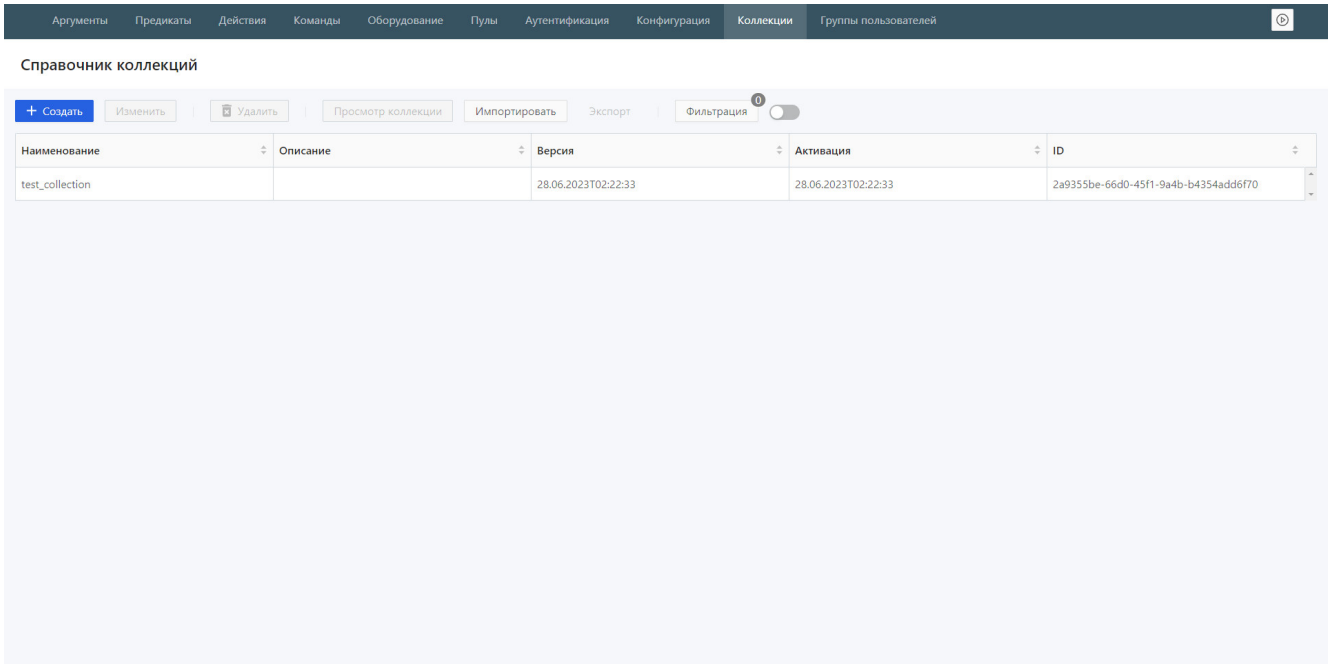
- для переноса части конфигурации (сущностей SAM) с одного экземпляра SAM на другой;
- для логического разделения конфигурации SAM на составные части по использованию, оборудованию и прочим критериям, что упрощает сопровождение конфигураций SAM и позволяет тиражировать общие действия, предикаты, команды управления конкретным оборудованием и т.п.;
- для осуществления прозрачного переноса частей общей конфигурации SAM с одной базы

данных на другую, а также подключения к существующей конфигурации SAM отдельных частей (коллекции), в том числе поставляемых в составе продуктов, основанных на SAM.

Web-интерфейс SAM поддерживает следующие операции с коллекциями и сущностями:

- [создание новой коллекции](#);
- [связывание сущности с коллекцией](#);
- [экспорт и импорт коллекций](#);
- [удаление коллекции](#).

Данные о коллекциях содержатся в справочнике (см. [Рис. 26](#)), который открывается с помощью пункта **Коллекции** главного меню.



Наименование	Описание	Версия	Активация	ID
test_collection		28.06.2023T02:22:33	28.06.2023T02:22:33	2a9355be-66d0-45f1-9a4b-b4354add6f70

Рис. 26. Справочник коллекций

5.9.1. Создание новой коллекции

Для создания новой коллекции:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание коллекции» (см. [Рис. 27](#)):
 - введите название и описание создаваемой коллекции;
 - нажмите на кнопку **Сохранить**.

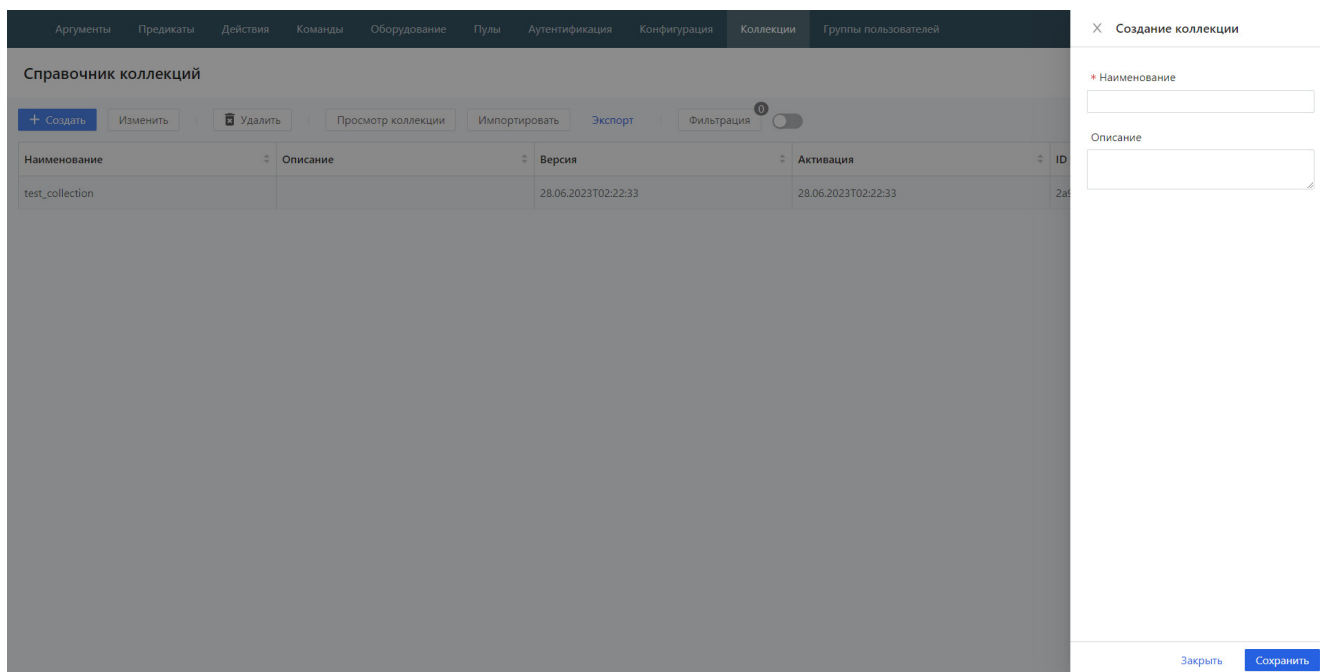


Рис. 27. Создание коллекции

5.9.2. Связывание сущности с коллекцией

В коллекцию могут быть включены любые виды сущностей SAM. Сущность добавляется в коллекцию без зависимостей (сущности, от которых зависит добавляемая сущность, в коллекцию автоматически не добавляются).

Конкретная сущность может входить только в одну коллекцию. При добавлении сущностей анализируется их вхождение в другие коллекции. В коллекцию добавляются только те сущности, которые не входят в другие коллекции.

Для добавления сущности в коллекцию:

1. Выберите коллекцию, состав сущностей которой требуется изменить, и нажмите на кнопку **Просмотр коллекции** – откроется форма «Коллекция» (см. [Рис. 28](#)).
2. Выберите на вкладках формы сущности, которые нужно добавить в коллекцию или исключить из нее.
3. Вернитесь на форму «Справочник коллекций», нажав на стрелку.

← Коллекция test_collection						
Команды Действия Предикаты Аргументы Аутентификация Оборудование Пулы						
	Код	Имя	Описание	Использовать ...	Внутренняя	Активна
<input type="checkbox"/>	Cactus	CactusName		Да	Да	Да
<input type="checkbox"/>	nexignCode	nexign	тестовая команда	Да	Да	Да

Рис. 28. Коллекция

5.9.3. Экспорт и импорт коллекций

Импорт коллекции выполняется для переноса сущностей коллекции на новый экземпляр. При импорте коллекций не выполняется очистка существующих сущностей. Сущности из коллекции добавляются на импортируемый экземпляр без удаления сущностей из существующей конфигурации.

При успешном прохождении всех проверок коллекция активируется автоматически, без дополнительного подтверждения со стороны пользователя.



Внимание!

При активации принудительное обновление конфигурации не выполняется. После импорта требуется выполнить принудительное обновление конфигурации вручную на каждом экземпляре, только после этого изменения вступят в силу.

Импортированная коллекция автоматически актуализируется при изменении сущностей, которые в нее входят. При удалении сущности она также удаляется из импортированной коллекции.

Импортированная коллекция может быть сохранена в файл на компьютере пользователя. При импорте коллекции выполняется проверка (по уникальному идентификатору) на существование сущностей на экземпляре:

- если сущность уже есть на экземпляре и входит в другую коллекцию, то активация невозможна;
- если сущность есть, она не привязана ни к какой коллекции и ее параметры отличаются от сущности из коллекции, то активация невозможна;
- если сущность есть, она не привязана ни к какой коллекции и ее параметры совпадают с сущностью из коллекции, то активация коллекции возможна только при подтверждении пользователя; такие сущности отображаются пользователю как дополнительная информация

и в случае активации привязываются к активируемой коллекции.

Сущности, которые входят в эту же коллекцию (импортированную ранее) и параметры которых отличаются от сущностей из активируемой коллекции, отображаются пользователю в двух вариантах: в существующем на экземпляре и из активируемой коллекции. При наличии таких сущностей активация возможна только при явном подтверждении пользователя.



Примечание.

Пользователь дает подтверждение относительно всей коллекции.

Сущности, которые не входят в активируемую коллекцию и входят в активную версию коллекции, после активации будут отвязаны от коллекции. При наличии таких сущностей для активации не требуется подтверждение пользователя.

Для экспорта коллекции выберите коллекцию и нажмите на кнопку **Экспорт** – файл в формате JSON будет сохранен на локальном компьютере.

Для импорта коллекции:

1. Нажмите на кнопку **Импортировать** – откроется окно импорта коллекции (см. [Рис. 29](#)).
2. Выберите файл JSON, содержащий коллекцию, и нажмите на кнопку **Импортировать**.

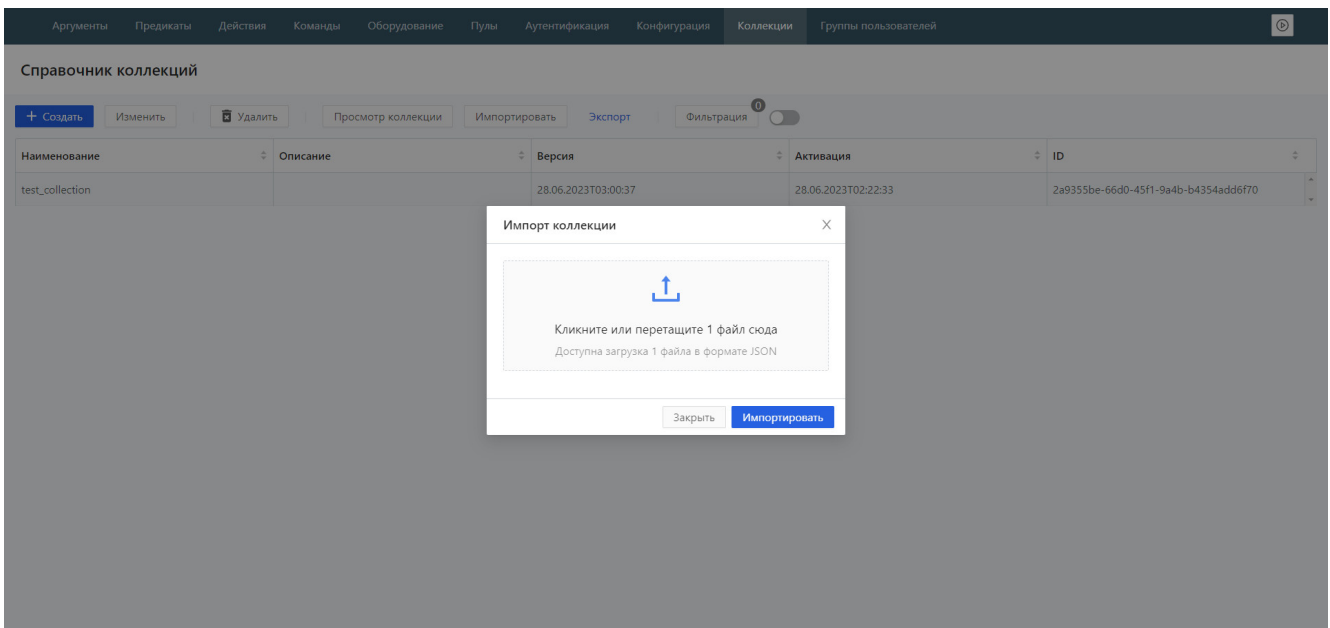


Рис. 29. Импорт коллекции

5.9.4. Удаление коллекции

Для удаления коллекции выберите коллекцию, которую требуется удалить, и нажмите на кнопку **Удалить**, после чего подтвердите действие в открывшемся окне.

5.10. Группы пользователей

Разграничение прав на выполнение команд реализуется с помощью справочника «Группы пользователей».

Права на выполнение команд назначаются путем назначения для них групп пользователей. Группе пользователей назначаются группы Active Directory. В случае если включена

аутентификация, по умолчанию выполнение команд запрещено.

Данные о группах пользователей содержатся в справочнике (см. [Рис. 30](#)), который открывается с помощью пункта **Группы пользователей** главного меню.

Имя	Описание	Код	Группы AD	Встроенная
Execute all commands	A group of users who are unconditionally allowed to execute all commands	allowedToExecuteAllCommands		Да
Execute commands	A group of users who are allowed to execute commands	allowedToExecute		Да

Рис. 30. Справочник групп пользователей

Справочник содержит встроенные группы пользователей:

- `allowedToExecute` – в группу будут автоматически включаться группы Active Directory, которые заданы для роли на выполнение; может быть назначена командам;
- `allowedToExecuteAllCommands` – в группу могут быть включены группы Active Directory; автоматически назначается всем командам.

Для создания новой группы пользователей:

1. Нажмите на кнопку **Создать**.
2. В открывшемся окне «Создание группы» (см. [Рис. 31](#)):
 - введите имя, описание и код группы пользователей;
 - выберите в раскрывающемся списке в поле **Группы AD** группы Active Directory, к которым будет относиться создаваемая группа;
 - нажмите на кнопку **Сохранить**.

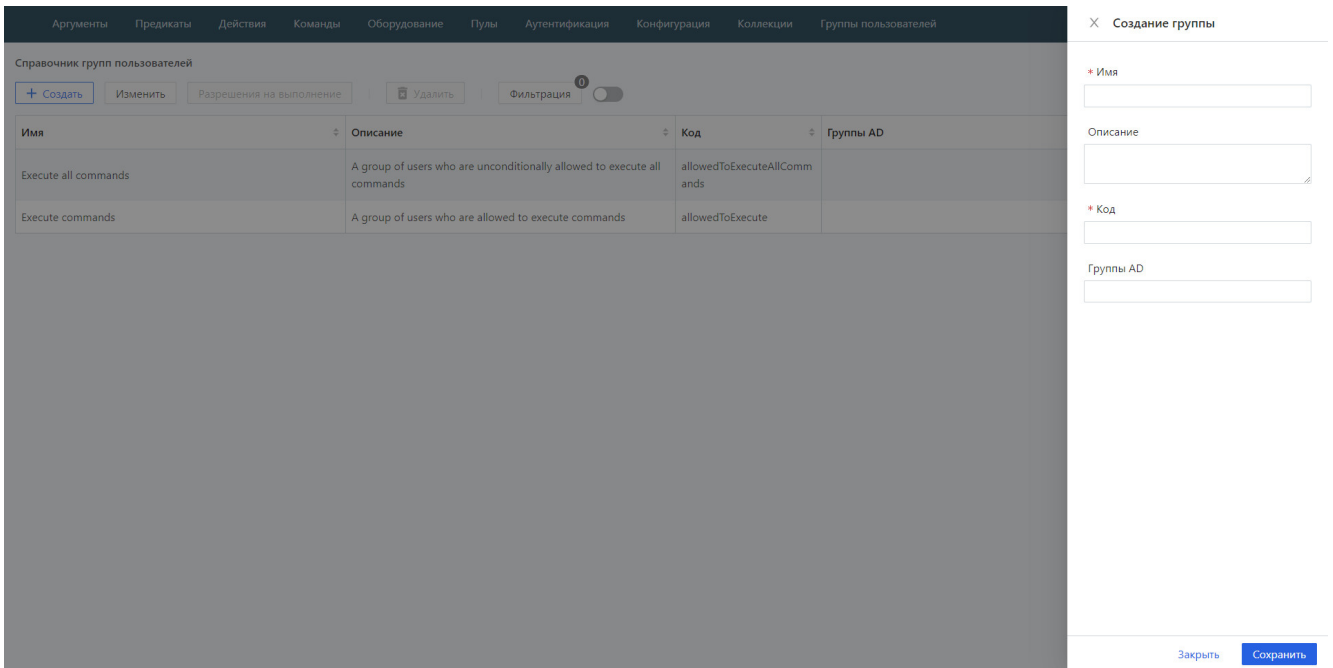


Рис. 31. Создание группы пользователей

5.11. Режим эмуляции

В ответ на запрос выполнения команды в режиме эмуляции всегда возвращается успешный статус. При этом команда на выполнение и запросы на оборудование не отправляются.



Примечание.

Режим эмуляции не влияет на выполнение команды в режиме отладки.

Для включения режима эмуляции установите значение `true` в параметре `emulationMode.enabled` в локальной конфигурации (см. [Настроечные параметры SAM](#)).

Если SAM запущен в режиме эмуляции, в интерфейсе в правом верхнем углу отображается пиктограмма красного цвета.

6. Журналирование

Для изменения уровня журналирования отредактируйте конфигурационный файл `/config/logback.xml`: задайте параметры для обработчиков запросов на журналирование и уровень журналирования для соответствующего обработчика.

Далее описаны правила ведения журналов (log-файлов), заданные через конфигурационный файл `/config/logback.xml`.

В указанном файле описаны:

- `logger` – обработчики запросов на журналирование;
- `appender` – получатели сообщений.

Обработчики запросов на журналирование

В описании обработчика (`logger`) указаны получатель сообщения (`appender`) и уровень журналирования.

В SAM используются следующие уровни журналирования:

- `TRACE` – самый глубокий уровень, трассировочная информация;
- `DEBUG` – отладочные сообщения;
- `INFO` – информационные сообщения;
- `WARN` – предупреждающие сообщения;
- `ERROR` – сообщения об ошибках, возникающих при выполнении каких-либо задач.

Приоритеты уровней журналирования перечислены в порядке возрастания.

Уровень журналирования корневого обработчика (`root`) задает общий уровень журналирования системы. Уровень журналирования, заданный для обработчика, распространяется и на его потомков.



Внимание!

Не задавайте низкоприоритетный уровень журналирования без особой необходимости, так как при большой нагрузке SAM генерирует большое количество журналов и записей в каждый журнал. Это заметно замедляет работу продукта, а также увеличивает расход памяти.

Получатели сообщений

В поставляемом файле `logback.xml` описаны следующие получатели сообщений (`appender`):

- `fileAppender` – ведёт основную часть журналов приложений; ведёт запись в файл `logs/<имя приложения>/<имя приложения>.log`;
- `auditFileAppender` – ведёт журнал действий пользователей; ведёт запись в файл `audit.log`;
- `springFileAppender` – ведёт журнал Spring Framework; ведёт запись в файл `spring.log`;
- `jettyFileAppender` – ведёт журнал контейнера сервлетов Jetty; ведёт запись в файл `jetty.log`.

В описании получателей сообщений (`appender`) указаны:

- `<file>` – файл для вывода текущих сообщений;
- `<rollingPolicy>` – условия ротации журнальных файлов:
 - `<fileNamePattern>` – формат имени журнального файла после ротации, например:

`${catalina.base}/logs/${contextName}/${contextName}-%d{yyyy-MM-dd}-%i.log.zip`, где:

- `yyyy-MM-dd` – дата (например, 2019-10-25);
- `<maxHistory>` – количество дней/часов, по истечении которых журнальные файлы будут удалены; значение по умолчанию – 30;
- `<maxFileSize>` – максимальный размер журнального файла; значение по умолчанию – 100 МБ;
- `<encoder>` – правила кодирования записей:
 - `<pattern>` – шаблон для записи сообщения в файл; пример шаблона:

```
[%d{yyyy.MM.dd HH:mm:ss.SSS}] [%-5level] %message%n
```

где:

- `yyyy.MM.dd HH:mm:ss.SSS HH:mm:ss.SSS` – дата и время создания записи;
- `-5level` – уровень журналирования.

Подробнее о правилах конфигурации журналирования см. <http://logback.qos.ch/manual/configuration.html>.

7. Мониторинг

SAM предоставляет метрики по протоколу HTTP в формате Prometheus.

В мониторинге используются группы метрик:

- основные показатели:
 - `sam_generalInfo_version`;
 - `sam_generalInfo`;
- показатели действий (`sam_actions`);
- показатели команд (`sam_commands`);
- показатели предикатов (`sam_predicates`);
- показатели оборудования (`sam equipments`);
- показатели пулов:
 - `sam_pools_poolProperties`;
 - `sam_pools`.

Для включения мониторинга установите значение `true` в параметре `monitoring.enabled` в локальной конфигурации (см. [Настроечные параметры SAM](#)).