Table of Contents

# Общая информация

## О продукте

### Назначение

Factor.Block (система) — инструмент, реализующий практику OSA и позволяющий блокировать загрузку компонентов с открытым исходным кодом, не отвечающих установленным в системе политикам безопасности.

Factor.Block предоставляет сервис по предотвращению атак на цепочку поставок ПО через компоненты с открытым исходным кодом.

### Состав

Система состоит из следующих частей:

* **Factor.Block** — приложение, состоящее из нескольких модулей, разворачиваемых локально, включая серверную (factor-block-backend) и клиентскую (factor-block-frontend) части, а также базу данных приложения (factor-block-db).
* **Factor.Plugin** — плагин, подключаемый к менеджеру репозиториев Nexus Repository Manager или JFrog Artifactory.
* **Factor.Feed** — облачное хранилище, содержащее информацию об уязвимостях, атрибутах компонентов и лицензиях.

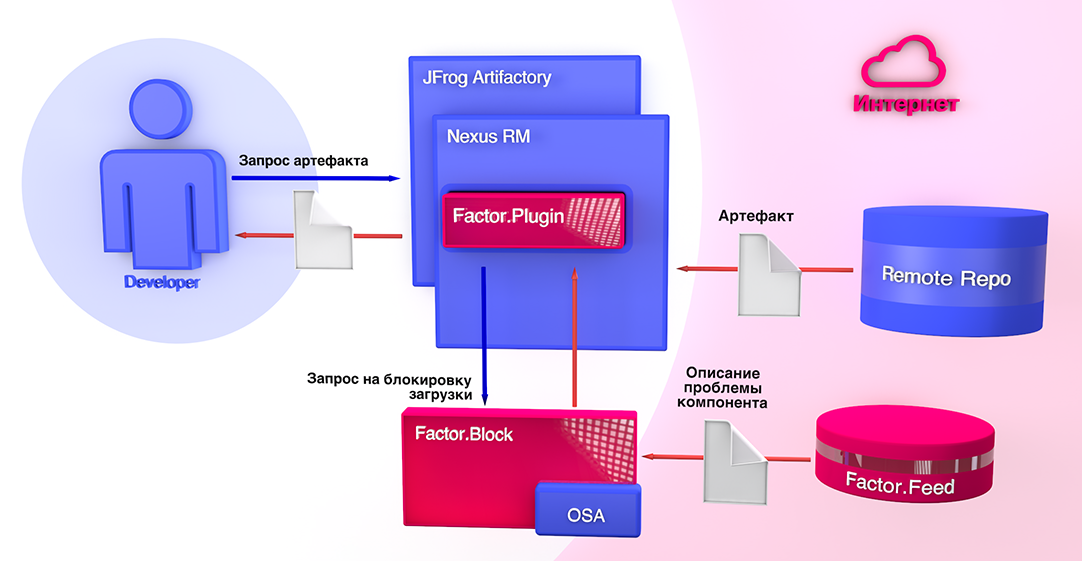
### Описание

Factor.Block анализирует сторонние компоненты с открытым исходным кодом, не проводя при этом активного сканирования этих компонентов.

Система имеет следующие особенности реализации:

* Собственная проприетарная база вредоносных компонентов.
* Рекомендации по выбору безопасной версии компонента.
* Интеграция с менеджером репозиториев с помощью установки плагина не требует изменения существующих CI/CD процессов.

Архитектура системы показана на рисунке ниже.



После обращения разработчика к менеджеру репозиториев на скачивание компонента с открытым исходным кодом включается механизм проверки данного компонента с помощью системы.

Перед возвращением запрошенного компонента разработчику срабатывает плагин Factor.Plugin. Он посылает в Factor.Block, где уже были настроены политики безопасности, запрос на анализ компонента, включающий имя пакета, его хеш и формат. С этими атрибутами Factor.Block обращается в базу компонентов и их уязвимостей — Factor.Feed и применяет к полученным результатам текущие политики безопасности. В случае нарушения политик компонент не будет скачан, а разработчик, запросивший скачивание, получит сообщение об ошибке.

Дальнейшие действия со стороны разработчика, а также возможные решения и шаги в отношении заблокированного компонента описаны в «[Руководстве пользователя](./working%20in%20rm.md)».

### Функциональные возможности

#### Анализ OSS-компонентов на наличие известных уязвимостей

Система осуществляет проверку компонентов с открытым исходным кодом на наличие в них известных уязвимостей. Это позволяет не допустить включения в состав разрабатываемых приложений уязвимых компонентов и значительно снизить риски безопасности.

#### Защита от вредоносных OSS-компонентов

Проверка компонентов с открытым исходным кодом позволяет также выявлять вредоносное ПО — malware или библиотеки с protestware, опубликованные после 24 февраля 2022 года, и исключить его из состава разрабатываемых приложений.

#### Гибкая настройка политик безопасности

Factor.Block предоставляет возможность формировать политики безопасности и применять их к анализируемым компонентам с открытым исходным кодом. Политики в системе могут быть двух типов — блокирующие загрузку и не блокирующие. При необходимости отдельные компоненты могут быть включены в список исключений, что позволяет гибко и оперативно управлять процессом разработки ПО.

#### Блокировка загрузки компонента при обнаружении проблем на этапе OSA

Каждый загружаемый пакет проверяется на нарушение политик безопасности, прежде чем он будет доступен для скачивания разработчиком. Система автоматически блокирует загрузку компонента разработчиком при обнаружении проблем на этапе OSA согласно настройкам политик.

#### Поддержка ключевых менеджеров репозиториев

Factor.Block работает с ключевыми менеджерами репозиториев:

* **Nexus Repository Manager**.
* **JFrog Artifactory**.

## Термины и сокращения

|  |  |
| --- | --- |
| Сокращение | Определение понятия |
| OSA (Open Source Analysis)Анализ ПО с открытым исходным кодом | Анализ библиотек и компонентов с открытым исходным кодом, которые не входят в периметр разработки программного обеспечения, на наличие известных уязвимостей и на лицензионную чистоту. Помогает удостовериться, что используемые библиотеки безопасны (в них не содержится известных уязвимостей) и что согласно политике их распространения их можно использовать в разрабатываемых продуктах (лицензионная чистота) |
| OSS (Open-Source Software)ПО с открытым исходным кодом | Исходный код ПО с открытым исходным кодом доступен для просмотра, изучения, изменения и использования для создания новых программ через заимствование исходного кода, если это позволяет совместимость лицензий |
| PURL (Persistent Uniform Resource Locator) | Постоянный единообразный локатор (определитель местонахождения) ресурса |
| ПО | Программное обеспечение |

# Руководство по установке и интеграции

## Требования к инфраструктуре

Установленные для эксплуатации Factor.Block технические средства должны быть совместимы между собой и поддерживать сетевой протокол TCP/IP.

### Серверное оборудование

Рекомендуемые технические характеристики серверного оборудования:

* Процессор: 2×CPU 2 ГГц.
* Оперативная память: 8 Гб.
* Свободное дисковое пространство: 20 Гб.

### ОС

Развертывание и функционирование системы осуществляется на ОС Linux. Поддерживаются следующие операционные системы.

|  |  |  |
| --- | --- | --- |
| Операционная система | Архитектура | Платформа |
| Linux | 64-bit | Ubuntu Server 18.04.6 x64 |
| Linux | 64-bit | CentOS/RHEL 7 и выше |
| Linux | 64-bit | Astra Linux SE 1.7 (Смоленск, Воронеж) |

### Прочее ПО

Для обеспечения функционирования система должна интегрироваться с менеджером репозиториев Nexus Repository Manager или JFrog Artifactory. Поддерживаются следующие версии менеджеров репозиториев.

|  |  |
| --- | --- |
| Менеджер репозиториев | Версия |
| Nexus Repository Manager | 3.20.0 и выше |
| JFrog Artifactory | 7.31.10 и выше |

Для установки и функционирования системы необходимо установить из стандартных репозиториев **Docker** и **Docker-compose**.

|  |  |
| --- | --- |
| Пакет | Версия |
| Docker | 20.10.0 и выше |

Docker-compose|1.29.2 и выше

## Подготовка инфраструктуры

### Установка Docker и Docker-compose

1. Обновите на сервере пакеты окружения.
2. Установите последнюю версию **Docker**. При необходимости обратитесь к [документации](https://docs.docker.com/install/linux/docker-ce/ubuntu/) на официальном сайте Docker.
3. Запустите **Docker**.
4. Убедитесь, что **Docker** успешно запущен.
5. Для запуска **Docker** после перезагрузки сервера используйте следующую команду.
6. Проверьте версию **Docker**.

### Установка Docker-compose

1. Установите **Docker-compose**. При необходимости обратитесь к [документации](https://docs.docker.com/compose/install/) на официальном сайте Docker.
2. Проверьте версию **Docker-compose**.

=== "Astra"  
 docker-compose --version

## Требования к рабочему месту пользователя

### Требования к оборудованию

Используются персональные компьютеры с операционной системой Windows, Linux или MacOS.

Рекомендуемые технические характеристики клиентских компьютеров:

* Процессор: не ниже 2 ГГц.
* Оперативная память: не менее 8 Гб.
* Свободное дисковое пространство: 10 Гб (+ размер прикладных метаданных).

### Требования к программному обеспечению

Должен быть установлен один из следующих браузеров:

* Google Chrome версии 89 и выше.
* Mozilla Firefox версии 50 и выше.
* Microsoft Edge версии 89 и выше.
* Safari версии 10 и выше.

В настройках браузера должно быть разрешено использование «cookies».

## Комплект поставки системы

В комплект поставки системы входят:

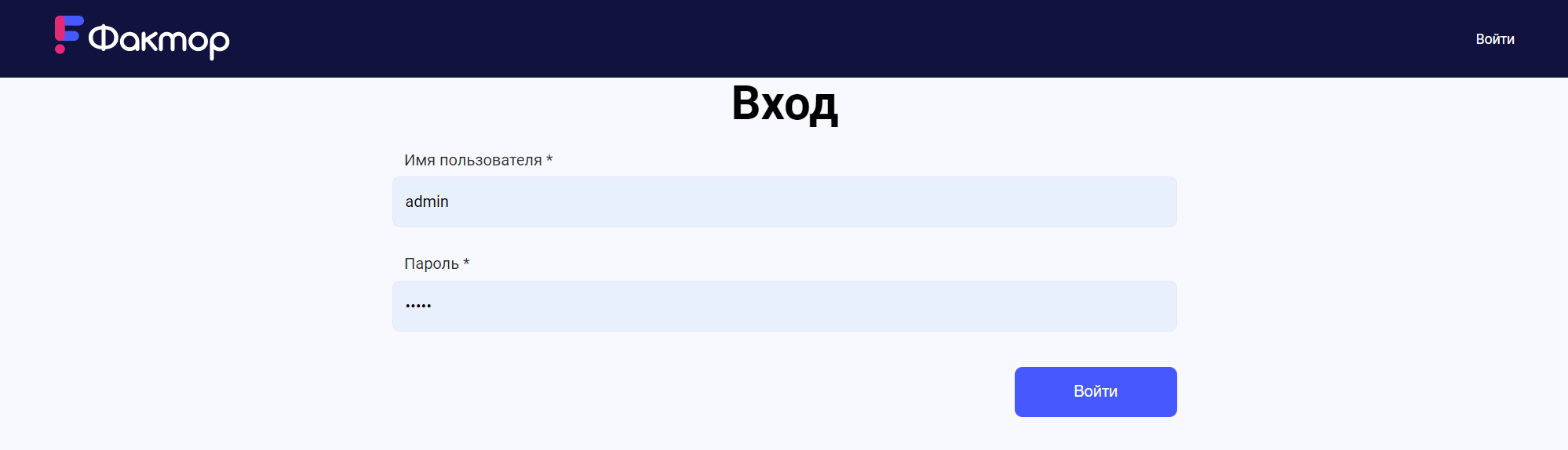
* Дистрибутив Factor.Block в виде zip-архива и трех Docker-образов (factor-block-backend, factor-block-frontend, а также база данных приложения — factor-block-db) или доступ к docker-репозиторию docker.swordfishsecurity.com для загрузки docker-контейнеров приложения.
* Плагин Factor.Plugin для Nexus Repository Manager / JFrog Artifactory.
* Сетевой доступ к базе данных Factor.Feed (https://feed.swordfishsecurity.com/).
* Токен для подключения к Factor.Feed.

## Установка

### Установка Factor.Block

??? note “Прежде чем приступить к установке, скопируйте zip-архив дистрибутива на сервер.”

1. Создайте папку для установки **Factor.Block** (рекомендуется — ***/opt/factor-block/***).
2. Распакуйте архив в созданную папку.
3. Перейдите в директорию **Factor.Block**.
4. Выполните скрипт подготовки окружения.
5. Загрузите образы в **Docker**. Предусмотрена возможность загрузки образов как из репозитория docker.swordfishsecurity.com, так и из дистрибутива.
6. Для проверки результатов предыдущих шагов выполните следующую команду:
7. В случае необходимости переопределения портов или других настроек отредактируйте файл переменных окружения ***.env***, содержащий в том числе пароли по умолчанию и версии используемых серверной и клиентской частей, а также базы данных Factor.Block, и конфигурационный файл ***docker-compose.yml***.
8. Запустите контейнеры Factor.Block.
9. Перейдите по адресу сервера, на котором развернут Factor.Block, по адресу, включающему фрагмент (например, ), и укажите данные пользователя и настройки подключения к базе данных, а также URL-адрес Factor.Feed:
10. После проверки подключения с помощью кнопки **Check** необходимо сохранить настройки с помощью кнопки **Save**. Система перезагрузится и будет готова к использованию.
11. Перейдите в пользовательский интерфейс Factor.Block по адресу хоста (например, ). Логин и пароль пользователя системы по умолчанию: admin/admin. Процедура смены пароля описана в Руководстве пользователя в разделе «[Изменение пароля пользователя](#X04975045cb7b4cdf85cd6d960b21f288297aa73)».



### Установка Factor.Plugin

Прежде чем приступить к использованию системы, ее необходимо интегрировать с менеджером репозиториев Nexus Repository Manager/JFrog Artifactory, используя входящий в комплект поставки Factor.Plugin.

#### Установка плагина в Nexus Repository Manager

!!! note “Примечание” Плагин совместим с Nexus Repository Manager версии 3.20.0 и выше.

1. KAR-файл плагина, входящий в комплект поставки, необходимо скопировать в директорию ***deploy*** на сервере с Nexus Repository. Если Nexus Repository развернут внутри контейнера, полный путь к этой директории выглядит так: ***/opt/sonatype/nexus/deploy***. Подробная инструкция по установке плагинов приведена в [документации Nexus Repository Manager](https://sonatype-nexus-community.github.io/nexus-development-guides/plugin-install.html).
2. После копирования файла плагина откройте в менеджере репозиториев раздел настроек — **Settings**, выберите в меню слева пункт **System** › **Capabilities** и нажмите на кнопку **Create capability**. В списке появляется новый плагин **OSA Firewall Configuration** — нажмите на него.
3. Откроется окно настроек. Сконфигурируйте параметры плагина.
4. После активации плагина любое обращение к компонентам в Nexus Repository будет сопровождаться проверкой со стороны Factor.Block. Дополнительно необходимо активировать проверку данного менеджера репозиториев в Factor.Block:
5. Для проверки корректности подключения репозитория и настройки плагина, вы можете загрузить через Nexus Repository еще какой-нибудь пакет (например: ). История запросов к Factor.Block отображается в системе в разделе «[Лог запросов](./query%20log.md)».

#### Установка плагина в Jfrog Artifactory

Для установки плагина и настройки подключения менеджера репозиториев JFrog Artifactory необходимо выполнить следующие шаги:

1. Полученный плагин в виде zip-архива необходимо распаковать в директорию ***JFROG\_HOME*** — это корневая папка инсталляции JFrog Artifactory. Подробная инструкция по установке плагинов приведена в [документации JFrog Artifactory](https://www.jfrog.com/confluence/display/JFROG/User+Plugins).
2. Отредактируйте конфигурационный файл плагина ***factor-block-plugin.properties*** для настройки его работы в JFrog Artifactory.
3. После установки и редактирования конфигурационного файла плагина необходимо перезапустить JFrog Artifactory.
4. Дальнейшие обновления плагина происходят автоматически по принципу «Live Reload» и регулируются параметром интервала обновления в файле ***$JFROG\_HOME/artifactory/var/etc/artifactory/artifactory.system.properties***, например:
5. Для проверки работоспособности установленного плагина запросите какой-либо компонент (например: ) и убедитесь, что он отображается в Логе запросов Factor.Block.

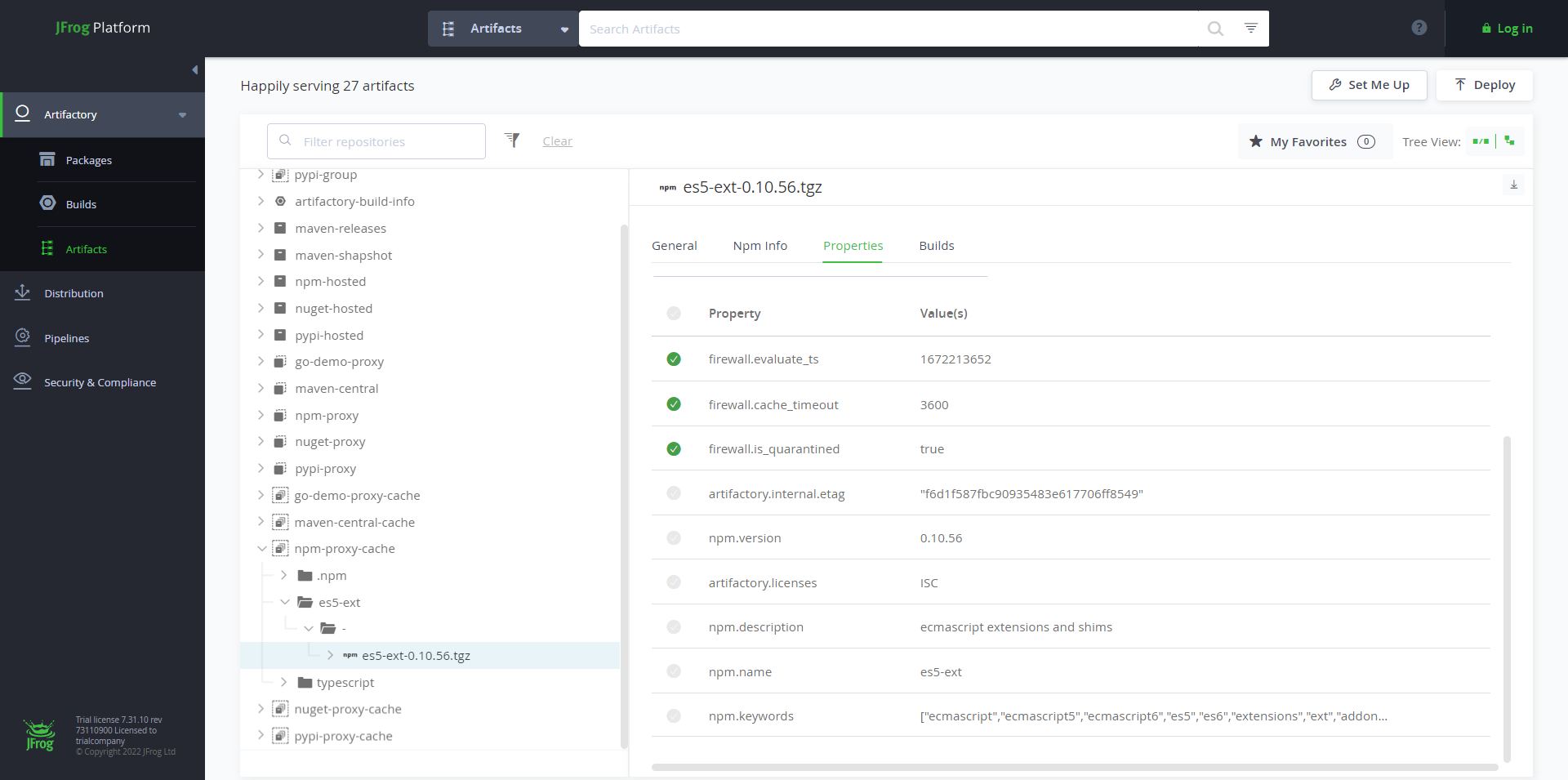
### Подключение Factor.Feed

Для подключения Factor.Feed используется токен, входящий в комплект поставки системы, см. раздел «[Сервис данных](./data%20service.md)». Кроме этого, выбрав в Factor.Block пункт меню **Репозитории**, необходимо указать IP-адрес репозитория (ев) с установленным Factor.Plugin, см. раздел «[Репозитории](./repositories.md)».

## Properties системы в пакетах в менеджерах репозиториев

При работе с пакетами Factor.Plugin добавляет к Properties этих пакетов три новых:

* — отметка времени, когда была проведена последняя проверка пакета.
* — время, на которое был закэширован пакет.
* — статус последней проверки пакета ( — находится в карантине, — не находится в карантине).

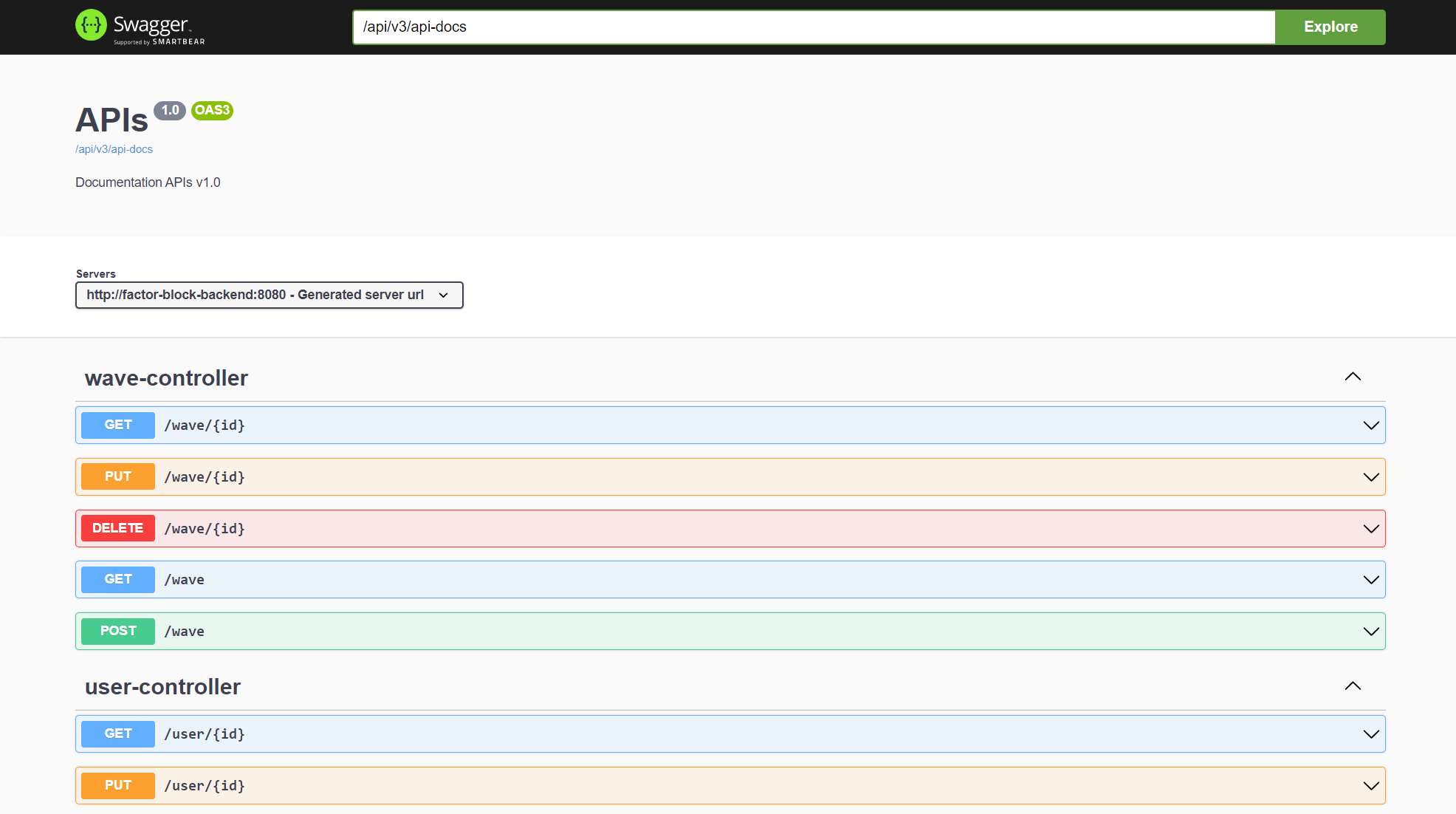


Система добавляет данные Properties при работе с обоими менеджерами репозиториев — Nexus Repository Manager и JFrog Artifactory для увеличения надежности работы и повышения достоверности предоставляемых результатов.

## Интеграционный API

Интеграции Factor.Block с различными инструментами и приложениями могут быть осуществлены с помощью интеграционного REST API системы, описание которого доступно в Swagger.

Чтобы посмотреть API системы, перейдите по [ссылке](https://factor.demo.sca.swordfishsecurity.com/api/webjars/swagger-ui/index.html#/) и в поле **Explore** укажите значение .



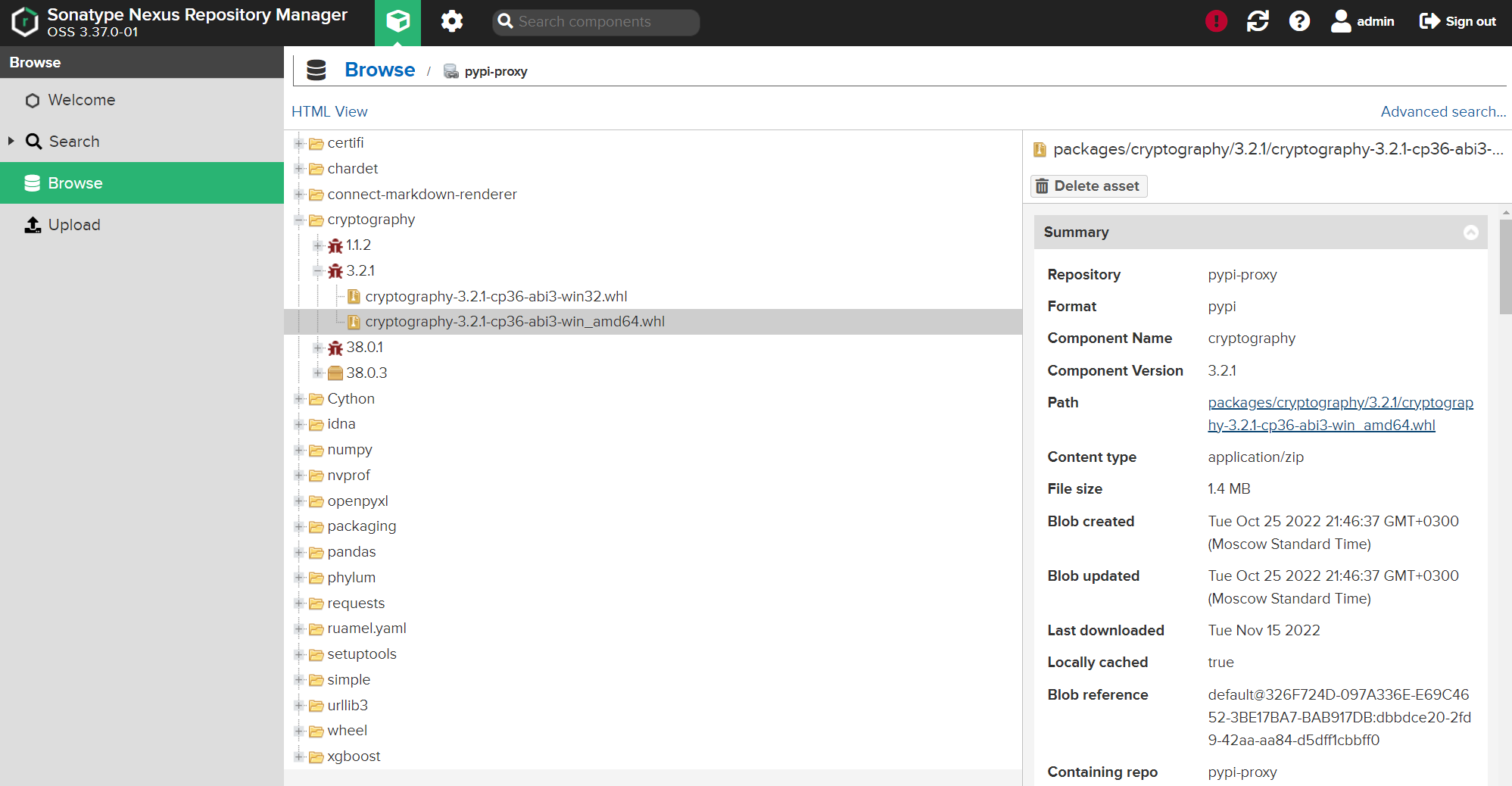
# Руководство пользователя

## Работа в менеджере репозиториев

!!! note “Примечание” Для выполнения всех нижеописанных действий пользователю не требуется аутентифицироваться в системе Factor.Block. Под пользователем в данном разделе понимается разработчик, работающий с менеджером репозиториев Nexus Repository Manager или JFrog Artifactory.

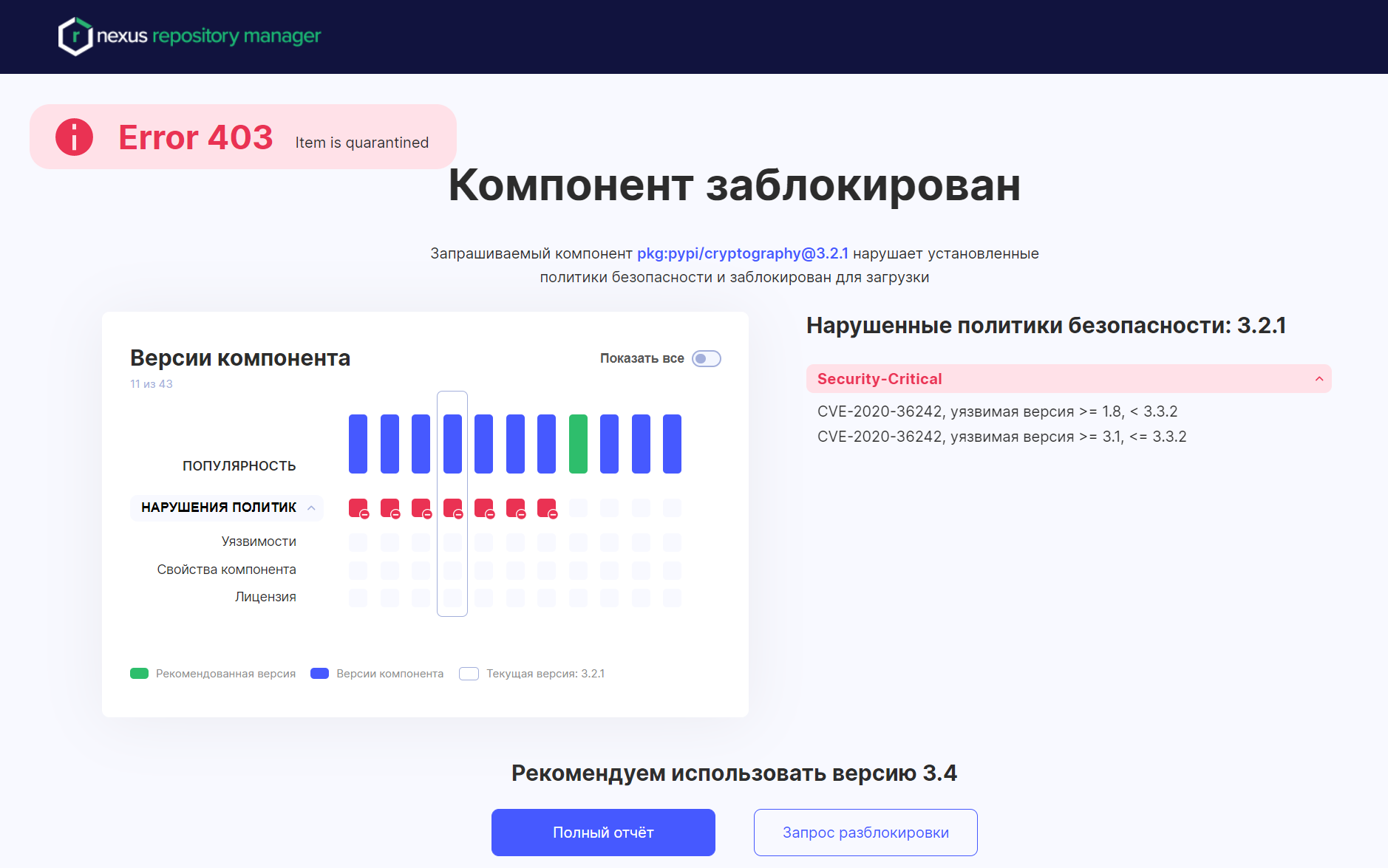
После установки Factor.Block и подключения плагина Factor.Plugin к Nexus Repository Manager или JFrog Artifactory работа пользователя в менеджере репозиториев будет проходить под контролем Factor.Block.

Результат попытки пользователя загрузить компонент из менеджера репозиториев теперь будет зависеть от того, отвечает ли данный компонент установленным в Factor.Block политикам безопасности.



Если выбранный компонент соответствует заданным в политиках безопасности условиям, он будет успешно загружен.

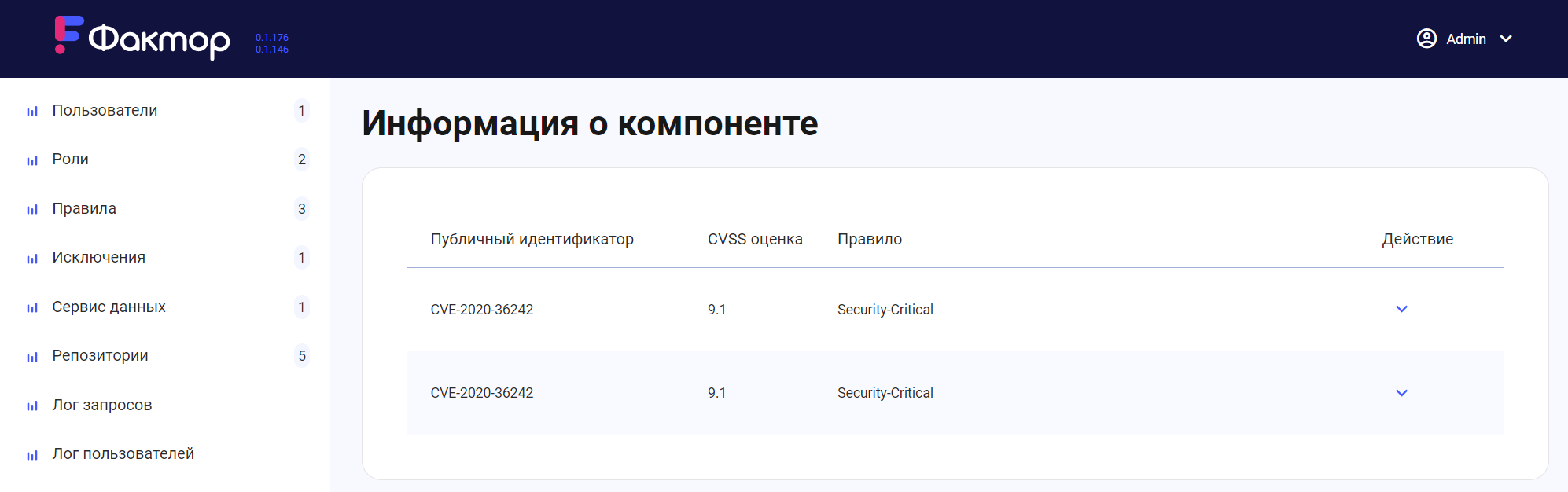
Если выбранный компонент не удовлетворяет заданным в политиках безопасности условиям, например содержит уязвимости, система заблокирует его загрузку и выдаст соответствующее сообщение вместе с информацией о нарушенных политиках безопасности и обнаруженных уязвимостях, а также рекомендации по выбору безопасной версии компонента.



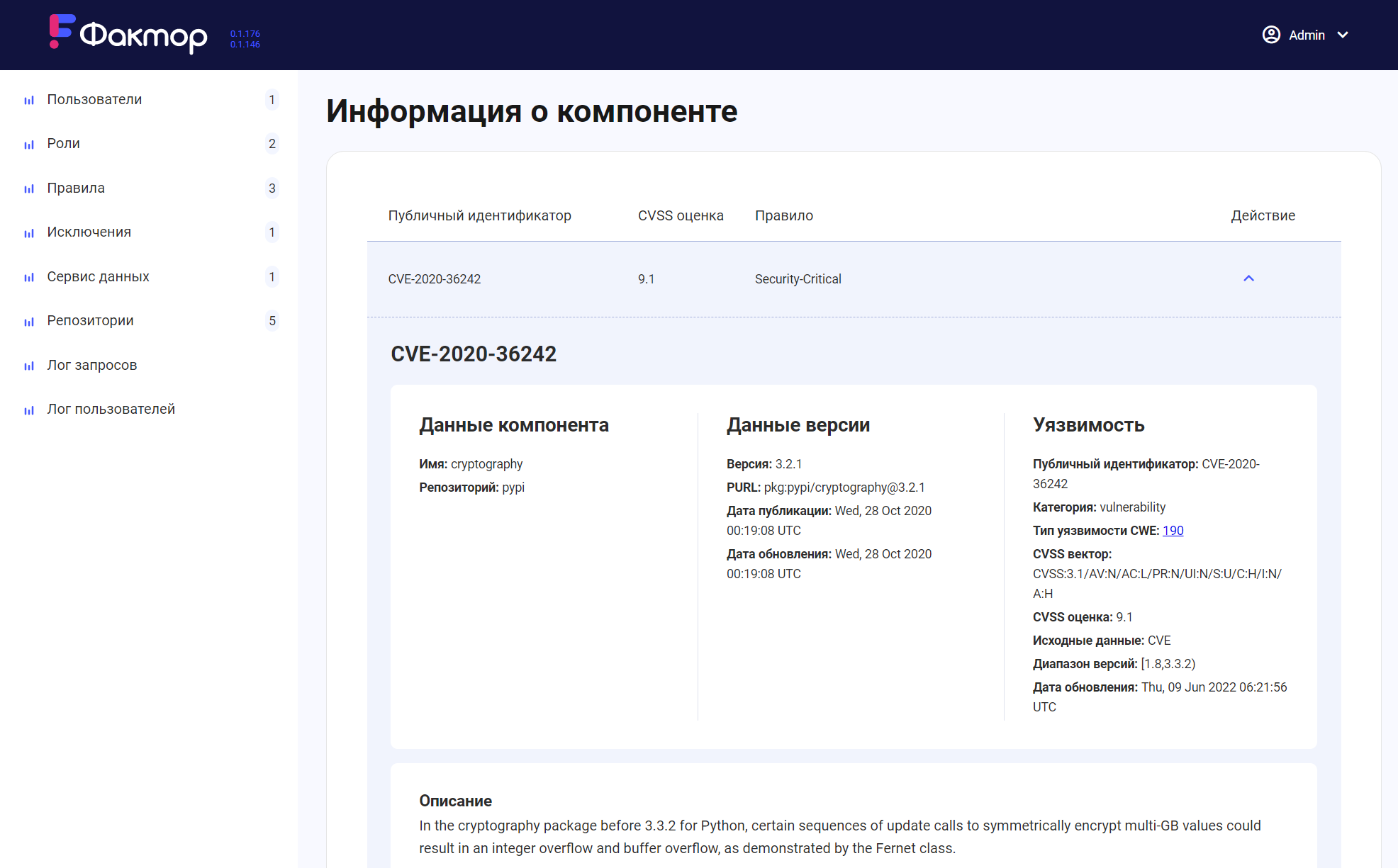
Внизу страницы с сообщением о блокировке компонента расположены две кнопки: **Полный отчет** и **Запрос разблокировки**.

### Полный отчет

При нажатии кнопки **Полный отчет** происходит переход на страницу системы **Информация о компоненте**, содержащей список всех обнаруженных во время проверки компонента уязвимостей и нарушений установленных политик безопасности.



Нажмите на строку уязвимости или нарушения политики безопасности, чтобы получить детальную информацию, включая данные компонента, данные версии, данные уязвимости, описание, объяснение, рекомендации по устранению и полезные ссылки.

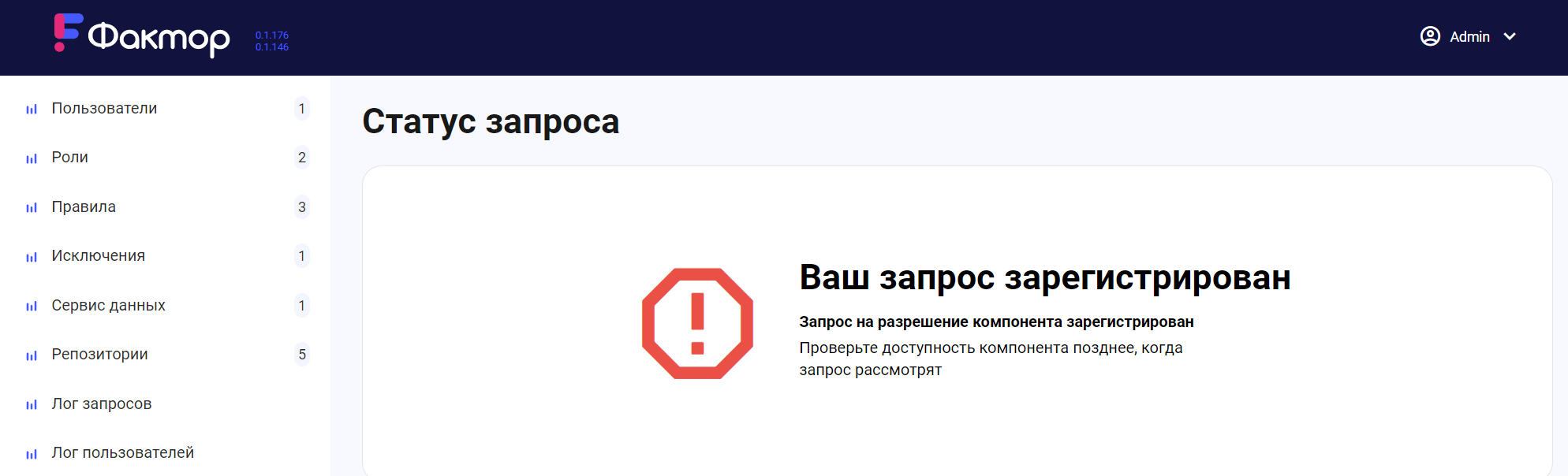


### Запрос разблокировки

В разработке ПО встречаются ситуации, когда не существует небезопасной версии компонента, предоставляющего критическую для проекта функциональность, или необходимо выпустить срочный релиз, в котором задействован небезопасный компонент, или руководство проекта просто принимает решение принять связанные с использованием такого компонента риски безопасности.

В таком случае для решения проблемы с заблокированным системой компонентом пользователю необходимо сделать запрос на его разблокировку. Перед принятием подобного решения рекомендуется тщательно изучить детальную информацию о компоненте и содержащихся в нем уязвимостях и оценить их влияние на функциональность разрабатываемого ПО.

Для разблокировки компонента необходимо нажать кнопку **Запрос разблокировки**. На экране появится сообщение о регистрации запроса в системе.



После запроса пользователю необходимо дождаться, пока в системе не будет разрешено скачивание запрошенного компонента или принято решение о запрете его использования.

### Работа с использованием CLI

Все действия по работе пользователя с менеджером репозиториев могут быть выполнены не только через UI, но и с использованием командной строки или сборщиков. Например, используя команду:

Выполнение стандартной команды по работе с менеджером репозиториев, так же как и при работе через UI, может закончиться как успешной загрузкой компонента, так и сообщением о его блокировке в зависимости от его соответствия заданным в политиках безопасности условиям.

В случае блокировки на экран будут выданы соответствующее сообщение вместе с информацией о нарушенных политиках безопасности и обнаруженных уязвимостях, рекомендации по выбору безопасной версии компонента, а также ссылки на полный отчет и запрос разблокировки.



## Пользователи и роли

### Роли пользователей

В системе нет предопределенных ролей пользователей, но существует гибкий механизм управления ими.

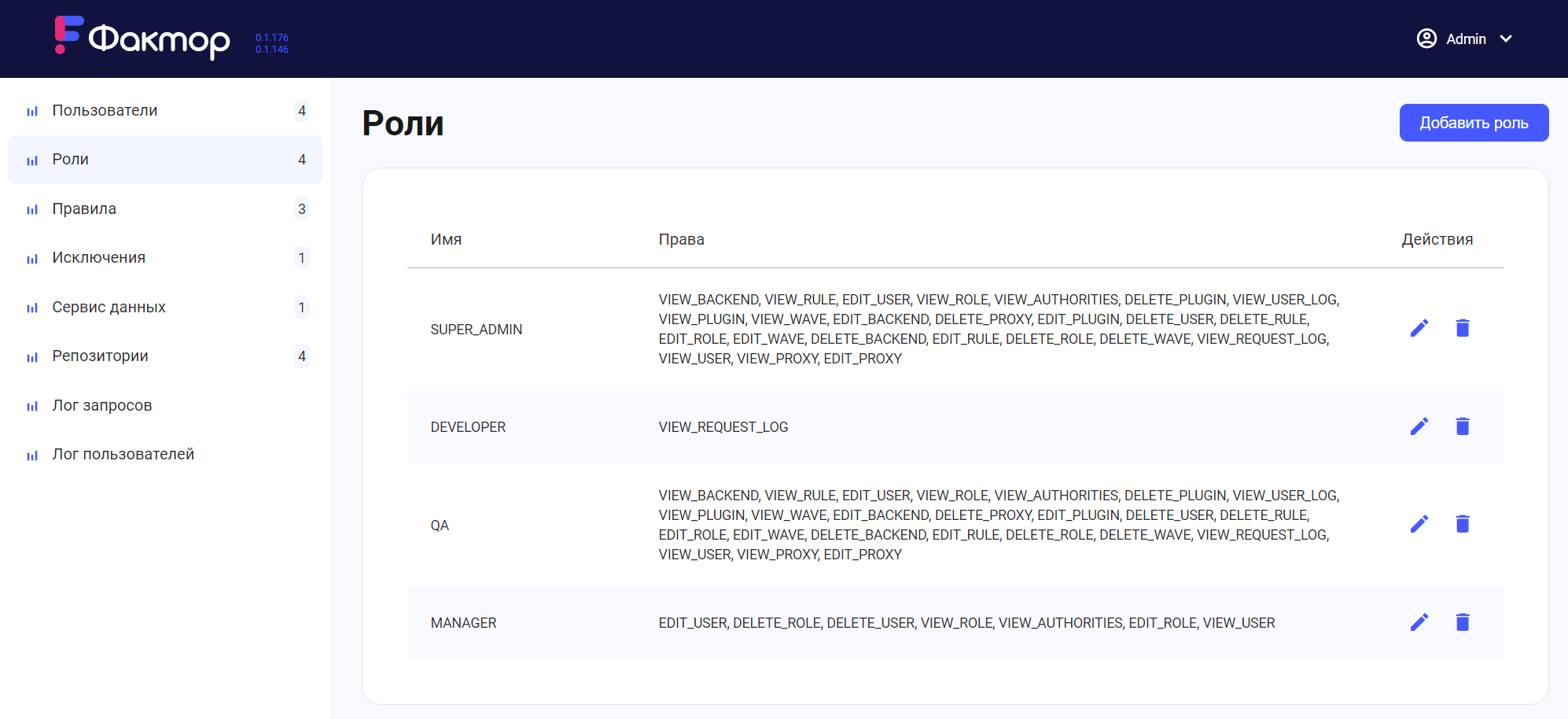
Управление ролями пользователей осуществляется в два этапа:

* Создание и редактирование ролей в системе.
* Присваивание созданных ролей пользователям.

### Создание и редактирование ролей в системе

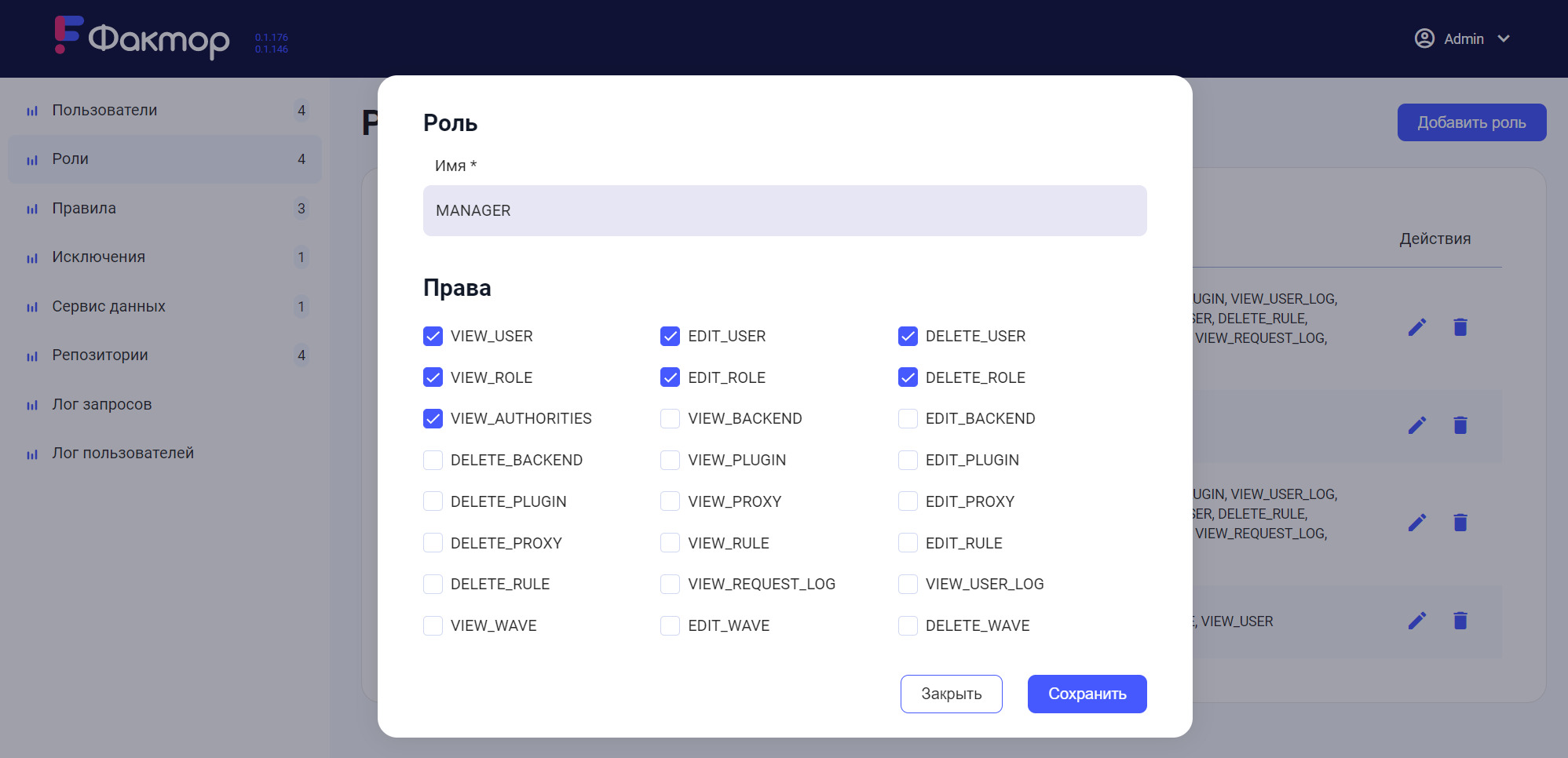
Выберите пункт **Роли** в меню слева, чтобы посмотреть список существующих в системе ролей.

Для каждой роли указаны ее имя и присвоенные ей в системе права. Более подробная информация о правах доступа приведена в [Приложении 1](./appendix1.md).



Права можно отредактировать, нажав на иконку редактирования , расположенную справа в строке роли.

В появившемся окне отметьте необходимые для роли права и нажмите кнопку **Сохранить**.



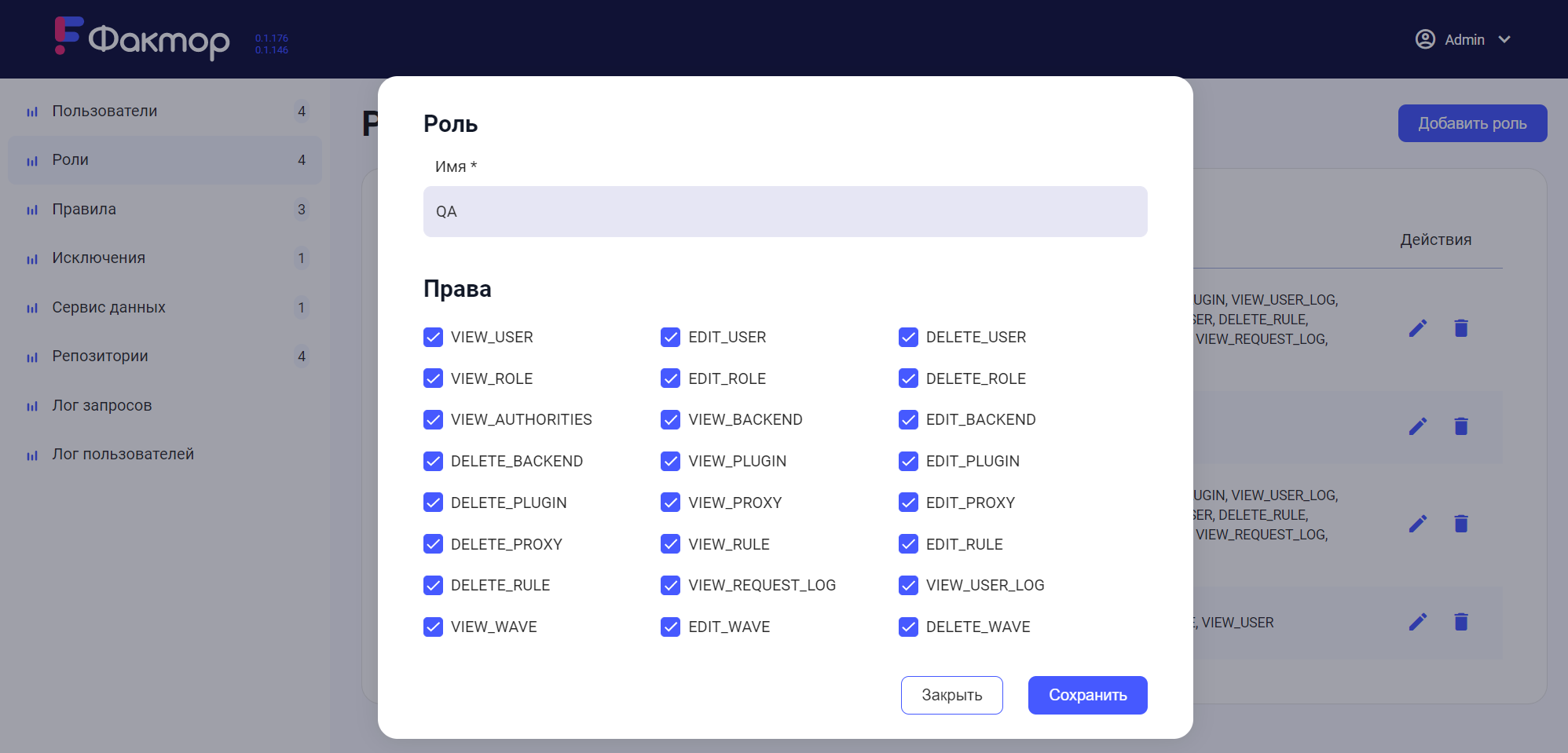
!!! note “Примечание” Отдельные права дают возможность просматривать, редактировать, создавать или удалять объекты определенного типа в системе. Например, для просмотра ролей в системе у пользователя должны быть права VIEW\_ROLE. Для редактирования прав роли у пользователя, редактирующего роль, должны быть права EDIT\_ROLE. Для удаления роли у пользователя, удаляющего роль, должны быть права DELETE\_ROLE.

В случае отсутствия прав на просмотр и совершение операций с объектами определенного типа в системе, пользователь не будет их видеть на соответствующей странице или не будет иметь возможности работать с элементами пользовательского интерфейса, позволяющими производить с этими объектами какие-либо действия.

Например, в случае отсутствия прав на просмотр правил в системе (VIEW\_RULE), пользователь будет видеть пустую страницу **Правила**.

Роль можно удалить, нажав на иконку  в ее строке.

Для создания роли в системе нажмите кнопку **Добавить роль** справа вверху на странице **Роли**. В появившемся окне введите имя новой роли, отметьте необходимые для роли права и нажмите кнопку **Сохранить**.

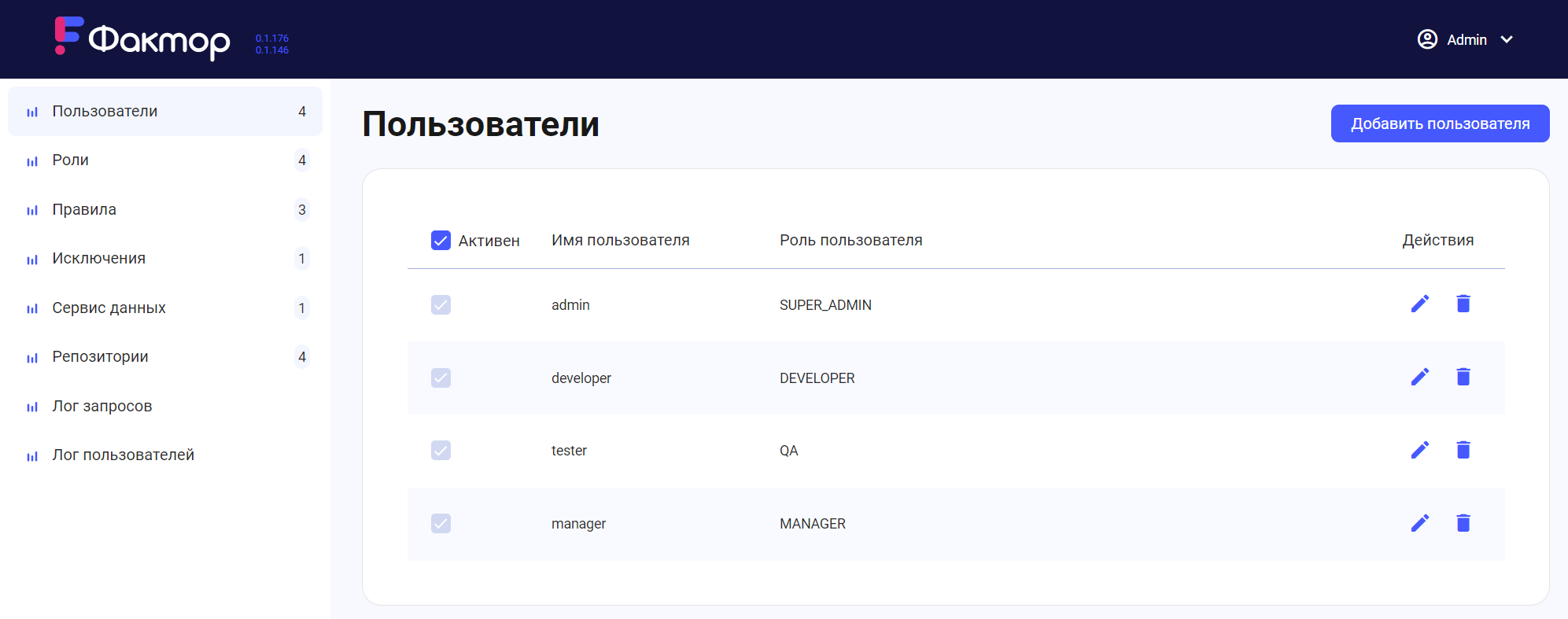


Вновь созданная роль появится в списке ролей в системе.

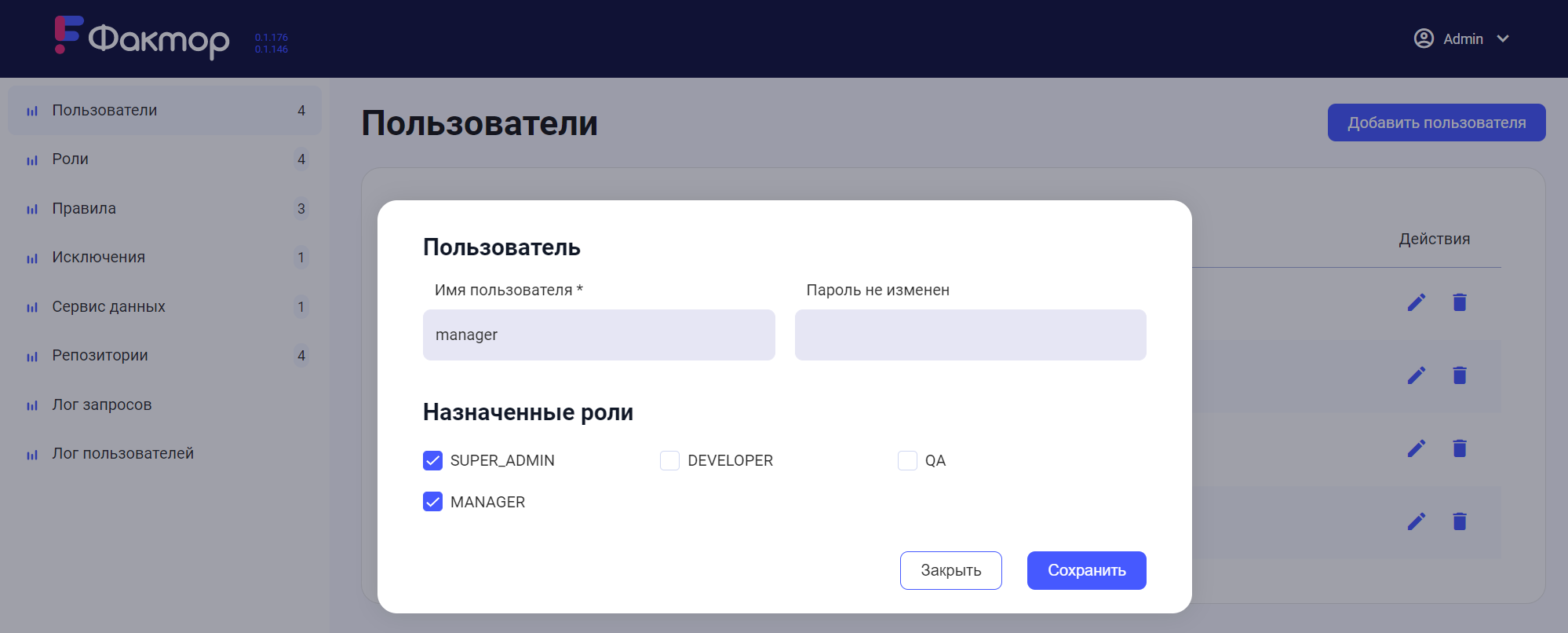
!!! note “Примечание” Имя роли может содержать только английские буквы и символ подчеркивания «\_».

### Присваивание ролей пользователям

Выберите пункт **Пользователи** в меню слева, чтобы посмотреть список существующих в системе пользователей. Для каждого пользователя указаны его статус в системе (активен/не активен), имя и присвоенные ему в системе роли.

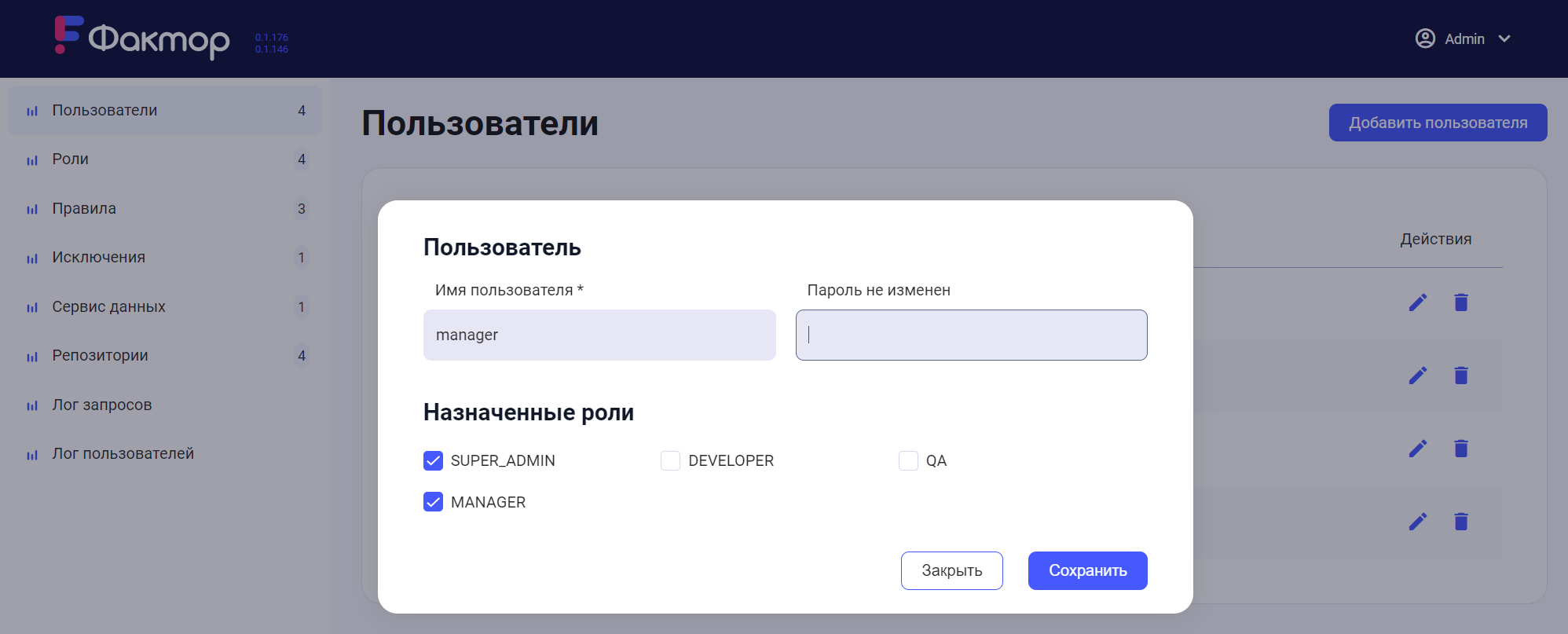


Роль(-и) пользователя в системе можно изменить, нажав на иконку редактирования , расположенную справа в строке пользователя. В появившемся окне отметьте необходимые для пользователя роли и нажмите кнопку **Сохранить**.



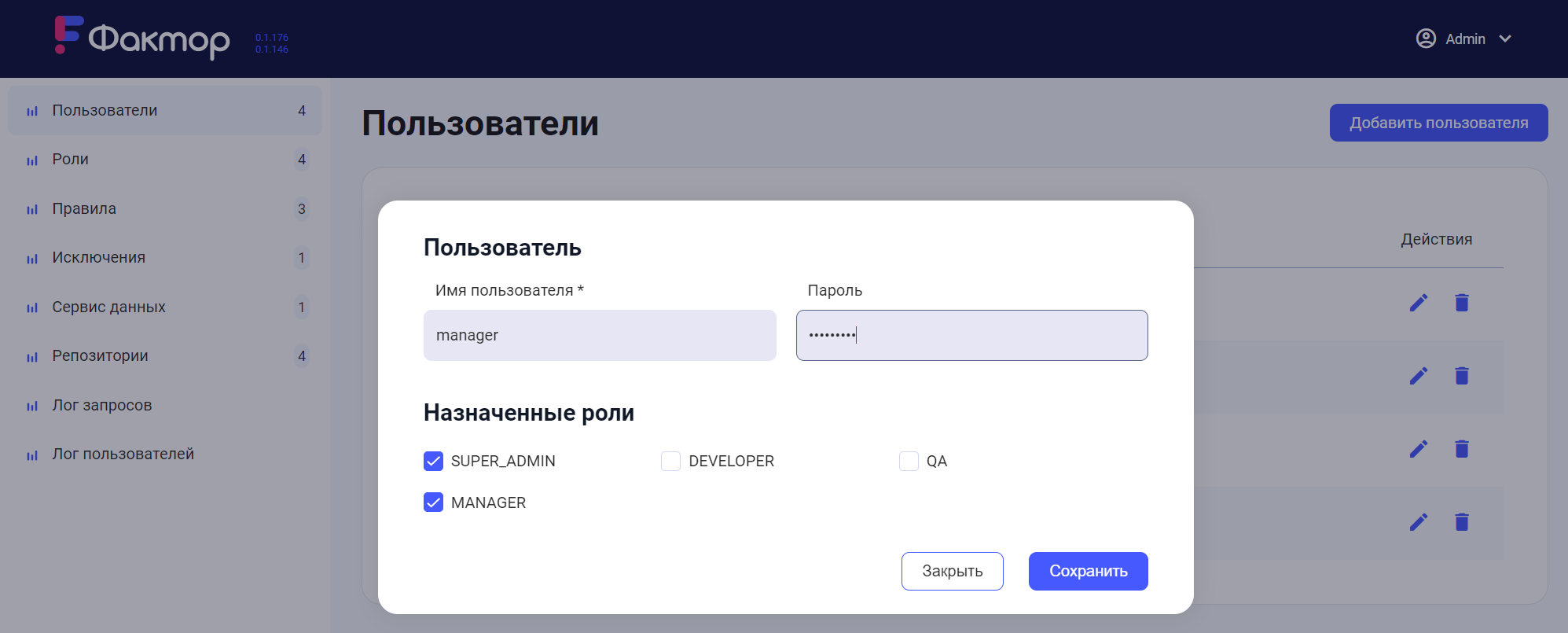
### Изменение пароля пользователя

Пароль пользователя в системе можно изменить, нажав на иконку редактирования  в строке пользователя. В появившемся окне в поле **Пароль не изменен** введите новый пароль, при этом название поля поменяется на **Пароль**. Нажмите кнопку **Сохранить**.



### Добавление пользователя

Чтобы создать нового пользователя в системе, выберите пункт **Пользователи** в меню слева и нажмите кнопку **Добавить пользователя** справа вверху. В появившемся окне введите имя и пароль нового пользователя, отметьте необходимые для пользователя роли и нажмите кнопку **Сохранить**.



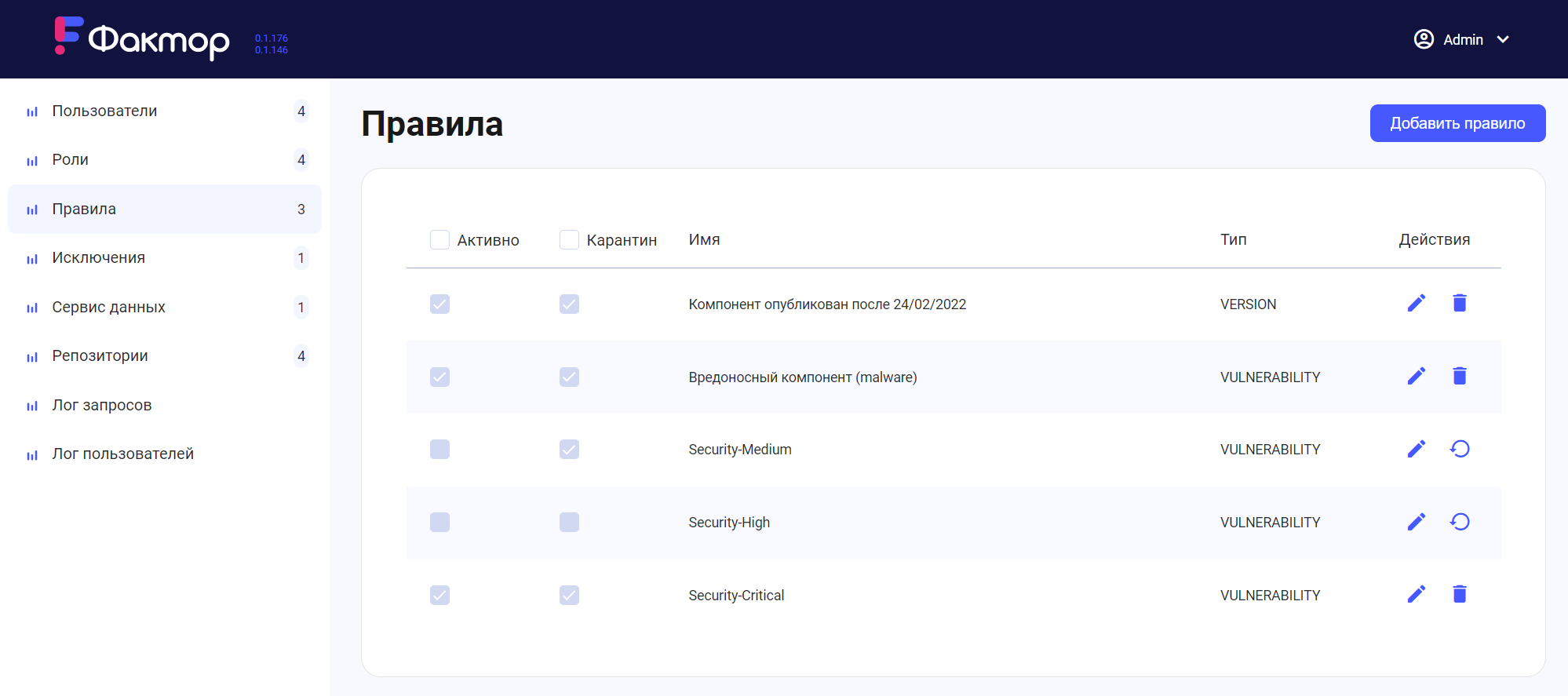
!!! note “Примечание” Имя пользователя может содержать только цифры, английские буквы, а также символы «@», «\_», «-» и «.», что позволяет использовать в качестве имени почтовый адрес.

## Политики безопасности

Политики безопасности в системе можно задавать с помощью правил. Активные правила применяются к загружаемым компонентам в соответствии со сформированными в них условиями проверок.

### Правила

Выберите пункт **Правила** в меню слева, чтобы посмотреть список существующих в системе правил (политик безопасности).



Для каждого правила указаны его статус в системе (активно/неактивно), является ли его нарушение блокирующим для загрузки, имя правила и его тип (проверка по уязвимости или по версии).

Если политика безопасности является блокирующей, компонент не будет загружен и будет помещен в карантин, что отображено в соответствующем поле — **Карантин**. Если политика безопасности не является блокирующей, компонент будет загружен, а запись о нарушении этой политики будет добавлена в лог запросов (см. детали в соответствующем разделе «[Лог запросов](./query%20log.md)»).

В случае нарушения политики безопасности разработчик, работающий с менеджером репозиториев, увидит ее имя в сообщении о блокировке компонента.

Политики безопасности, основанные на проверке компонента на наличие определенной уязвимости, имеют тип **VULNERABILITY**. Политики безопасности, построенные на проверке версии компонента, имеют тип **VERSION**.

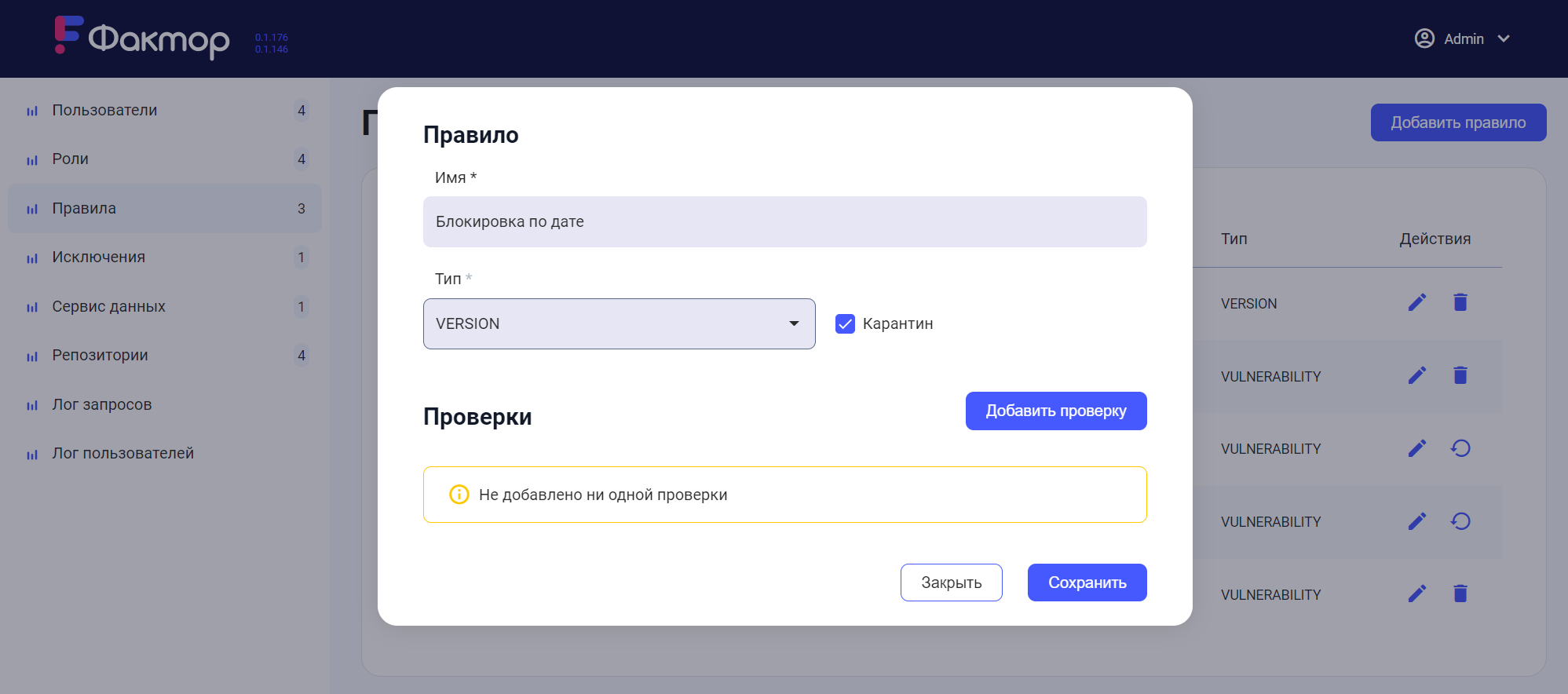
Правило можно отредактировать, нажав на кнопку  в конце его строки.

Неактивное правило можно активировать, нажав на кнопку  в конце его строки. Активное правило можно деактивировать, нажав на кнопку  в конце его строки.

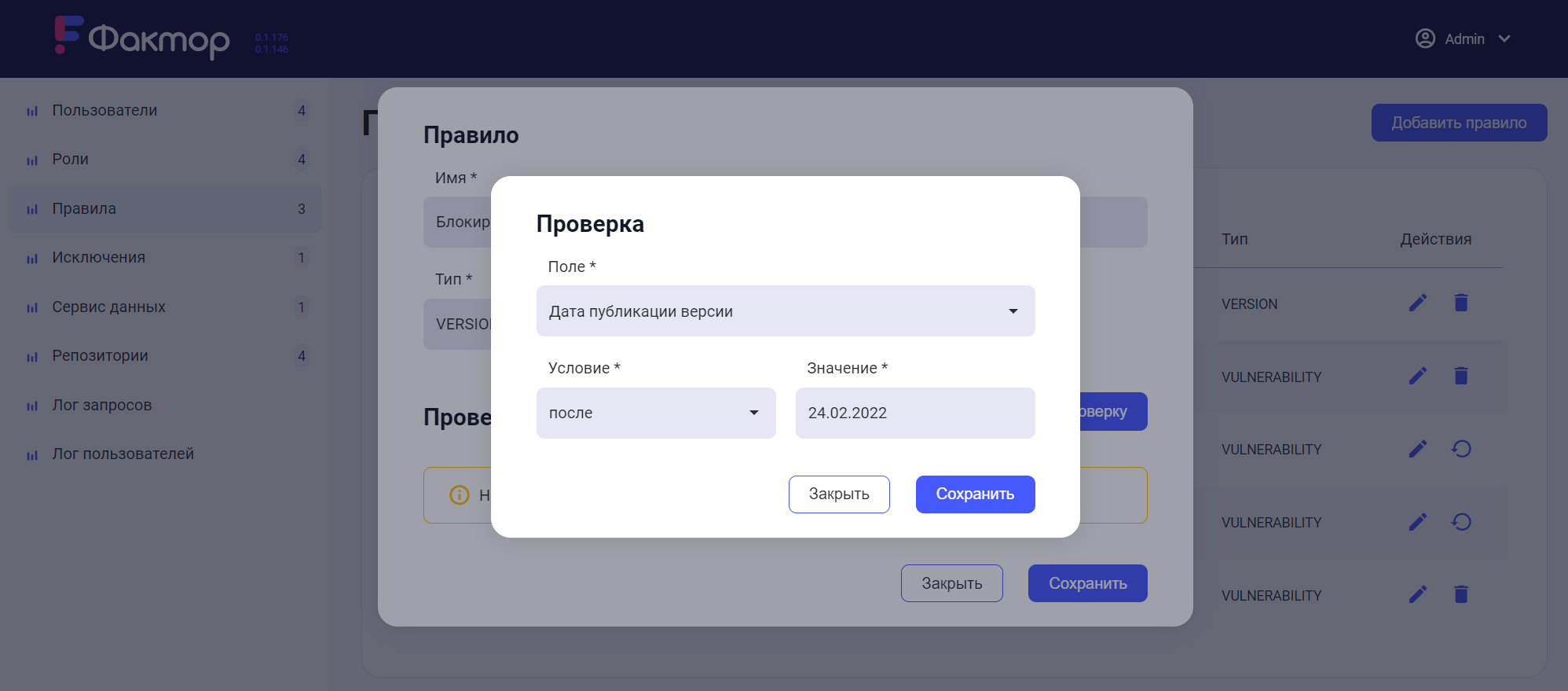
### Добавление правила

Чтобы создать новое правило в системе, выберите пункт **Правила** в меню слева и нажмите кнопку **Добавить правило** справа вверху.

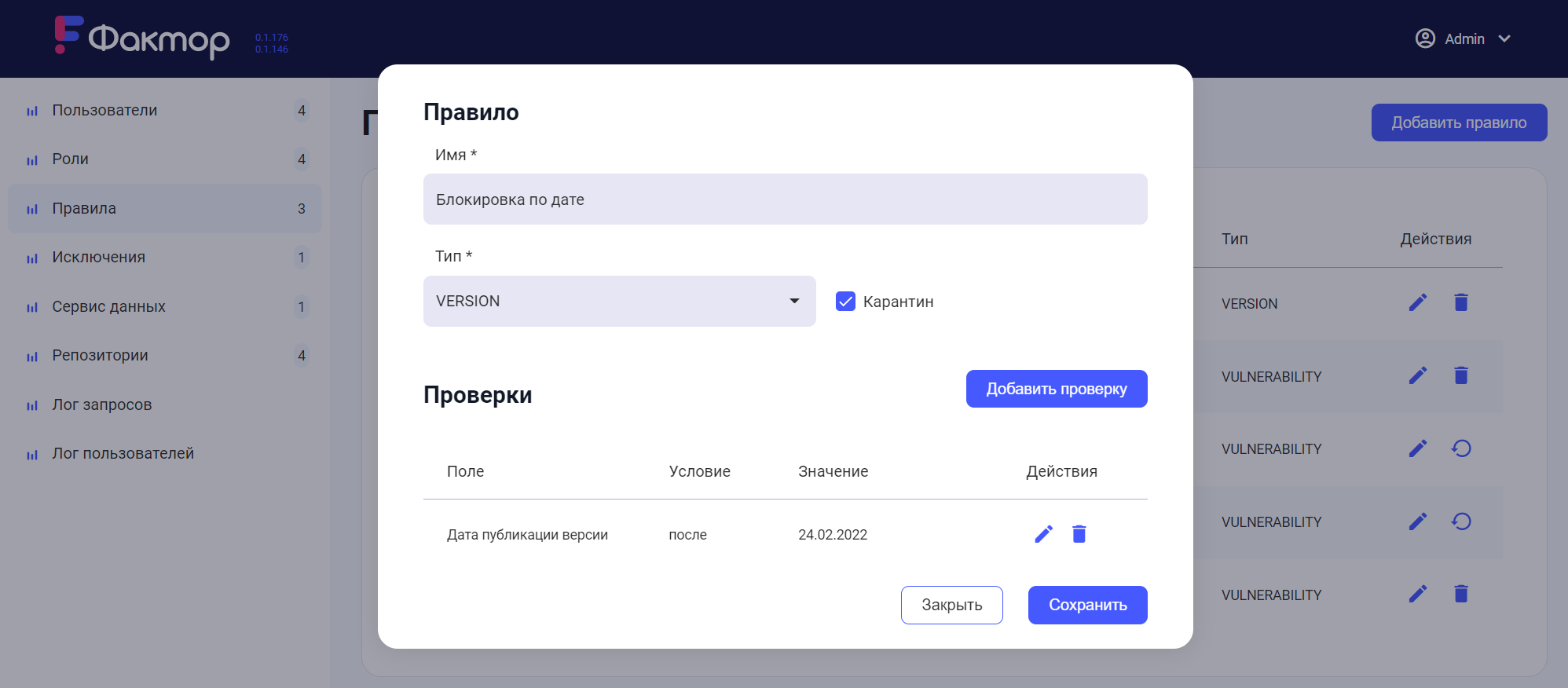
В появившемся окне введите имя правила, выберите из выпадающего списка его тип (VULNERABILITY или VERSION) и определите, будет ли оно блокирующим загрузку или нет. Для блокирующего правила выберите опцию **Карантин**.



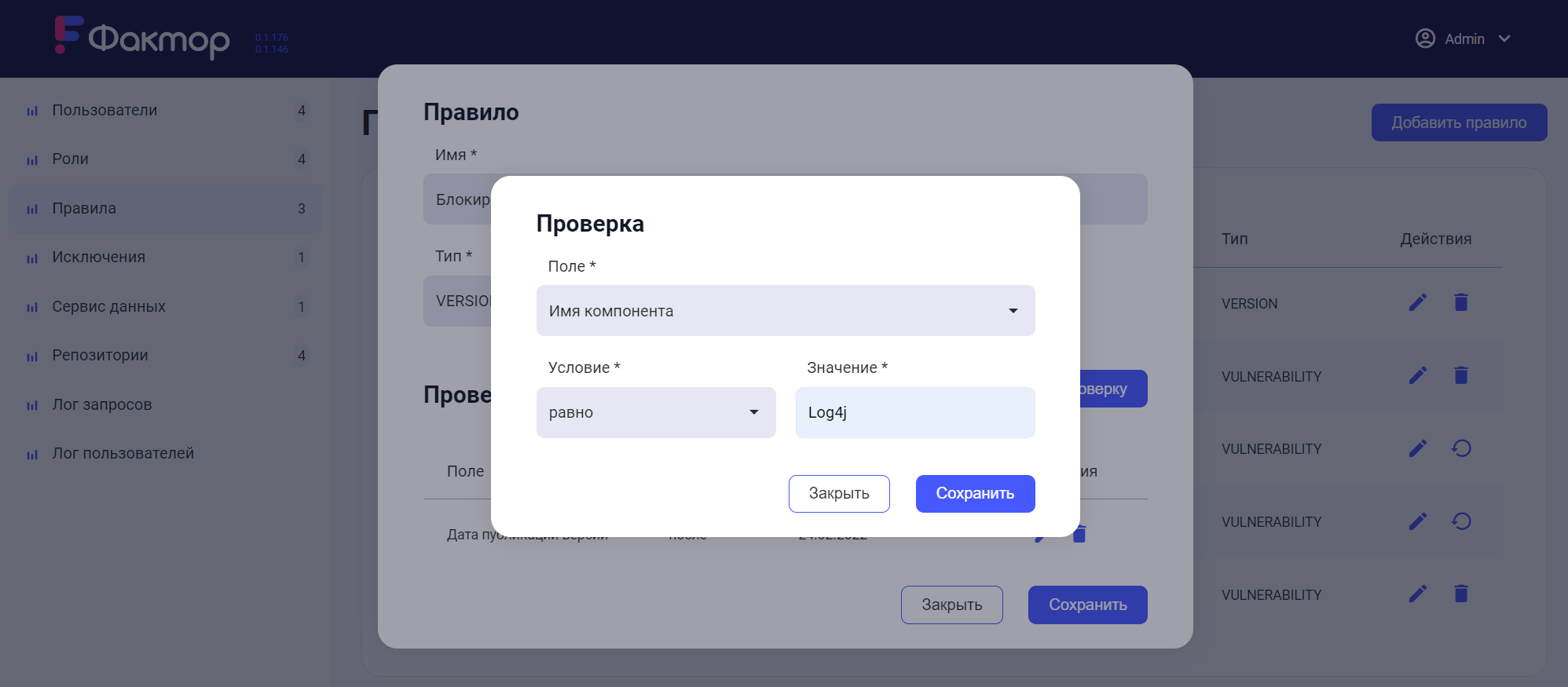
Нажмите кнопку **Добавить проверку** и в появившемся окне выберите из выпадающего списка критерий проверки, условие проверки и его значение и нажмите кнопку **Сохранить**.



Добавленная проверка появится в списке проверок вновь создаваемого правила.



Правило в системе может содержать одну или несколько проверок. Например, во вновь создаваемое правило можно добавить проверку, по которой данное правило будет работать только для конкретного компонента — Log4j. Для этого нажмите кнопку **Добавить проверку** и задайте следующие значения.



Вторая добавленная проверка появится в списке проверок вновь создаваемого правила. Теперь данная политика состоит из двух условий — по дате публикации версии и по имени компонента. Политика будет срабатывать только в том случае, когда оба этих условия будут выполнены.

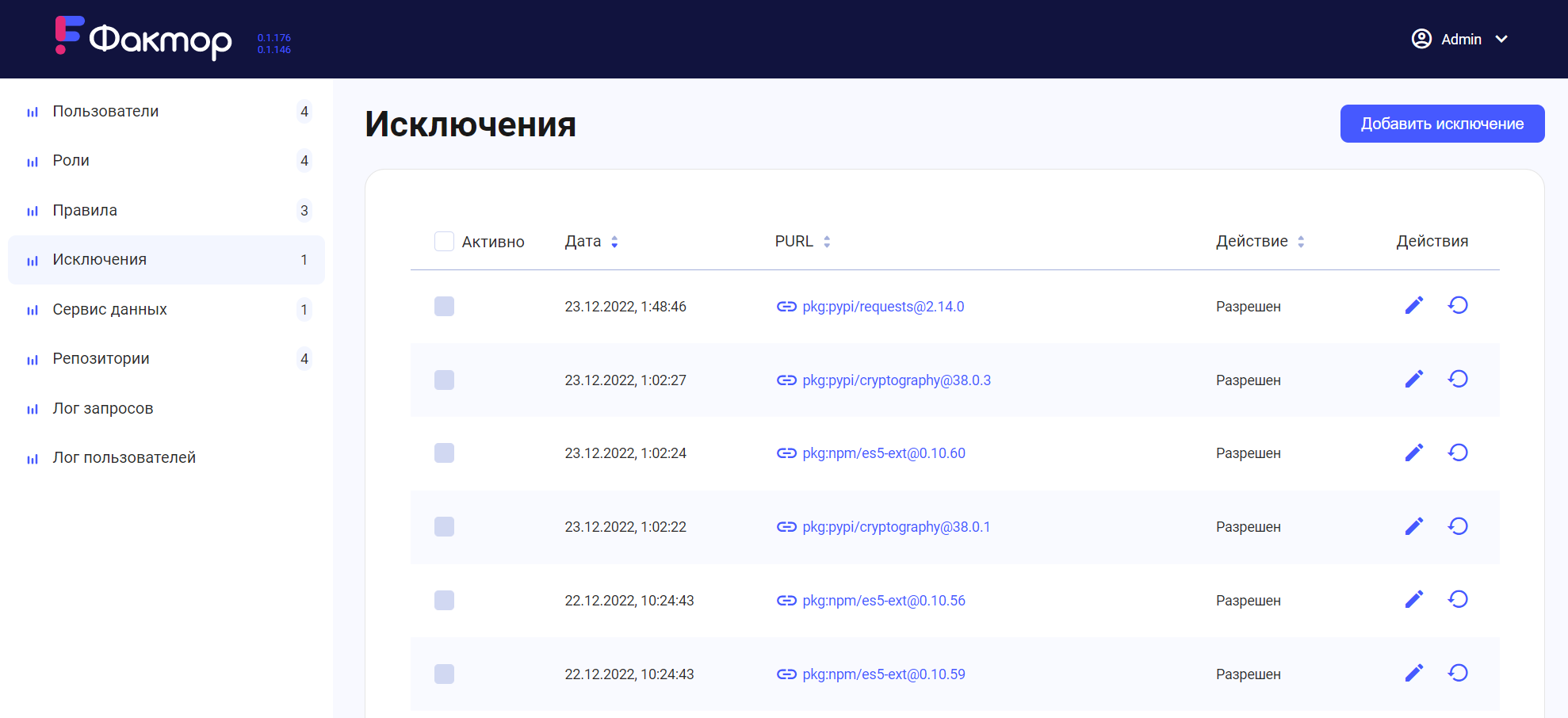
Каждую из проверок в правиле можно отредактировать или удалить. Добавьте в политику все необходимые проверки и нажмите кнопку **Сохранить**.

Вновь созданная политика появится в списке правил в системе. После этого заданная данным правилом политика безопасности будет применяться ко всем загружаемым компонентам.

## Исключения

Для обеспечения гибкости при работе с политиками безопасности система предоставляет возможность создавать для отдельных компонентов исключения. Это позволяет загружать данные компоненты даже в том случае, если они нарушают установленные в системе политики безопасности.

Выберите пункт **Исключения** в меню слева, чтобы посмотреть список существующих в системе исключений.



Для каждого исключения указаны его статус в системе (активно/неактивно), дата создания, PURL компонента, для которого создано исключение, и действие исключения (Запрещен/Разрешен).

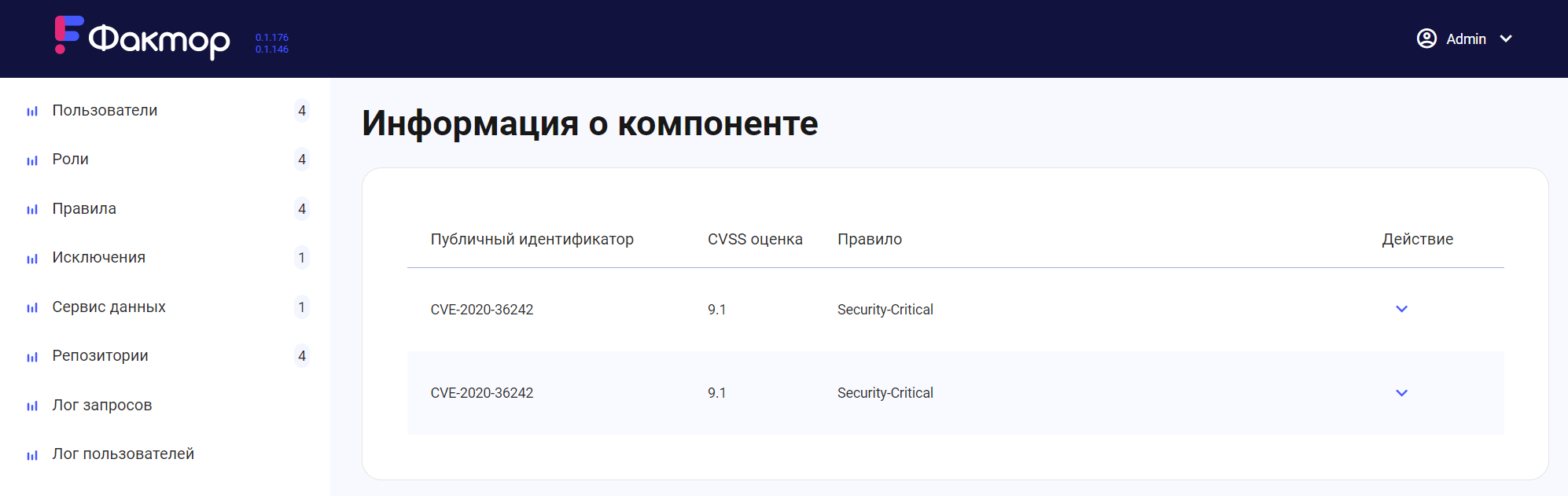
Исключение может разрешать или запрещать загрузку компонента. Исключение, действие которого определено как «Запрещен», запрещает загрузку компонента, даже если он не имеет известных проблем безопасности. Исключение, действие которого определено как «Разрешен», разрешает загрузку компонента, даже если он имеет известные проблемы безопасности.

Исключение можно отредактировать, нажав на кнопку  в конце его строки.

Неактивное исключение можно активировать, нажав на кнопку  в конце его строки. Активное исключение можно деактивировать, нажав на кнопку  в конце его строки.

Если для компонента в системе действует активное исключение, при его загрузке в лог запросов будет добавлена соответствующая запись о том, что компонент разрешен к использованию (см. детали в разделе «[Лог запросов](./query%20log.md)»).

При нажатии на PURL компонента в строке исключения на экране появится страница **Информация о компоненте**, содержащая список всех обнаруженных во время проверки компонента уязвимостей и нарушений установленных политик безопасности.



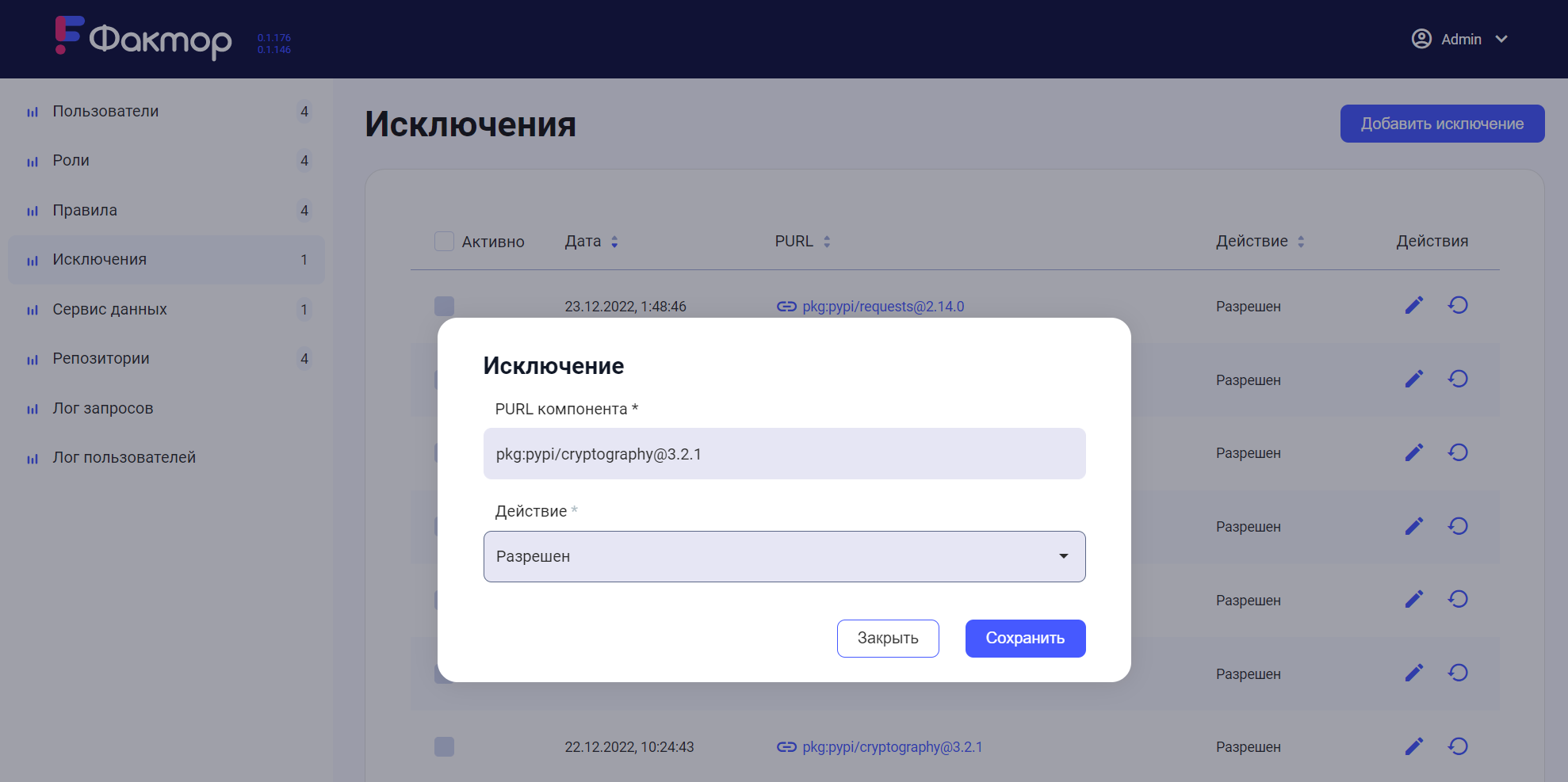
### Добавление исключения

Для внесения компонента в список исключений пользователю, работающему с менеджером репозиториев, необходимо сделать запрос на его разблокировку. После этого на странице **Исключения** появится запись с данными об этом запросе. При этом само исключение будет находиться в статусе «неактивно». В случае принятия решения о разблокировке компонента необходимо будет перевести исключение в статус «активно», нажав на кнопку активации  в конце его строки.

Также можно внести компонент в исключения непосредственно в Factor.Block на странице **Исключения**.

Чтобы создать новое исключение в системе, выберите пункт **Исключения** в меню слева и нажмите кнопку **Добавить исключение** справа вверху.

В появившемся окне введите PURL компонента, для которого создается исключение, выберите из выпадающего меню действие исключения (Запрещен или Разрешен) и нажмите кнопку **Сохранить**.

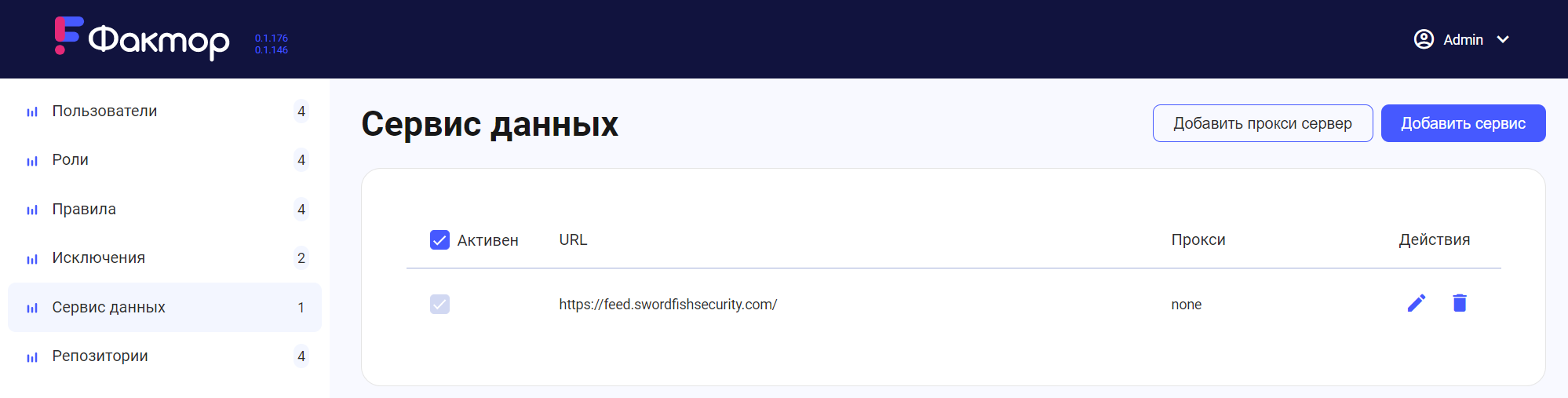


Вновь созданное исключение появится в списке исключений в системе.

## Сервис данных

Сервис данных определяет адрес используемой в системе базы данных — Factor.Feed. Кроме этого, на странице **Сервис данных** указывается используемый для работы системы токен подключения.

Выберите пункт **Сервис данных** в меню слева.



Для каждого сервиса данных указаны его статус в системе (активен/неактивен), URL, а также, если используется, прокси-сервер.

Сервис данных можно отредактировать, нажав на кнопку  в конце его строки.

Неактивный сервис данных можно активировать, нажав на кнопку  в конце его строки. Активный сервис данных можно деактивировать, нажав на кнопку  в конце его строки.

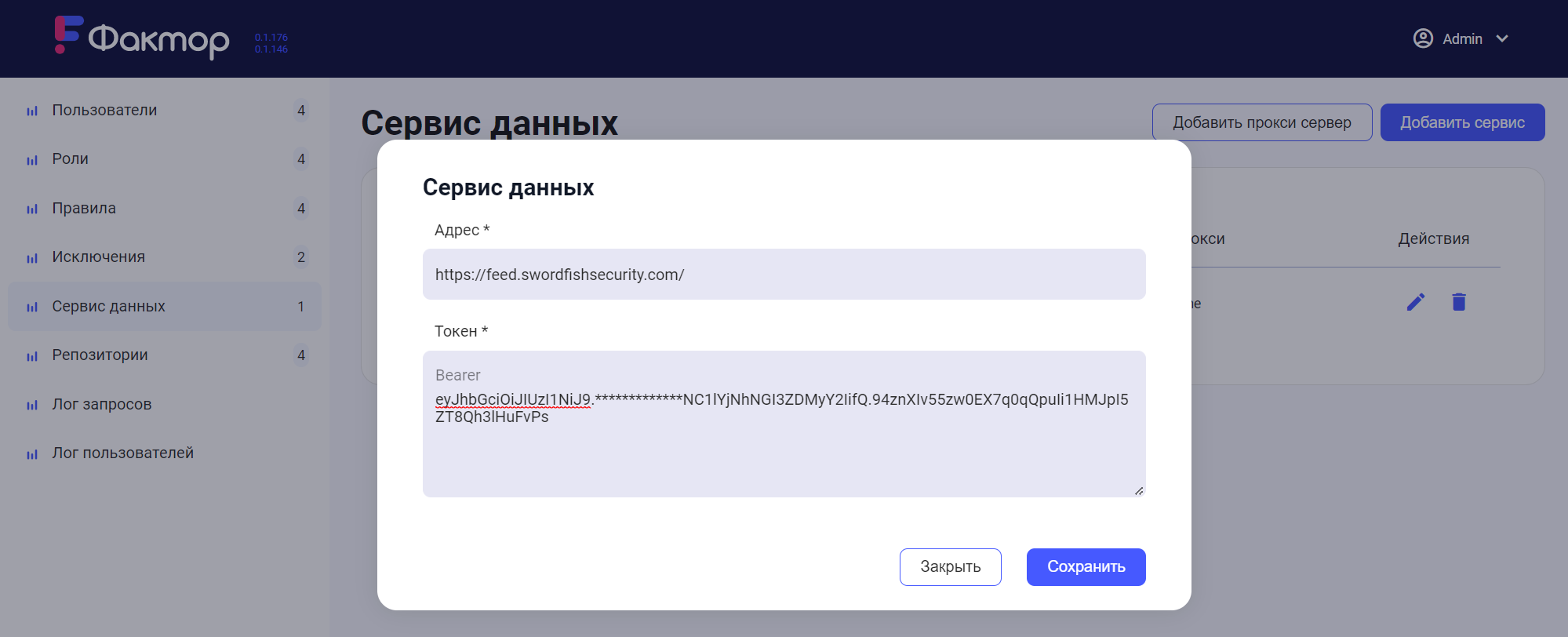
Стандартной схемой работы является использование одного сервиса данных, что соответствует одному активному сервису данных в системе. Если в системе используется нестандартная конфигурация и одновременно присутствуют два активных сервиса данных, Factor.Block будет обращаться к ним последовательно и объединять полученные результаты.

### Токен подключения

Токен подключения является в системе идентификатором лицензии пользователя. Токен для подключения к Factor.Feed входит в комплект поставки системы.

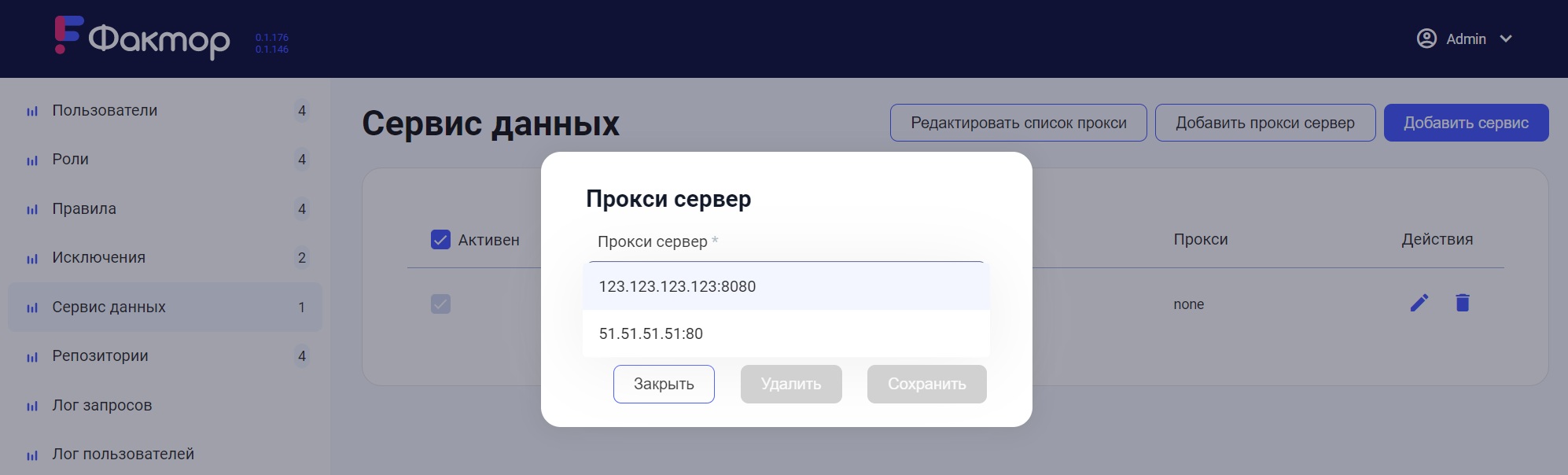
Чтобы просмотреть данные используемого токена или изменить их, нажмите кнопку редактирования  в конце строки сервиса данных.

В появившемся окне в поле **Токен** указан текущий токен подключения. Если необходимо заменить его на новый, введите значение нового токена и нажмите кнопку **Сохранить**.

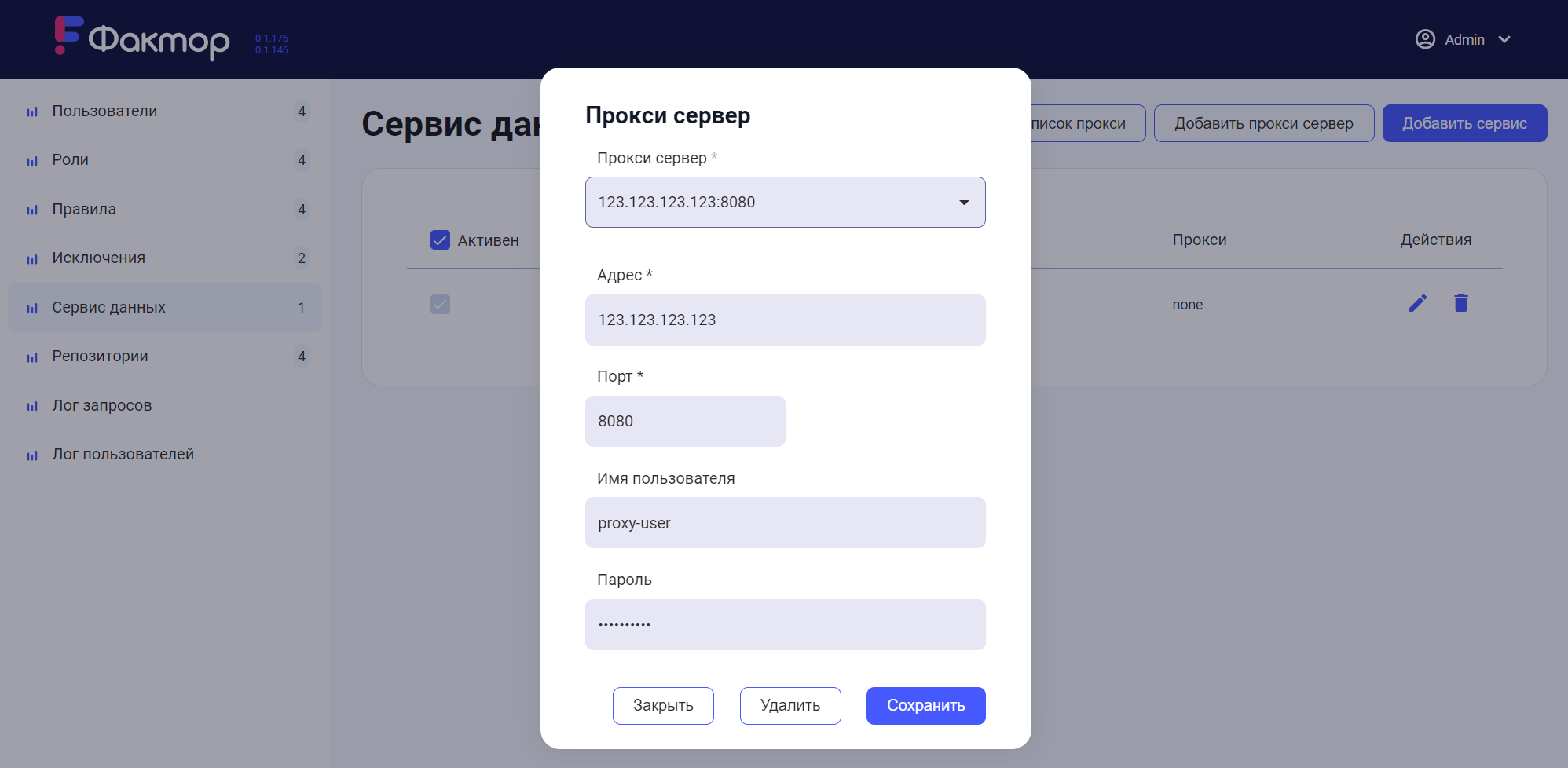


### Прокси-серверы

Данные обо всех прокси-серверах, используемых при работе с сервисом данных, хранятся в списке прокси-серверов. Чтобы просмотреть список прокси-серверов, нажмите кнопку **Редактировать список прокси** справа вверху. Все прокси-сервера будут перечислены в выпадающем списке в появившемся окне.

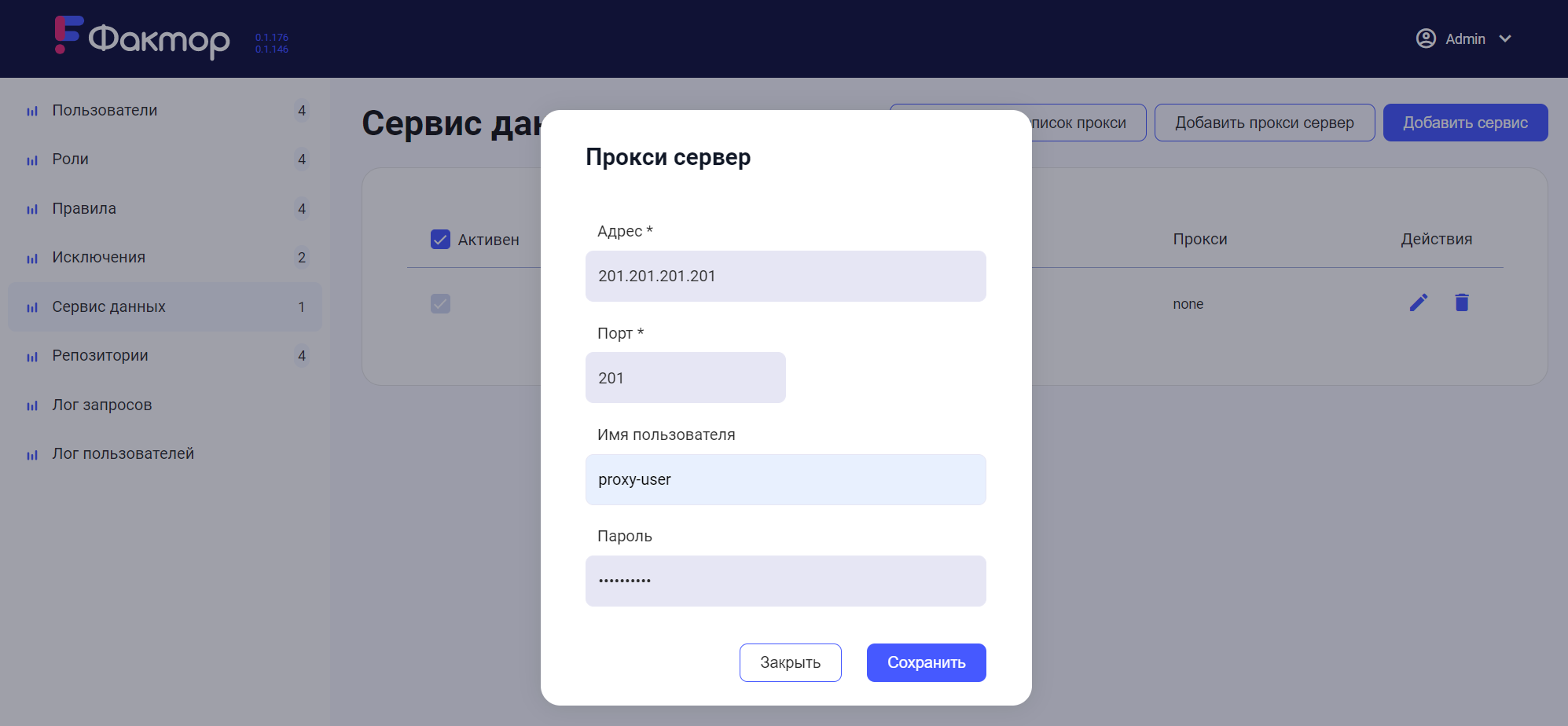


В данном окне выберите прокси-сервер из выпадающего списка. На экране появится окно с детальной информацией о прокси-сервере, включая его адрес, порт, а также имя и пароль пользователя.



Информацию о прокси-сервере можно отредактировать в данном окне и сохранить.

Чтобы добавить прокси-сервер, нажмите кнопку **Добавить прокси сервер** справа вверху. В появившемся окне введите информацию о прокси-сервере, включая его адрес и порт (обязательные параметры), а также имя и пароль пользователя (необязательные параметры) и нажмите кнопку **Сохранить**.

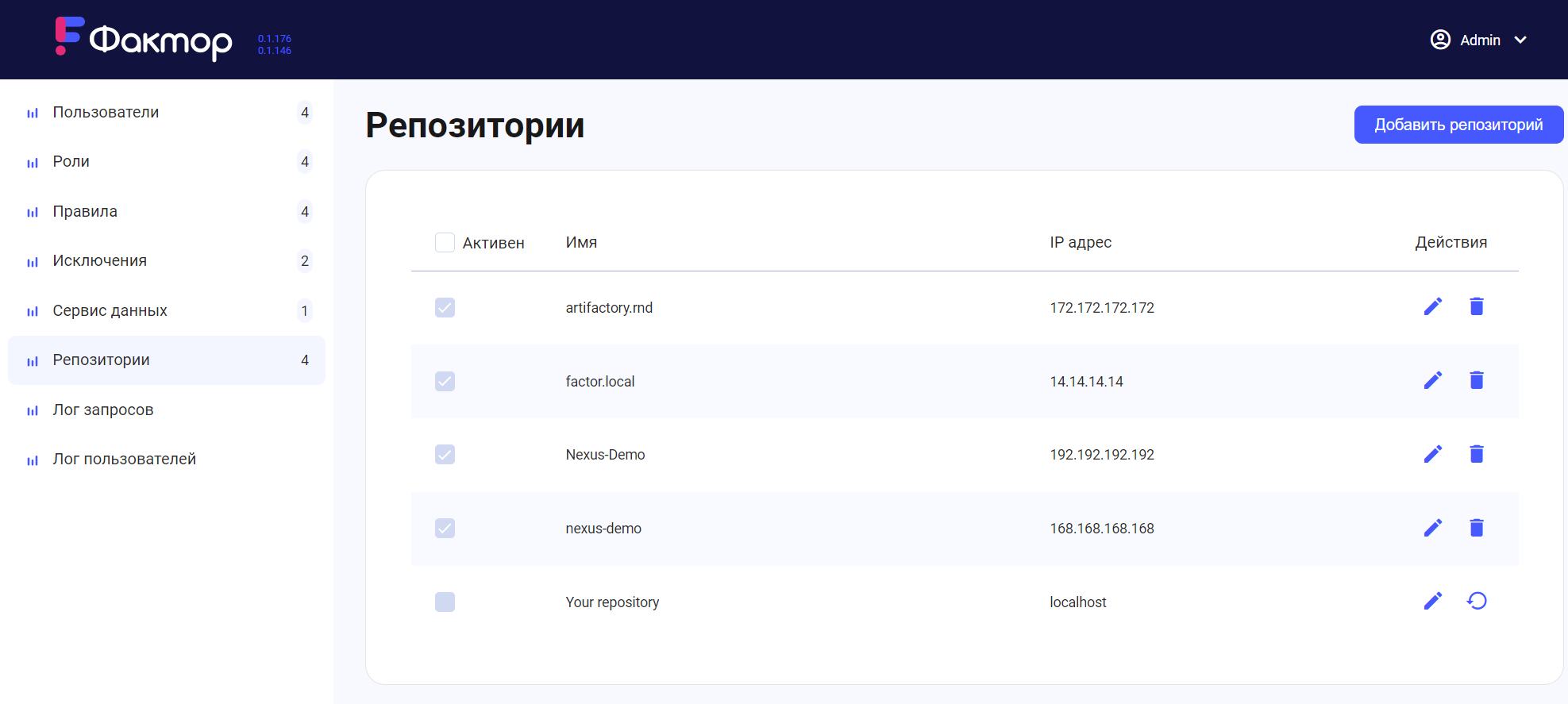


Вновь созданный прокси-сервер будет добавлен в список прокси-серверов.

## Репозитории

В системе представлены данные обо всех подключенных репозиториях Nexus Repository Manager и JFrog Artifactory, в которых установлены плагины Factor.Plugin.

Выберите пункт **Репозитории** в меню слева.



Для каждого репозитория указаны его статус в системе (активен/неактивен), имя и IP-адрес. Система работает только с активными репозиториями.

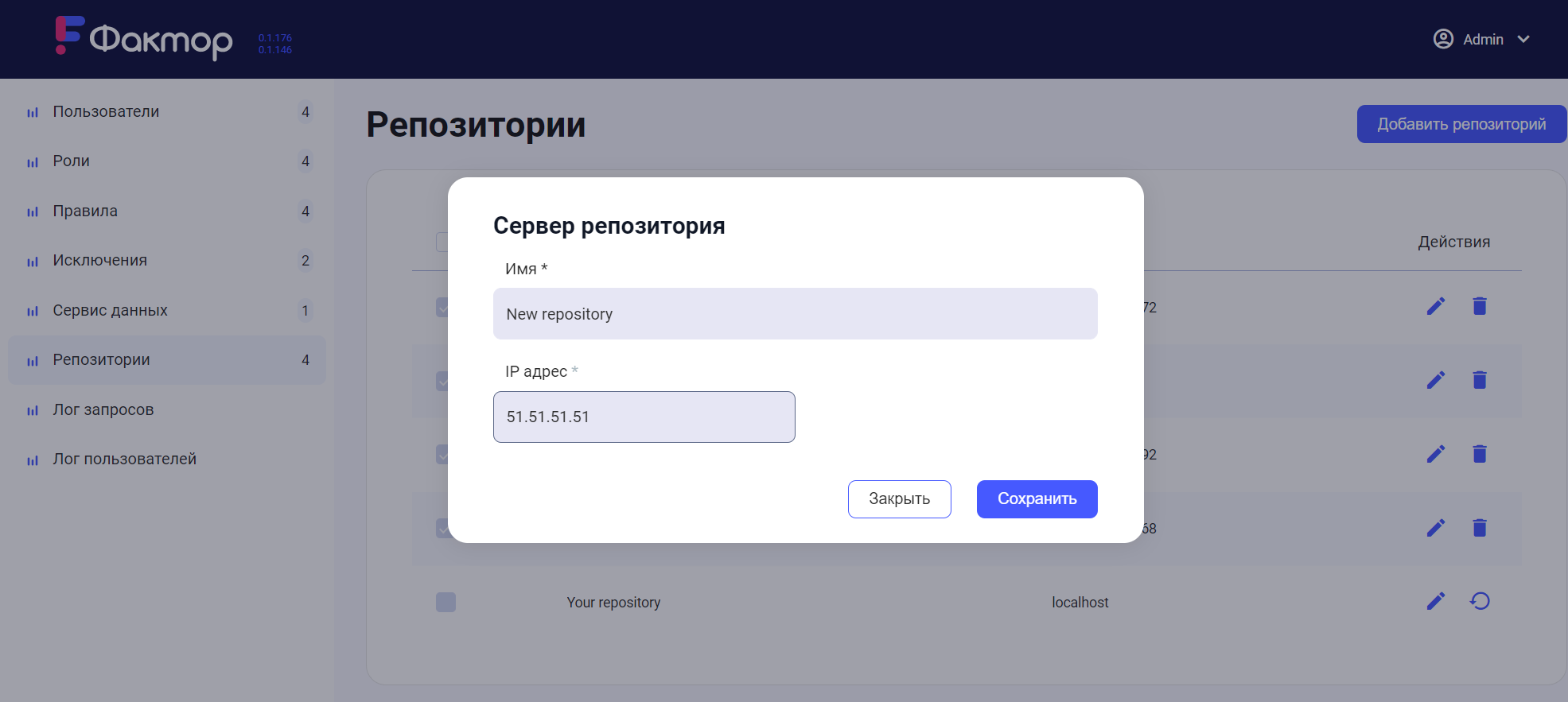
Репозиторий можно отредактировать, нажав на кнопку  в конце его строки.

Неактивный репозиторий можно активировать, нажав на кнопку  в конце его строки. Активный репозиторий можно деактивировать, нажав на кнопку  в конце его строки.

### Добавление репозитория

Чтобы добавить репозиторий в систему, выберите пункт **Репозитории** в меню слева и нажмите кнопку **Добавить репозиторий** справа вверху.

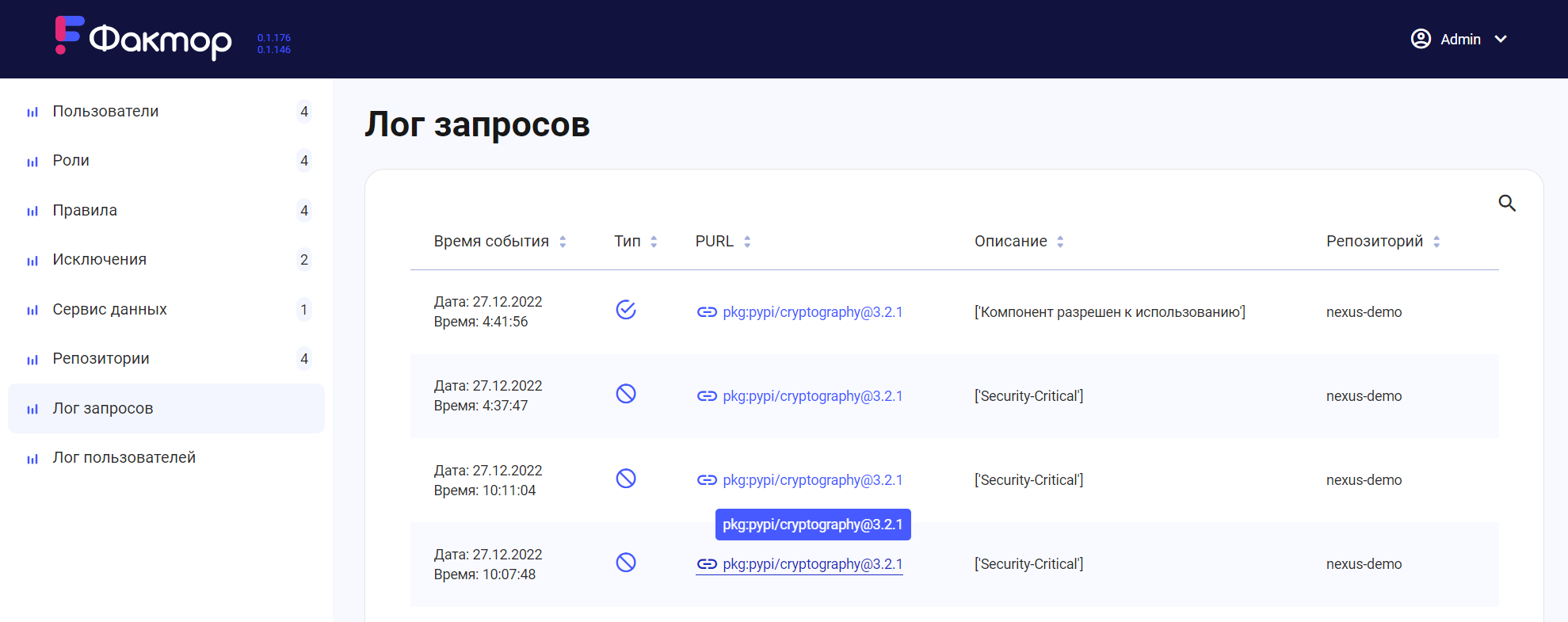
В появившемся окне введите имя и IP-адрес сервера репозитория и нажмите кнопку **Сохранить**.



## Лог запросов

Лог запросов содержит информацию обо всех произведенных пользователями менеджеров репозиториев запросах на скачивание компонентов.

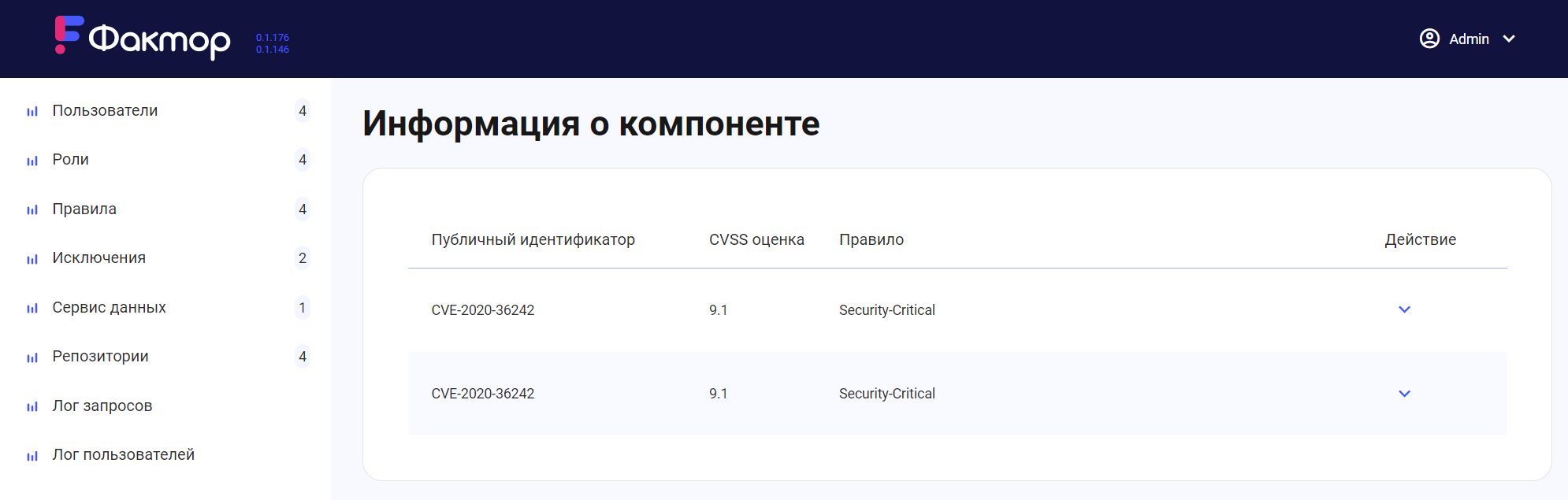
Выберите пункт **Лог запросов** в меню слева.



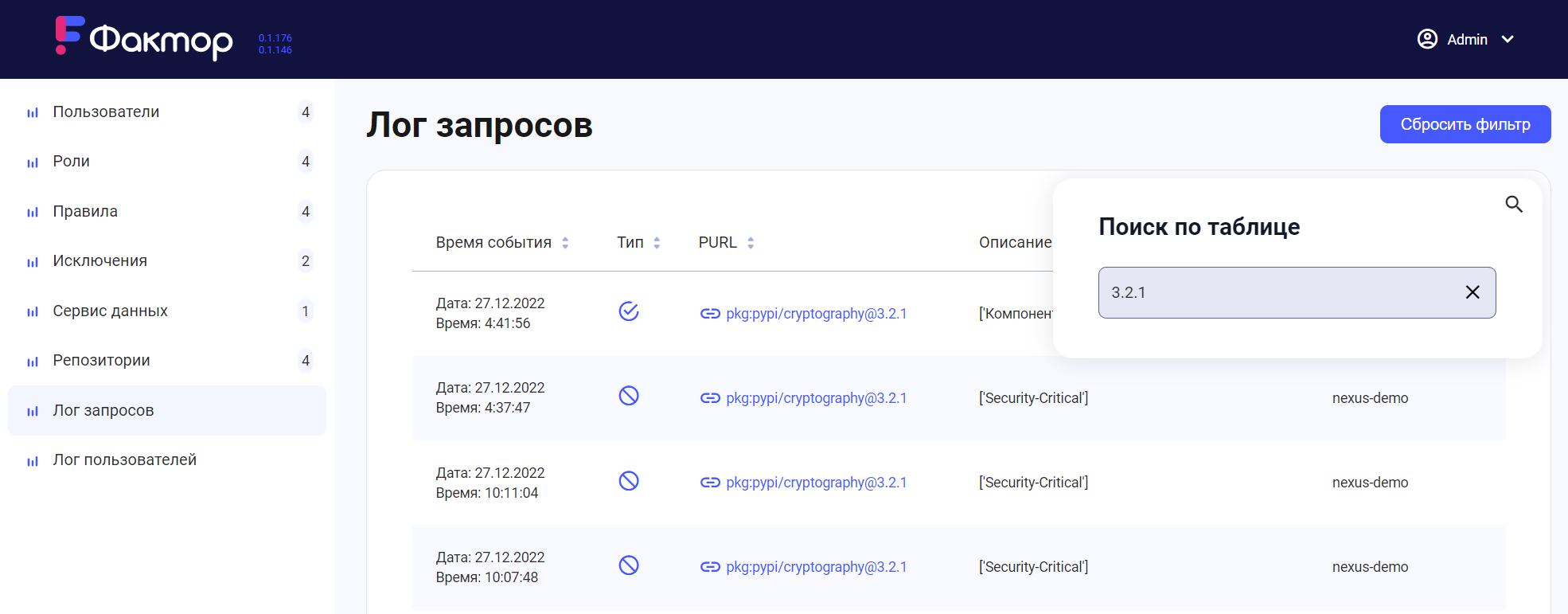
Для каждого запроса указаны дата и время запроса, его тип ( — компонент разрешен к скачиванию /  — компонент заблокирован, скачивание не было произведено из-за блокирующей политики безопасности /  — во время скачивания произошла ошибка), PURL запрошенного на скачивание компонента, описание результата запроса и имя репозитория.

Если политика безопасности не является блокирующей, компонент может быть успешно скачан даже при наличии в описании результата запроса найденных уязвимостей или нарушений установленной политики безопасности. Для компонентов, добавленных в исключения, в описании результата запроса приводится информация «Компонент разрешен к использованию».

При нажатии на PURL компонента в строке запроса на экране появится страница **Информация о компоненте**, содержащая список всех обнаруженных во время проверки компонента уязвимостей и нарушений установленных политик безопасности. Нажмите на строку уязвимости или нарушения политики безопасности, чтобы получить более детальную информацию, включая данные компонента.



В логе запросов реализована фильтрация по информации в колонках **PURL**, **Описание** и **Репозиторий**. Чтобы отфильтровать необходимые запросы, нажмите на кнопку  справа вверху и в появившемся окне **Поиск по таблице** введите информацию для поиска.

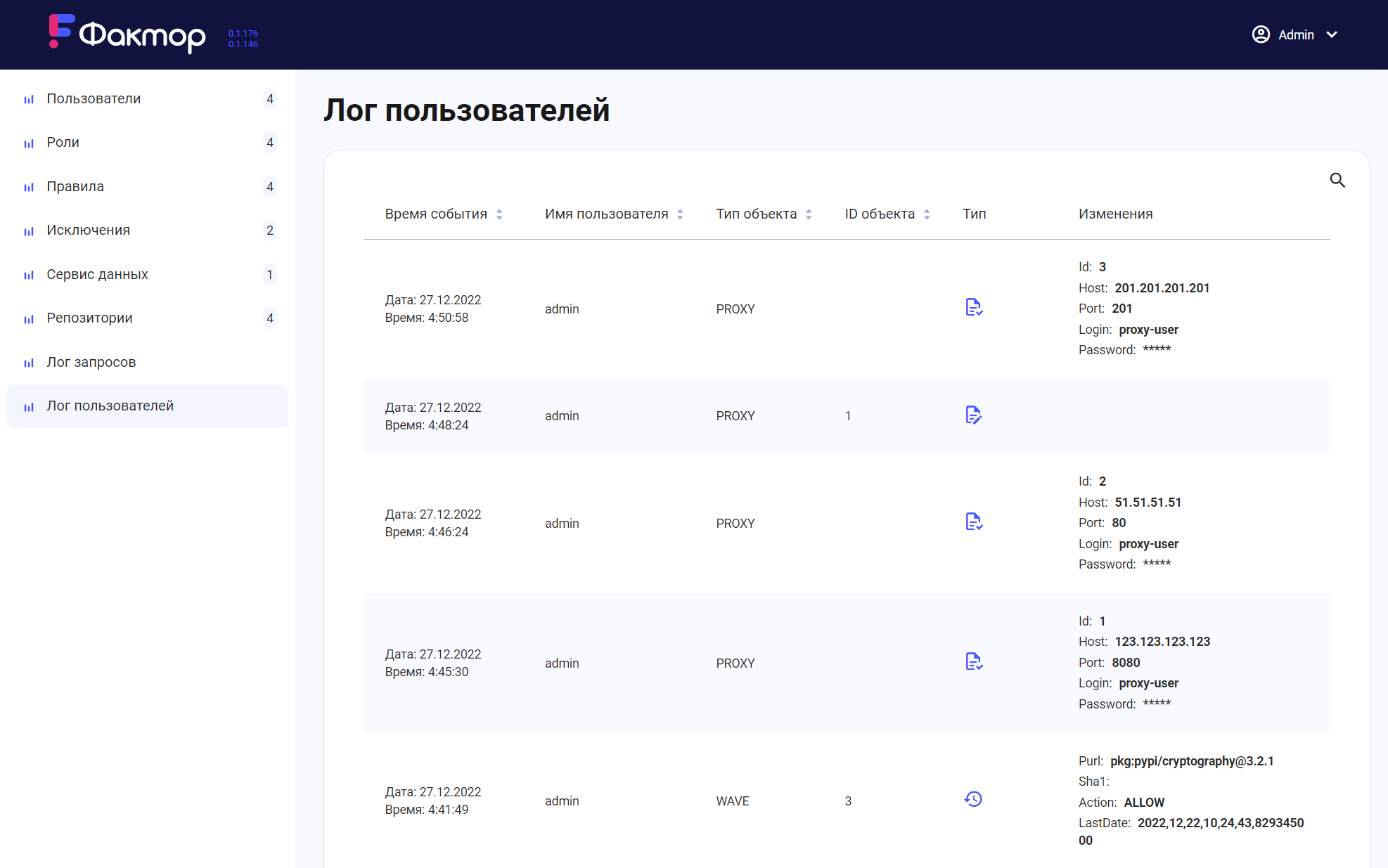


Нажмите кнопку **Сбросить фильтр**, чтобы вернуться к полному списку запросов.

## Лог пользователей

Лог пользователей содержит информацию обо всех действиях, произведенных пользователями в системе.

Выберите пункт **Лог пользователей** в меню слева.

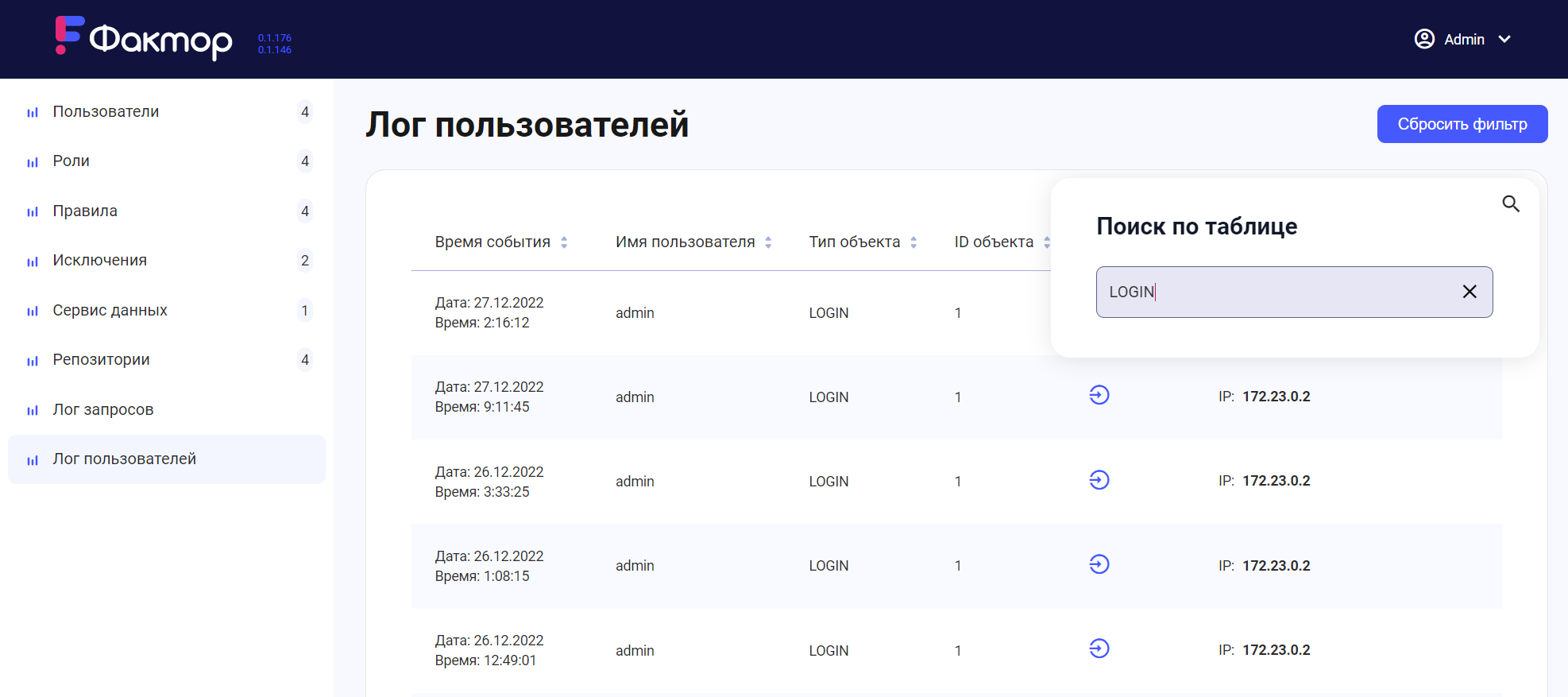


Для каждого запроса указаны дата и время действия в системе, имя пользователя, совершившего действие, тип объекта, с которым совершено действие, ID объекта в системе, тип произведенного действия и краткое описание произведенных изменений.

Типы объектов, с которым пользователь совершил действие, включают в себя все основные сущности, с которыми работает система, например, LOGIN — при входе пользователя в систему, ROLE, USER ROLES и ROLE\_AUTHORITY — при работе с ролями пользователей в системе, WAVE — при работе с исключениями, RULE — при работе с политиками безопасности, PLUGIN — при работе с репозиториями, BACKEND — при работе с сервисами данных.

Тип действия зависит от объекта, с которым оно было произведено. Стандартными типами действий в системе являются: вход пользователя в систему, создание объекта, редактирование объекта, активация объекта, деактивация объекта.

В логе пользователей реализована фильтрация по информации в колонках **Имя пользователя**, **Тип объекта**, **ID объекта** и **Изменения**. Чтобы отфильтровать необходимые действия, нажмите на кнопку  справа вверху и появившемся окне **Поиск по таблице** введите информацию для поиска.



Нажмите кнопку **Сбросить фильтр**, чтобы вернуться к полному списку действий пользователей.

## Приложение 1. Права доступа

|  |  |
| --- | --- |
| Обозначение | Описание |
| VIEW\_USER | Просмотр списка пользователей |
| EDIT\_USER | Редактирование пользователей (изменение паролей, ролей) |
| DELETE\_USER | Деактивация пользователей |
| VIEW\_ROLE | Просмотр списка ролей |
| EDIT\_ROLE | Редактирование ролей (изменение имени, добавление прав) |
| DELETE\_ROLE | Деактивация ролей |
| VIEW\_AUTHORITIES | Просмотр списка прав доступа |
| VIEW\_BACKEND | Просмотр списка подключенных облачных сервисов |
| EDIT\_BACKEND | Редактирование списка облачных сервисов |
| DELETE\_BACKEND | Деактивация подключенного сервиса |
| VIEW\_PLUGIN | Просмотр списка подключенных репозиториев |
| EDIT\_PLUGIN | Редактирование списка подключенных репозиториев |
| DELETE\_PLUGIN | Деактивация подключенного репозитория |
| VIEW\_PROXY | Просмотр списка прокси (используется для доступа к облачному сервису) |
| EDIT\_PROXY | Редактирование прокси-серверов |
| DELETE\_PROXY | Деактивация прокси-сервера |
| VIEW\_RULE | Просмотр правил |
| EDIT\_RULE | Редактирование правил |
| DELETE\_RULE | Деактивация правил |
| VIEW\_REQUEST\_LOG | Просмотр логов запросов со стороны плагина |
| VIEW\_USER\_LOG | Просмотр логов действий пользователей в системе |
| VIEW\_WAVE | Просмотр исключений |
| EDIT\_WAVE | Редактирование исключений |
| DELETE\_WAVE | Деактивация исключений |