



МПК-ПЕРЕСВЕТ

Программа для ЭВМ «МПК-Пересвет»

Руководство администратора

Москва 2025

ГЛОССАРИЙ

Термин	Определение
Модель технического объекта	Совокупность статической и динамической моделей объекта.
Статическая модель объекта	Иерархия сущностей. Также в иерархию включаются сущности, описывающие информационную систему. В простейшем случае статическая модель может представлять собой просто линейный список тегов (параметров объектов).
Динамическая модель	Совокупность методов, реализуемых на каких-либо языках программирования. В виде методов реализуются бизнес-процессы, протекающие на моделируемом объекте, а также выполняются расчёты значений рассчитываемых тегов. Выполнение методов инициируется при возникновении событий в модели.
Сущность	Основные сущности статической модели технического объекта: теги, объекты, тревоги, методы, расписания, константы. Дополнительные сущности, описывающие информационную систему: хранилище данных, источник данных. Также есть возможность создавать дополнительные сущности, самостоятельно реализуя логику работы с ними.
Тег	Параметр объекта. Например: температура, давление, сила тока и т.д. Значения тегов хранятся в том или ином хранилище данных. Каждое значение тега обязательно имеет метку времени. Кроме того, может иметь показатель качества значения. Тег в иерархии представляется узлом, дочерним по отношению к какому-либо объекту. Поддерживаемые типы значений тегов: целый, вещественный, строковый, json.
Объект	Основная сущность при моделировании технических объектов. Объектом может быть: предприятие, цех, участок, технологическая линия, агрегат, датчик и т.д. Каждый объект может содержать любое количество объектов-потомков. Также объект может содержать любое количество тегов.
Расписание	Заданная последовательность, определяющая моменты времени, в которые будут выполняться те или иные задачи. Задачами могут быть: запуск расчёта вычисляемых тегов, запуск на исполнение каких-либо методов (к примеру, генерация отчётов и рассылка их по почте).
Событие	В платформе поддерживаются три основных типа событий: изменение тега, возникновение тревоги, событие расписания. К этим событиям можно привязывать выполнение различных методов, тем самым «оживляя» модель технического объекта. События являются инициаторами действий.

Тревога	<p>Это состояние тега, которое может возникнуть при изменении тега и при выполнении некоторых заданных условий. Есть четыре вида стандартных тревог: LoLo, Lo, Hi, HiHi.</p> <p>В «МПК-Пересвет», в отличие от типовых тревог LoLo-Lo-Hi-HiHi, есть возможность создать любое их количество. Более того, возможно создавать сложные тревоги, возникновение которых учитывает значение других тегов. Это достигается тем, что условием возникновения тревоги может быть указано не только какое-то значение тега, а результат расчёта метода, который может реализовывать любую сложную логику.</p>
Метод	<p>Программный код, реализующий логику поведения моделируемого объекта. Методы вызываются на исполнение возникающими событиями и используются для: расчёта значений вычисляемых тегов, определения факта возникновения/пропадания тревоги, вызова внешних процессов и т.д.</p>
Хранилище данных	<p>База данных, в которой хранятся исторические данные (значения тегов). Платформа может поддерживать одновременно несколько хранилищ данных разных типов. Рекомендуемое хранилище данных для систем промышленной автоматизации - PostgreSQL. Также есть драйвер для VictoriMetrics. Возможно написание драйверов для любых других типов хранилищ данных.</p>
Коннектор	<p>Программа, обеспечивающая сбор данных с устройства по определённому протоколу. Обычно для каждого протокола создаётся отдельный коннектор.</p>

Содержание

Описание.....	5
Область применения.....	5
Возможности.....	5
Уровень подготовки администратора.....	5
Архитектура МПК-Пересвет.....	6
Микросервис.....	8
OpenLDAP.....	8
PostgreSQL.....	8
Redis.....	9
Grafana.....	9
Nginx.....	9
RabbitMQ.....	9
Коннектор.....	10
Требования к аппаратному обеспечению.....	10
Администрирование.....	10
Файл с переменными.....	10
Docker-сеть.....	11
Запуск МПК-Пересвет.....	11
HTTP.....	11
HTTPS. Самоподписанные сертификаты.....	12
HTTPS. Сертификаты Let's Encrypt.....	13
Мониторинг состояния.....	13
Конфигуратор модели.....	15
Папки модели.....	16
Объекты.....	17
Теги.....	17
Коннекторы.....	17
Расписания.....	17
Хранилища данных.....	17
Создание новых узлов.....	17
Редактирование свойств узла.....	19
Удаление узла.....	20
Создание экранов в Grafana.....	21
Создание нового экрана и панели.....	21

Описание

МПК-Пересвет (далее по тексту — платформа, система)— платформа-конструктор для создания моделей технических объектов и информационных систем.

Область применения

Платформа может использоваться для сбора, хранения, обработки данных, а также автоматизации процессов, протекающих в рамках моделируемого технического объекта.

1. Промышленная автоматизация уровней SCADA, диспетчеризация/мониторинг, MES.
2. Умный дом, умное здание.
3. Создание произвольных программно-аппаратных комплексов, предполагающих работу по следующей схеме: автоматический сбор данных с большого количества источников → хранение исторических данных → обработка данных, запуск процессов → отображение данных на динамических экранах/мнемосхемах.

Возможности

1. Создание моделей объектов и информационных систем.
2. Автоматический сбор данных с большого количества источников с дальнейшим сколь угодно долгим хранением всех собранных данных.
3. Создание и управление систем распределённых вычислений, моделирующих поведение объектов.
4. Наличие API, позволяющего полностью управлять системой.

Уровень подготовки администратора

Базовые навыки администратора МПК-Пересвет:

1. Операционные системы семейства Linux: пользователь, минимальные навыки администрирования (управление правами, запуск sh-скриптов).
2. Установка Docker, Docker compose.
3. Настоящее руководство администратора.

Дополнительные навыки для случая масштабирования системы и повышенного уровня безопасности:

1. Более глубокие навыки администрирования ОС семейства Linux.
2. Понимание принципов функционирования TLS, умение работать с сертификатами безопасности.
3. Администрирование Docker, Docker compose: управление экземплярами контейнеров, настройка сетей между контейнерами, работа с томами данных.

4. Конфигурирование Nginx.

Мониторинг работы и тонкая настройка компонентов, входящих в состав МПК-Пересвет предполагает знание:

1. OpenLDAP.
2. RabbitMQ.
3. Nginx.
4. Uvicorn или Nginx.Unit.
5. Redis.
6. PostgreSQL или другая база, используемая для хранения исторических данных.

Создание экранов пользователя предполагает наличие навыков администрирования Grafana.

При необходимости создания сложных экранов администратор должен обладать навыками web-программирования.

Архитектура МПК-Пересвет

На рисунке ниже представлена архитектура компонентов МПК-Пересвет:

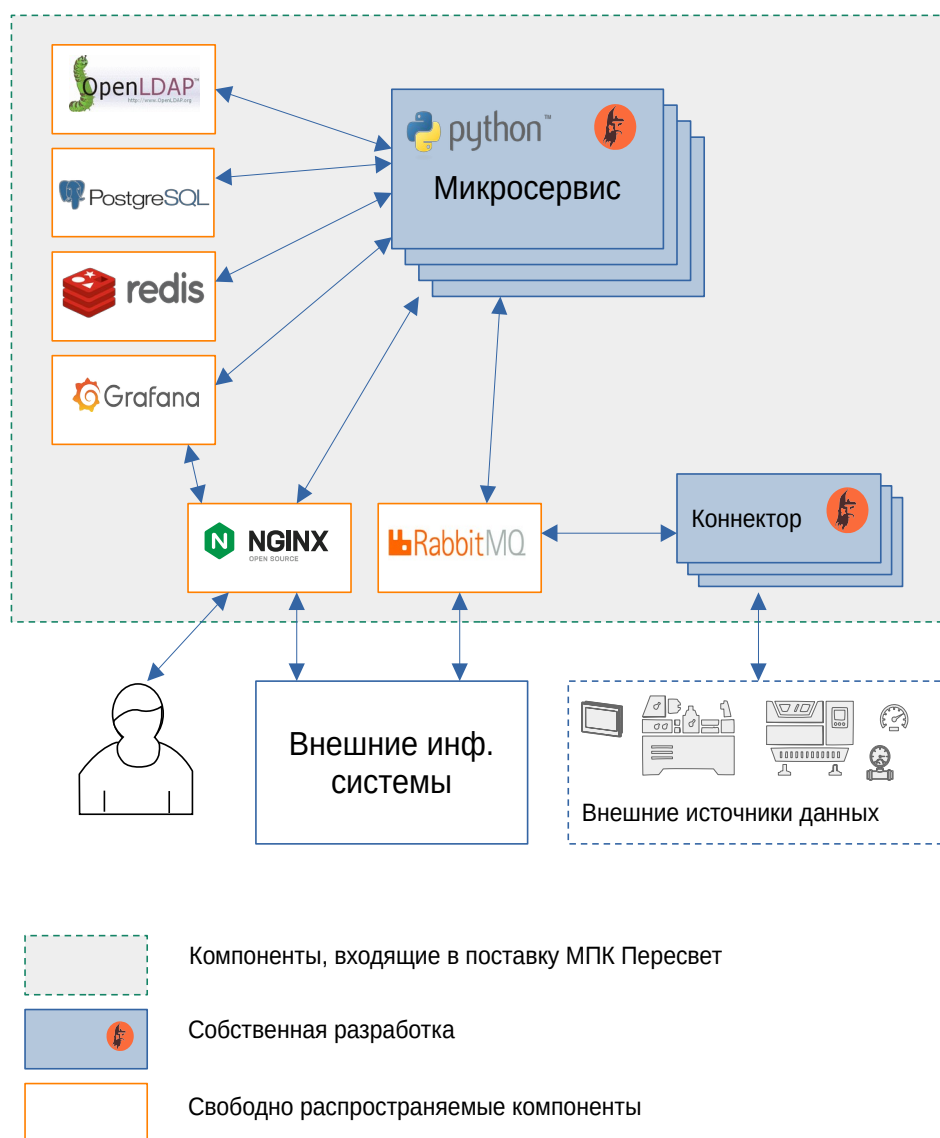


Рисунок 1: Архитектура компонентов МПК-Пересвет

Назначение компонентов, указанных на рисунке 1, приведено в таблице 1.

Таблица 1: Список компонентов МПК-Пересвет

№	Компонент	Описание
1	Микросервис	Набор микросервисов, реализующих собой основную функциональность МПК-Пересвет.
2	OpenLDAP	Иерархическая база данных. Хранение статической модели объекта, аутентификация и авторизация пользователей.
3	PostgreSQL	Исторические данные.
4	Redis	Кэш-сервер.

5	Grafana	Конструктор для построения интерфейсов пользователя.
6	Nginx	Обратный прокси. Единая точка входа в систему для всех клиентов.
7	RabbitMQ	Брокер сообщений. Общение между микросервисами, между платформой и коннекторами. Также его могут использовать внешние системы.
8	Коннектор	Специализированная программа для автоматического сбора данных с разных источников. Для каждого протокола обмена данными — свой коннектор.

Микросервис

Функциональное ядро МПК-Пересвет выполнено в виде набора микросервисов, общающихся между собой с помощью брокера сообщений RabbitMQ.

Общее правило: для каждой сущности, присутствующей в модели объекта, создаётся свой набор микросервисов, который «умеет» работать с данной сущностью. В общем случае для каждой сущности может создаваться набор из четырёх микросервисов:

<сущность>_api_crud — микросервис реализует собой REST API для задач создания-чтения-обновления-удаления экземпляров данной сущности в модели, проверяет корректность входных данных.

<сущность>_model_crud — микросервис, работающий со статической моделью (LDAP); только этот микросервис может создавать-обновлять-удалять данные из иерархии.

<сущность>_app — микросервис, реализующий основную функциональность сущности; например, для методов это: логика вызова методов на исполнение, возврат результатов работы методов и т. д.

<сущность>_app_api — микросервис, обеспечивающий REST API для клиентов сущности; например, для тегов это — API для чтения/записи данных.

OpenLDAP

Иерархическая модель технического объекта и всей созданной информационной системы, так как в модели регистрируются, кроме объектов, хранилища данных и привязка к ним тегов; коннекторы, поставляющие данные в платформу; расписания, генерирующие события и т. д.

Иерархия может иметь любой уровень вложенности. В самом простейшем случае объект может моделироваться простым линейным списком тегов.

Кроме того, OpenLDAP выполняет функции аутентификации и авторизации пользователей.

PostgreSQL

Наиболее часто используемая СУБД для хранения исторических данных.

Архитектура МПК-Пересвет позволяет использовать для хранения истории любые базы данных, при создании для работы с ними соответствующего микросервиса. Более того, возможна одно-временная поддержка нескольких баз данных в рамках одной модели, причём базы данных могут быть разного типа.

Но в большинстве случаев достаточно PostgreSQL.

Для каждого тега, зарегистрированного в модели, в базе данных создаётся отдельная таблица с названием «t_<id тега>». В этой таблице — 4 поля:

id — автоинкрементный идентификатор записи.

x — целочисленная метка времени.

y — значение тега.

q — метка качества.

Также создаётся отдельная таблица и для каждой тревоги, только префикс в названии - «a_».

Redis

Redis используется как кэш-сервер. Выполняет важную роль во взаимодействии нескольких экземпляров одного и того же микросервиса.

Grafana

Конструктор для создания пользовательского интерфейса. Преимущества:

1. Стандартный инструмент, а не специализированная разработка.
2. Обилие существующих виджетов.
3. Возможность строить BI на базе Apache e-Charts.
4. Работа с картами.
5. Отображение видеопотоков с камер.
6. Объединение в рамках одного пользовательского интерфейса множества информационных систем предприятия.

Nginx

Обратный прокси, обеспечивающий единую точку входа в систему.

Позволяет балансировать нагрузку между экземплярами микросервисов.

Также выполняет роль TLS-прокси.

RabbitMQ

Брокер сообщений. Все микросервисы общаются между собой исключительно через этот брокер.

Очереди сообщений по умолчанию конфигурируются таким образом, что возможно запускать одновременно несколько экземпляров одного и того же сервиса для обработки большого потока сообщений.

Допустимо создание новых микросервисов, новые микросервисы могут добавлять в модель работу со своими сущностями.

Коннектор

Коннектор — это специализированная программа, которая работает с внешним источником данных по определённому протоколу передачи данных. Можно создать коннектор, который будет работать по нескольким протоколам, но лучше придерживаться двух правил:

1. Один коннектор — один протокол.
2. Один коннектор — один источник данных.

Требования к аппаратному обеспечению

Требования к аппаратному обеспечению полностью зависят от масштабов создаваемой модели технического объекта и от потоков данных как внутрь платформы, так и из неё.

Платформа МПК-Пересвет может работать как на одноплатных компьютерах типа Raspberry Pi, так и на кластерах серверов в случае больших потоков данных.

Минимальные требования: Raspberry Pi 4 Model B с 4 Гб оперативной памяти и флэшкой на 32 Гб.

Администрирование

Файл с переменными

Файл «docker → compose → .cont_one_app.env» содержит значения всех переменных, которые используются при сборке docker-контейнеров.

Значение ключевых переменных (число перед двоеточием — номер строки в файле):

```
5: TIME_ZONE=Europe/Moscow
```

Переменная влияет на работу с метками времени в платформе МПК-Пересвет. Так как правильная работа с метками времени — ключевой вопрос работы всей системы, выставляйте эту переменную на ту временную зону, в которой работает контейнер с основными микросервисами МПК-Пересвет.

Следующие переменные хранят номера публикуемых портов. Если на хосте, на котором запускается МПК-Пересвет, какие-то из указанных в этих переменных порты уже заняты, то необходимо заменить значения указанных переменных перед сборкой контейнеров.

```
89: LDAP_PORT=389
```

Стандартный порт LDAP-сервера.

```
96: NGINX_HTTP_PORT=80
97: NGINX_HTTPS_PORT=443
```

Порты по умолчанию для всех веб-серверов.

```
105: POSTGRES_PORT=5432
```

Порт доступа к базе PostgreSQL.

```
118: RABBIT_PORT=5672
119: RABBIT_UI_PORT=15672
120: RABBIT_MQTT_PORT=1883
```

Порты, используемые брокером сообщений RabbitMQ. Порт 1883 используется плагином MQTT для связи с коннекторами.

```
132: PORT_REDIS=6379
133: PORT_REDIS_INSIGHT=8001
```

Порты, используемые кэш-сервером Redis.

Docker-сеть

По умолчанию между docker-контейнерами создаётся сеть:

```
networks:
  peresvet:
    driver: bridge
    ipam:
      config:
        - subnet: 10.66.0.0/16
          gateway: 10.66.0.1
```

В случае конфликта сети необходимо установить другие параметры.

Запуск МПК-Пересвет

HTTP

Для простых случаев использования платформы, допустим, для работы внутри надёжно защищённых сетей можно использовать работу по протоколу http. Для этого необходимо запустить МПК-Пересвет скриптом

```
$ ./run_one_app.sh
```

После старта МПК-Пересвет доступ к пользовательскому интерфейсу возможен по адресу <http://localhost/grafana>:

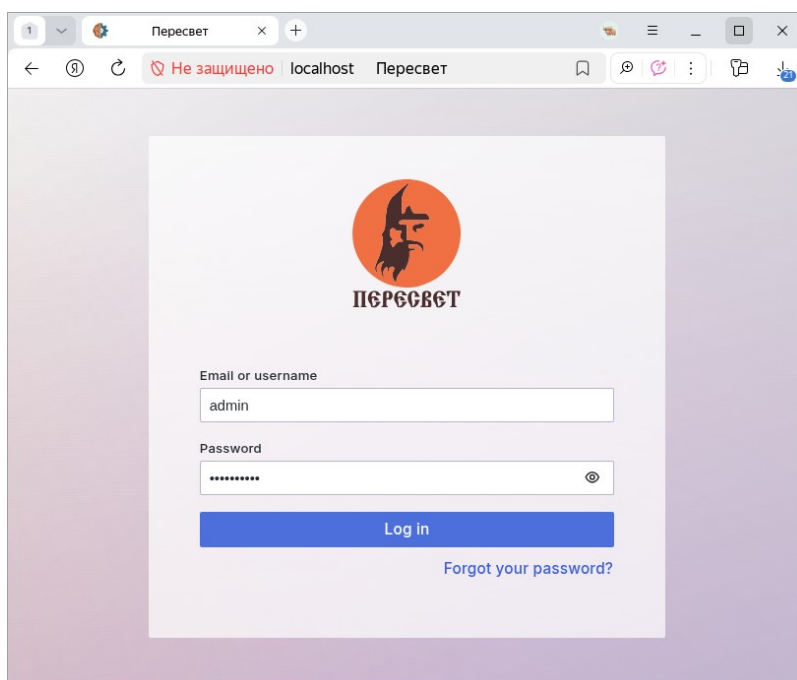


Рисунок 2: Экран платформы. Протокол HTTP.

Внимание! При первом запуске страницы платформы пароль пользователя «admin» также «admin». Сразу будет предложено сменить пароль.

HTTPS. Самоподписанные сертификаты

Первым шагом генерируем самоподписанные сертификаты:

Заходим в каталог «certificates» и запускаем на исполнение скрипт:

```
$ ./gen_crt.sh
```

Этот скрипт запускает поочерёдно три других скрипта, расположенных в этом же каталоге, с параметрами по умолчанию:

```
01. gen_root.sh
02. gen_server.sh
03. gen_client.sh
```

01. gen_root.sh — генерация сертификата удостоверяющего центра.

02. gen_server.sh — генерация сертификата для сервера.

03. gen_client.sh — генерация сертификата для клиента.

В результате работы скрипт «gen_crt.sh» создаст структуру каталогов с сертификатами:

```
tls
├── clients
│   └── CN=<guid>
├── rootCA
└── servers
    └── CN=<guid>
```

В качестве имени сервера по умолчанию будет использовано имя хоста, на котором запускался скрипт.

Если необходимо сгенерировать сертификаты для другого сервера, необходимо запустить отдельно скрипт:

```
$ ./02.\ gen_server.sh -h <server_name> -d <days> -k <key_length> -s <subj>
```

где:

h — имя сервера, для которого генерируется сертификат; по умолчанию — имя локального хоста;

d — срок действия сертификата в днях; по умолчанию — 3654 дней (10 лет);

k — длина ключа в битах; по умолчанию — 4096

s — информация, вносимая в сертификат; по умолчанию — /CN=<server_name>.

После генерации сертификатов запускаем МПК-Пересвет на исполнение скриптом

```
$ ./run_one_app_ssl.sh
```

HTTPS. Сертификаты Let's Encrypt

Сертификаты от Let's Encrypt можно получить только для серверов, доступных из интернета.

В этом случае: предполагаем, что выполняем указанные ниже действия на сервере, доступном в интернете под именем «server.ru».

Запускаем скрипт:

```
$ ./run_one_app_ssl_letsencrypt_generate_certificates.sh server.ru
```

В качестве параметра передаём скрипту имя сервера, как он виден в сети интернет!

В процессе работы скрипта следуем его указаниям.

После работы скрипта запускаем другой скрипт:

```
$ ./run_one_app_ssl_letsencrypt.sh
```

В результате:

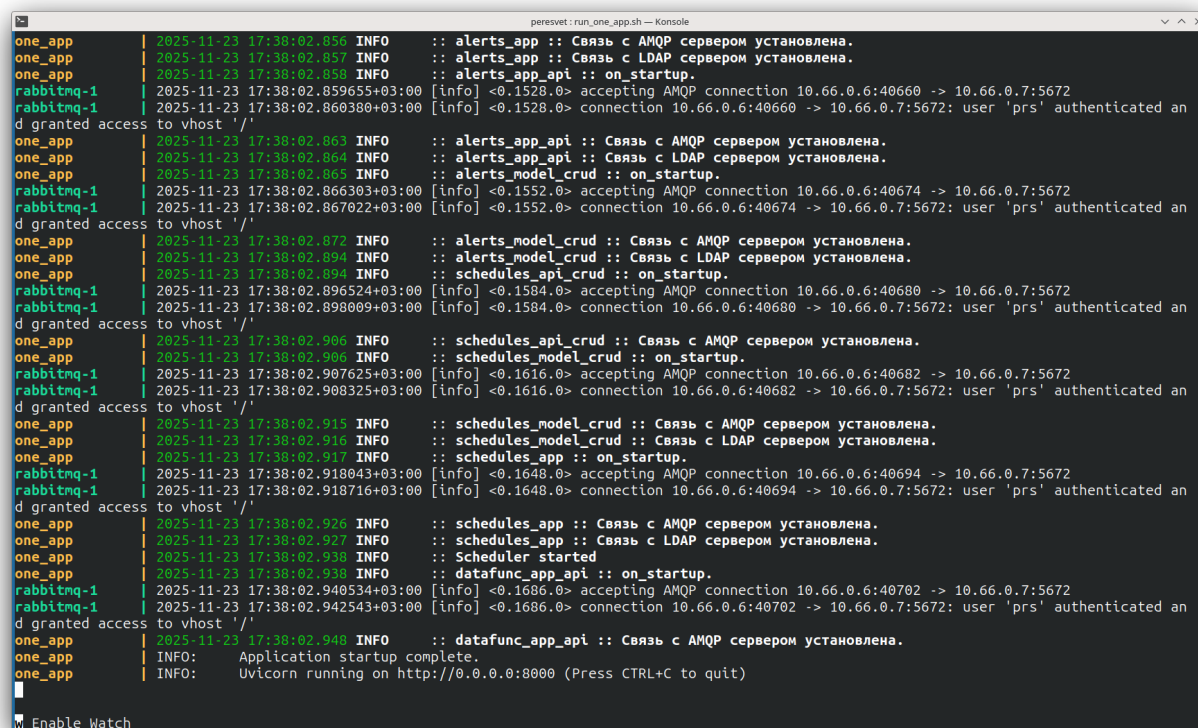
1. Запускается МПК-Пересвет.
2. При перезапуске сервера МПК-Пересвет перезапускается.
3. Каждые 11 недель происходит автоматическое обновление сертификатов.

Мониторинг состояния

Мониторинг состояния компонентов МПК-Пересвет обуславливается способом его установки.

Основной способ установки, доступный по умолчанию — в виде docker-контейнеров, поэтому отслеживать состояние платформы возможно с помощью средств и инструментов docker, docker compose, а также сторонних инструментов типа Portainer.

Успешный запуск платформы можно определить по выводу логов её сервисов в консоль. При этом экран имеет приблизительно такой вид:



```
peresvet: run_one_app.sh — Konsole
one_app | 2025-11-23 17:38:02.856 INFO :: alerts_app :: Связь с AMQP сервером установлена.
one_app | 2025-11-23 17:38:02.857 INFO :: alerts_app :: Связь с LDAP сервером установлена.
one_app | 2025-11-23 17:38:02.858 INFO :: alerts_app_api :: on_startup.
rabbitmq-1 | 2025-11-23 17:38:02.859655+03:00 [info] <0.1528.0> accepting AMQP connection 10.66.0.6:40660 -> 10.66.0.7:5672
rabbitmq-1 | 2025-11-23 17:38:02.860380+03:00 [info] <0.1528.0> connection 10.66.0.6:40660 -> 10.66.0.7:5672: user 'prs' authenticated and
d granted access to vhost '/'
one_app | 2025-11-23 17:38:02.863 INFO :: alerts_app_api :: Связь с AMQP сервером установлена.
one_app | 2025-11-23 17:38:02.864 INFO :: alerts_app_api :: Связь с LDAP сервером установлена.
one_app | 2025-11-23 17:38:02.865 INFO :: alerts_model_crud :: on_startup.
rabbitmq-1 | 2025-11-23 17:38:02.866303+03:00 [info] <0.1552.0> accepting AMQP connection 10.66.0.6:40674 -> 10.66.0.7:5672
rabbitmq-1 | 2025-11-23 17:38:02.867022+03:00 [info] <0.1552.0> connection 10.66.0.6:40674 -> 10.66.0.7:5672: user 'prs' authenticated and
d granted access to vhost '/'
one_app | 2025-11-23 17:38:02.872 INFO :: alerts_model_crud :: Связь с AMQP сервером установлена.
one_app | 2025-11-23 17:38:02.894 INFO :: alerts_model_crud :: Связь с LDAP сервером установлена.
one_app | 2025-11-23 17:38:02.894 INFO :: schedules_api_crud :: on_startup.
rabbitmq-1 | 2025-11-23 17:38:02.896524+03:00 [info] <0.1584.0> accepting AMQP connection 10.66.0.6:40680 -> 10.66.0.7:5672
rabbitmq-1 | 2025-11-23 17:38:02.898009+03:00 [info] <0.1584.0> connection 10.66.0.6:40680 -> 10.66.0.7:5672: user 'prs' authenticated and
d granted access to vhost '/'
one_app | 2025-11-23 17:38:02.906 INFO :: schedules_api_crud :: Связь с AMQP сервером установлена.
one_app | 2025-11-23 17:38:02.906 INFO :: schedules_model_crud :: on_startup.
rabbitmq-1 | 2025-11-23 17:38:02.907625+03:00 [info] <0.1616.0> accepting AMQP connection 10.66.0.6:40682 -> 10.66.0.7:5672
rabbitmq-1 | 2025-11-23 17:38:02.908325+03:00 [info] <0.1616.0> connection 10.66.0.6:40682 -> 10.66.0.7:5672: user 'prs' authenticated and
d granted access to vhost '/'
one_app | 2025-11-23 17:38:02.915 INFO :: schedules_model_crud :: Связь с AMQP сервером установлена.
one_app | 2025-11-23 17:38:02.916 INFO :: schedules_model_crud :: Связь с LDAP сервером установлена.
one_app | 2025-11-23 17:38:02.917 INFO :: schedules_app :: on_startup.
rabbitmq-1 | 2025-11-23 17:38:02.918043+03:00 [info] <0.1648.0> accepting AMQP connection 10.66.0.6:40694 -> 10.66.0.7:5672
rabbitmq-1 | 2025-11-23 17:38:02.918716+03:00 [info] <0.1648.0> connection 10.66.0.6:40694 -> 10.66.0.7:5672: user 'prs' authenticated and
d granted access to vhost '/'
one_app | 2025-11-23 17:38:02.926 INFO :: schedules_app :: Связь с AMQP сервером установлена.
one_app | 2025-11-23 17:38:02.927 INFO :: schedules_app :: Связь с LDAP сервером установлена.
one_app | 2025-11-23 17:38:02.938 INFO :: Scheduler started
one_app | 2025-11-23 17:38:02.938 INFO :: datafunc_app_api :: on_startup.
rabbitmq-1 | 2025-11-23 17:38:02.940534+03:00 [info] <0.1686.0> accepting AMQP connection 10.66.0.6:40702 -> 10.66.0.7:5672
rabbitmq-1 | 2025-11-23 17:38:02.942543+03:00 [info] <0.1686.0> connection 10.66.0.6:40702 -> 10.66.0.7:5672: user 'prs' authenticated and
d granted access to vhost '/'
one_app | 2025-11-23 17:38:02.948 INFO :: datafunc_app_api :: Связь с AMQP сервером установлена.
one_app | INFO: Application startup complete.
one_app | INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

Рисунок 3: Лог успешного запуска МПК-Пересвет

Признаком успешного запуска МПК-Пересвет является наличие двух записей от каждого микро-сервиса об успешном соединении с сервером AMQP (брокер RabbitMQ) и OpenLDAP, а также записей «Application startup complete» и «Uvicorn running on <http://0.0.0.0:8000>».

В случае возникновения ошибок сообщения о них будут появляться в консоли, отмеченные красным цветом.

Лог контейнера с микросервисами платформы можно просмотреть с помощью команды

```
$ docker logs one_app
```

Первые записи лога будут, скорее всего, содержать ошибки соединения с брокером RabbitMQ:

```
2025-11-23 17:37:56.939 DEBUG : src.common.base_svc._init_.34 :: dataStorages_app_postgresql :: Начало инициализации сервиса dataStorages_app_postgresql.
2025-11-23 17:37:56.940 DEBUG : src.common.base_svc._init_.47 :: dataStorages_app_postgresql :: Смена петли событий...
2025-11-23 17:37:57.005 DEBUG : src.common.base_svc._init_.34 :: methods_app :: Начало инициализации сервиса methods_app.
2025-11-23 17:37:57.005 DEBUG : src.common.base_svc._init_.47 :: methods_app :: Смена петли событий...
INFO: Started server process [1]
INFO: Waiting for application startup.
2025-11-23 17:37:57.261 INFO :: connectors_api_crud :: on_startup.
2025-11-23 17:37:57.264 INFO :: Connection to amqp://prs:*****@rabbitmq/ closed. Reconnecting after 5 seconds.
2025-11-23 17:37:57.264 ERROR :: connectors_api_crud :: Ошибка связи с брокером: [Errno 111] Connection refused
2025-11-23 17:38:02.315 INFO :: connectors_api_crud :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.316 INFO :: connectors_mqtt_app :: on_startup.
2025-11-23 17:38:02.329 INFO :: connectors_mqtt_app :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.330 INFO :: connectors_mqtt_app :: Связь с LDAP сервером установлена.
2025-11-23 17:38:02.334 INFO :: connectors_mqtt_app :: Коннектор * привязан.
2025-11-23 17:38:02.334 INFO :: connectors_model_crud :: on_startup.
2025-11-23 17:38:02.344 INFO :: connectors_model_crud :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.345 INFO :: connectors_model_crud :: Связь с LDAP сервером установлена.
2025-11-23 17:38:02.349 INFO :: connectors_app_api :: on_startup.
2025-11-23 17:38:02.355 INFO :: connectors_app_api :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.355 INFO :: alerts_api_crud :: on_startup.
2025-11-23 17:38:02.362 INFO :: alerts_api_crud :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.362 INFO :: dataStorages_model_crud :: on_startup.
2025-11-23 17:38:02.370 INFO :: dataStorages_model_crud :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.371 INFO :: dataStorages_model_crud :: Связь с LDAP сервером установлена.
2025-11-23 17:38:02.371 INFO :: dataStorages_app_postgresql :: on_startup.
2025-11-23 17:38:02.395 INFO :: dataStorages_app_postgresql :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.395 INFO :: dataStorages_app_postgresql :: Связь с LDAP сервером установлена.
2025-11-23 17:38:02.600 INFO :: dataStorages_app_postgresql :: Связь с базой данных f7905474-7824-103f-8d2a-bd0f47223918 установлена.
2025-11-23 17:38:02.707 INFO :: methods_api_crud :: on_startup.
2025-11-23 17:38:02.715 INFO :: methods_api_crud :: Связь с AMQP сервером установлена.
2025-11-23 17:38:02.716 INFO :: methods_model_crud :: on_startup.
```

Рисунок 4: Лог контейнера МПК-Пересвет

Это не является ошибкой, так как брокер стартует медленнее основных сервисов, но сервисы продолжают в цикле попытки соединения с брокером до тех пор, пока соединение не будет успешным.

Хорошей практикой для мониторинга состояния контейнеров является использование какого-либо инструмента с UI, например — Portainer.

Конфигуратор модели

После запуска МПК-Пересвет необходимо создать модель объекта. Для этого служит «Конфигуратор», реализованный в виде панели в Grafana. Панель «Конфигуратор» является панелью по умолчанию для пользователя «admin».

Здесь и далее: на рисунках в строке адреса указан `bdhome:3000`.

Вместо этого пишем `http://localhost/grafana`

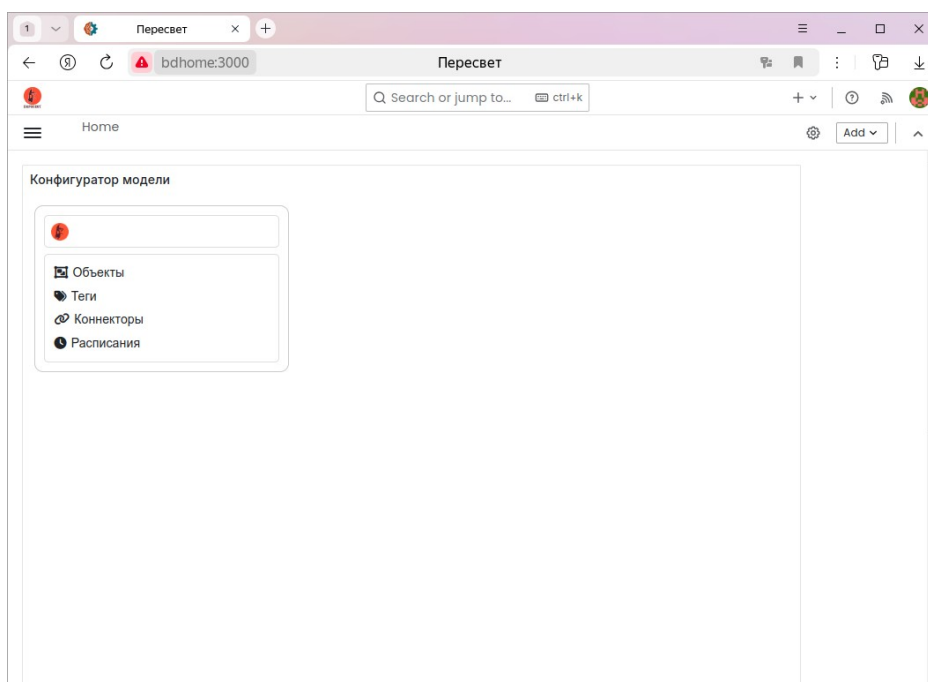


Рисунок 5: Экран "Конфигуратор"

При первом запуске в низу экрана автоматически добавляется панель, которую можно удалить:

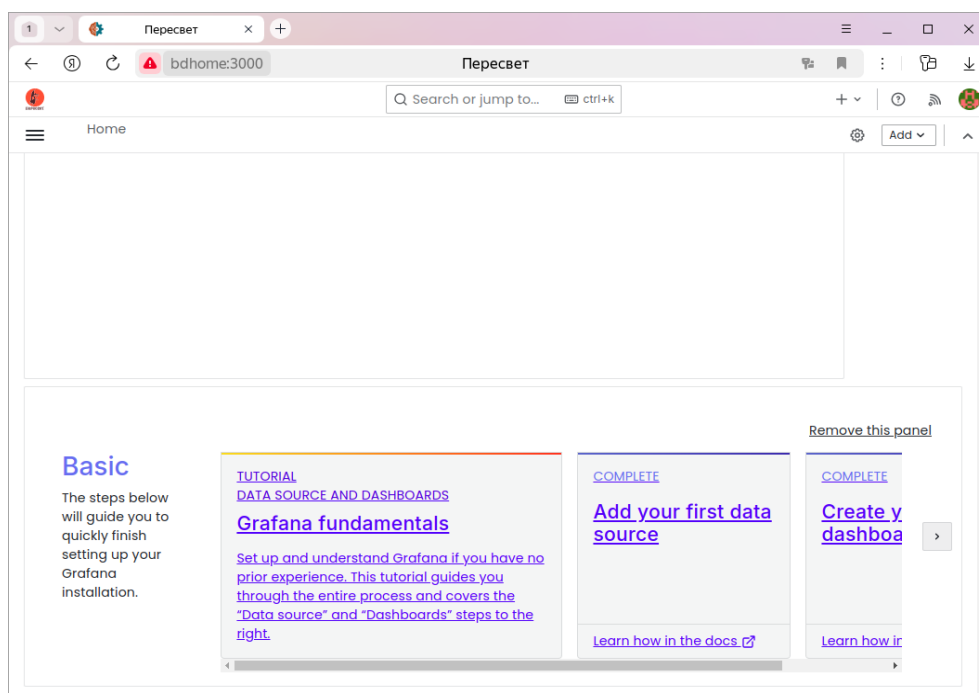


Рисунок 6: Экран при первом запуске

Папки модели

Создаваемая модель объекта и информационной системы имеет иерархическую структуру.

На верхнем уровне иерархии находятся узлы, в которых создаются экземпляры соответствующих сущностей.

Объекты

Внутри этой папки создаётся основная иерархия - модель объекта. Иерархия имеет любой уровень вложенности и может включать в себя объекты, теги, тревоги, методы.

Теги

Наиболее простой случай модели объекта - это линейный список его параметров (тегов). В таком случае теги можно создавать, не привязывая их к какому-либо объекту. Для этого случая служит отдельная папка Теги.

Коннекторы

В этой папке создаются описания коннекторов. Параметры, задаваемые в узле-описании коннектора, зависят от конкретного коннектора и указаны в документации на данный коннектор.

Расписания

Папка для создания расписаний - генераторов событий.

Хранилища данных

Папка, в которой хранятся описания хранилищ данных. Исторические данные, факты возникновения, пропадания и квитирования тревог хранятся в этих хранилищах.

Открытая версия платформы «Пересвет» поддерживает работу только с одним хранилищем, поэтому в конфигураторе для открытой платформы отсутствует узел «Хранилища данных».

Создание новых узлов

1. Выбираем нужный узел, внутри которого хотим создать дочерний.
2. Выбираем нужную синюю кнопку и, нажав её, создаём новый экземпляр необходимой нам сущности.

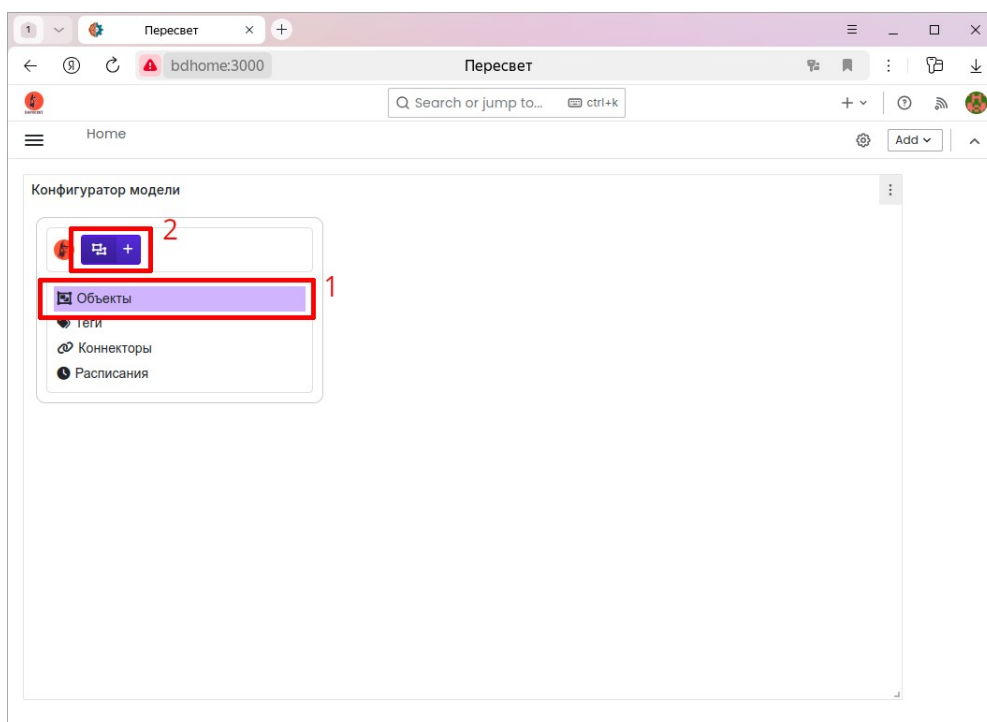


Рисунок 7: Конфигуратор. Создание нового объекта.

После нажатия кнопки в иерархию добавится новый экземпляр выбранной сущности. Узел в иерархии, соответствующий данному экземпляру, станет текущим и справа появится панель со свойствами нового узла.

Кроме того, изменится набор кнопок в верхней части конфигуратора.

Синие кнопки показывают, экземпляры каких сущностей могут быть созданы внутри выбранного узла.

Например: внутри объекта могут быть созданы другие объекты или теги; внутри тега могут быть созданы методы и тревоги.



Рисунок 8: Конфигуратор. Кнопка "Новый объект"



Рисунок 9: Конфигуратор. Кнопка "Новый тег"



Рисунок 10: Конфигуратор. Кнопка удаления выбранного узла из иерархии

При создании нового узла в иерархии с помощью конфигуратора он создаётся с параметрами по умолчанию. Поэтому имя нового узла совпадает с его идентификатором:

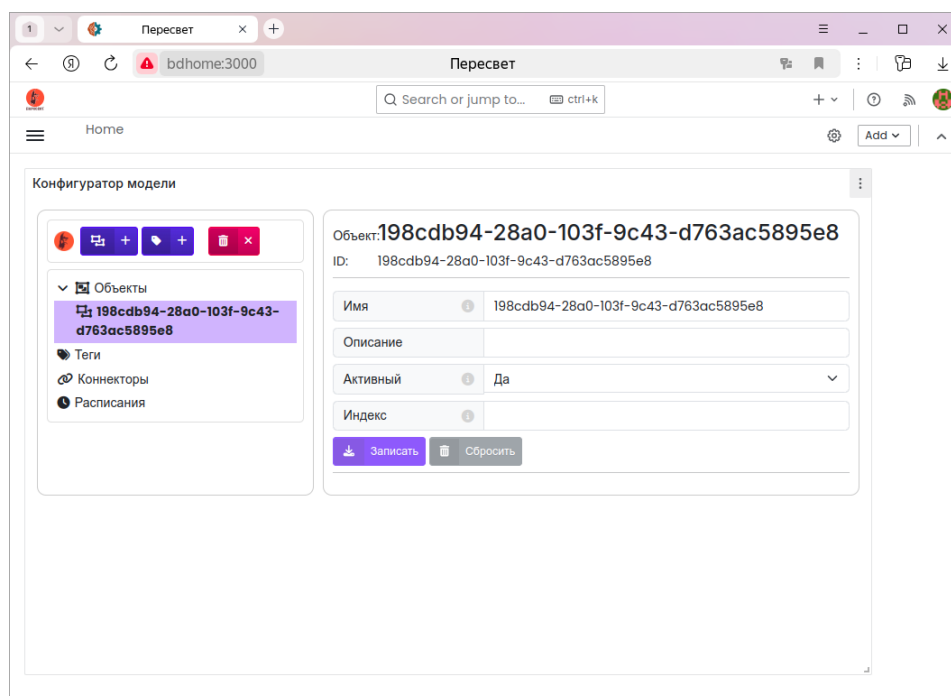


Рисунок 11: Конфигуратор. Новый объект

Редактирование свойств узла

На панели справа редактируем свойства узла. При этом изменённые свойства обводятся синей рамкой.

Если у узла изменено хотя бы одно свойство, то активизируются две кнопки:

Записать - сохранение в модели изменённых свойств и

Сбросить - отмена внесённых изменений, возврат к начальным параметрам.

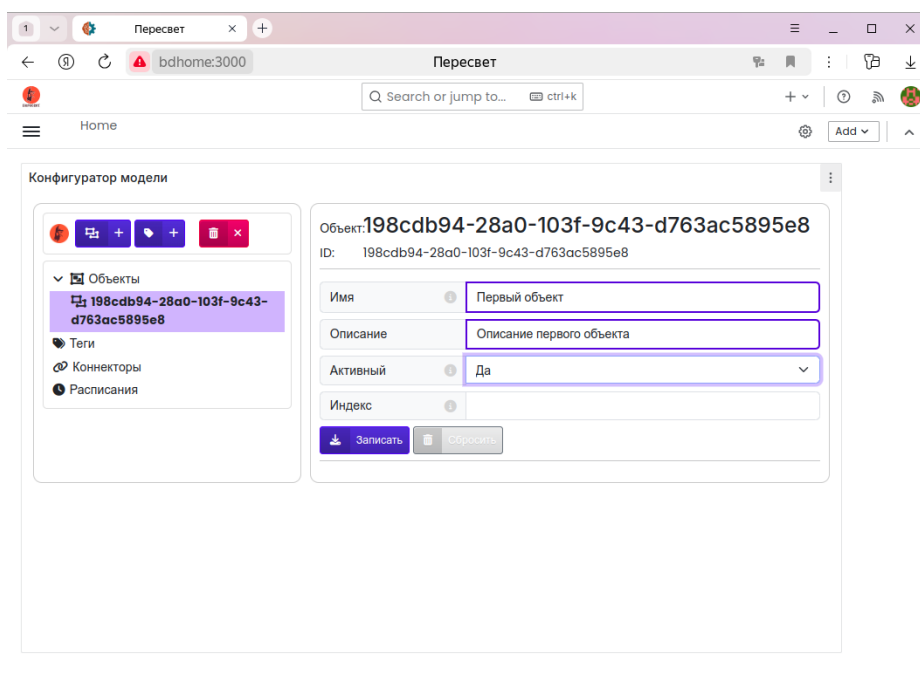


Рисунок 12: Конфигуратор. Редактирование параметров узла

Нажимаем кнопку **Записать**:

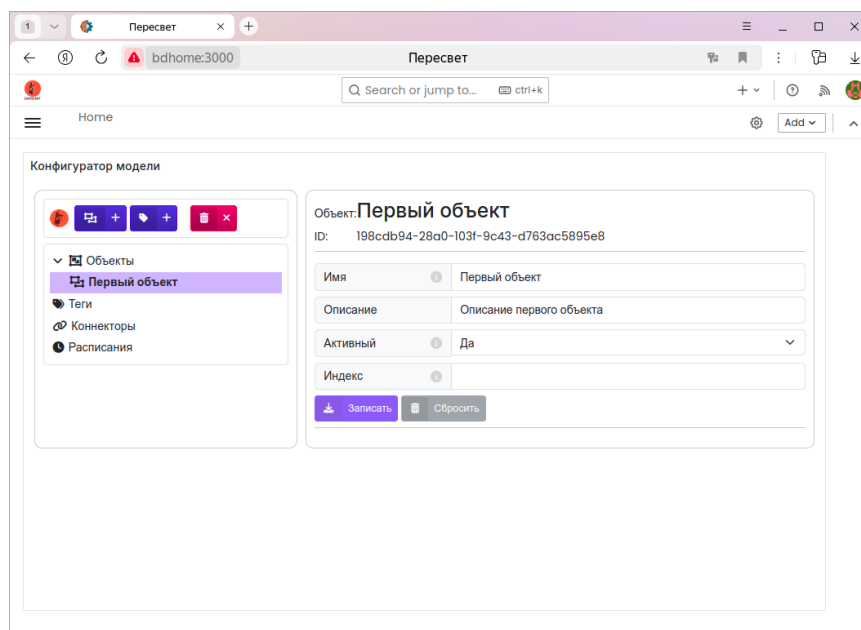


Рисунок 13: Конфигуратор. Запись изменённых параметров

Удаление узла

Выбираем нужный узел в иерархии и нажимаем красную кнопку:



Рисунок 14: Конфигуратор. Кнопка удаления выбранного узла из иерархии

Из иерархии удаляется узел и все его потомки.

Создание экранов в Grafana

По умолчанию при запуске контейнеров МПК-Пересвет регистрируется как источник данных в Grafana. Поэтому остановимся на создании экранов, предполагая, что МПК-Пересвет уже зарегистрирован как источник данных по умолчанию.

Допустим, в нашей модели есть тег с идентификатором «58993cd8-f3e9-103e-97a9-c54148266c5f», для которого в исторической базе данных накоплены данные. Тогда для создания тренда для этого тега выполним следующие шаги:

Создание нового экрана и панели

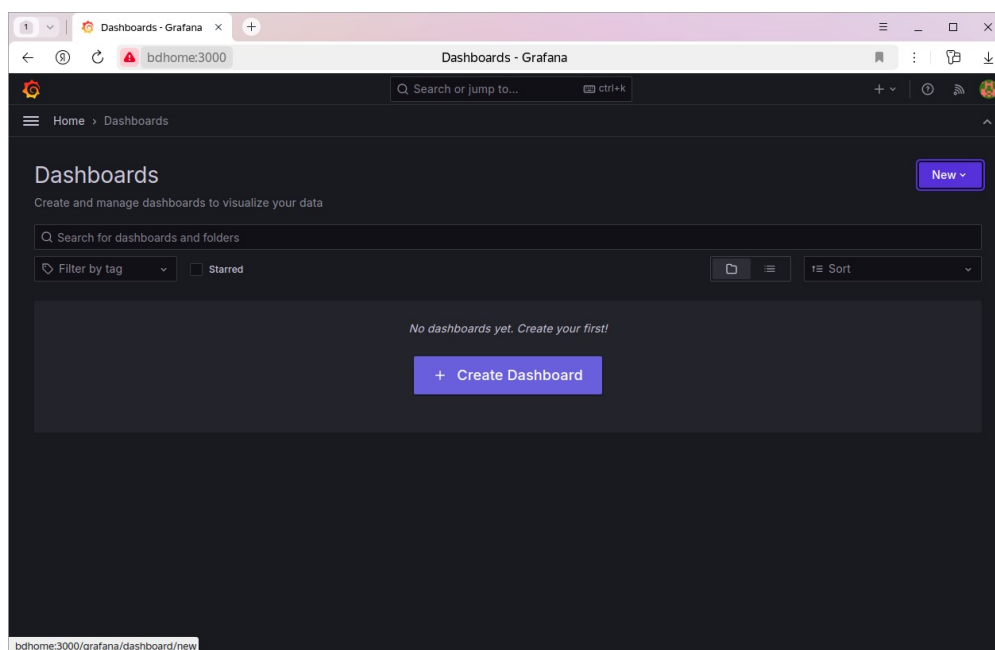


Рисунок 15: Создание нового экрана

...и новую панель:

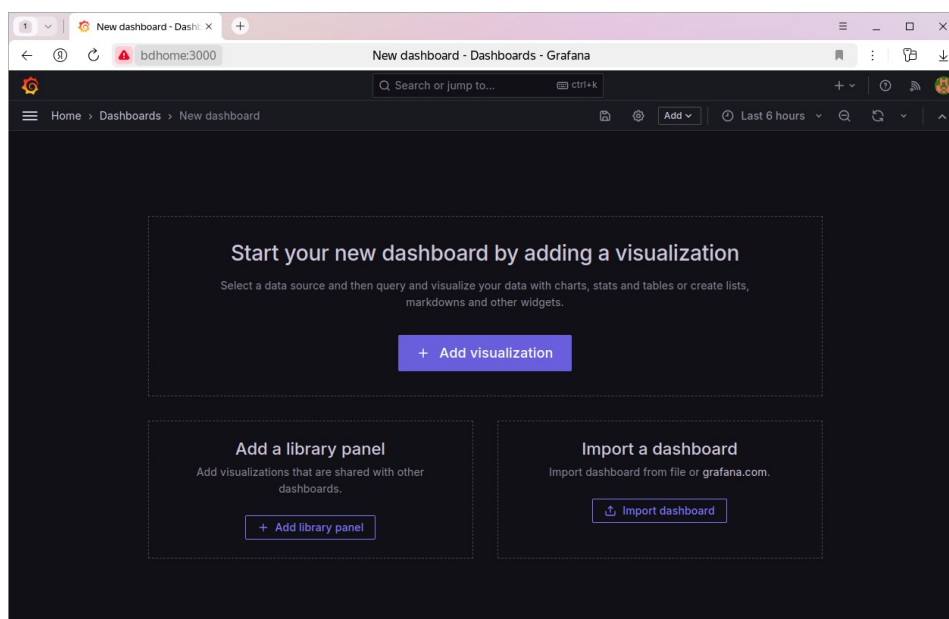


Рисунок 16: Создание новой панели

В появившемся окне выберем созданный нами источник данных:

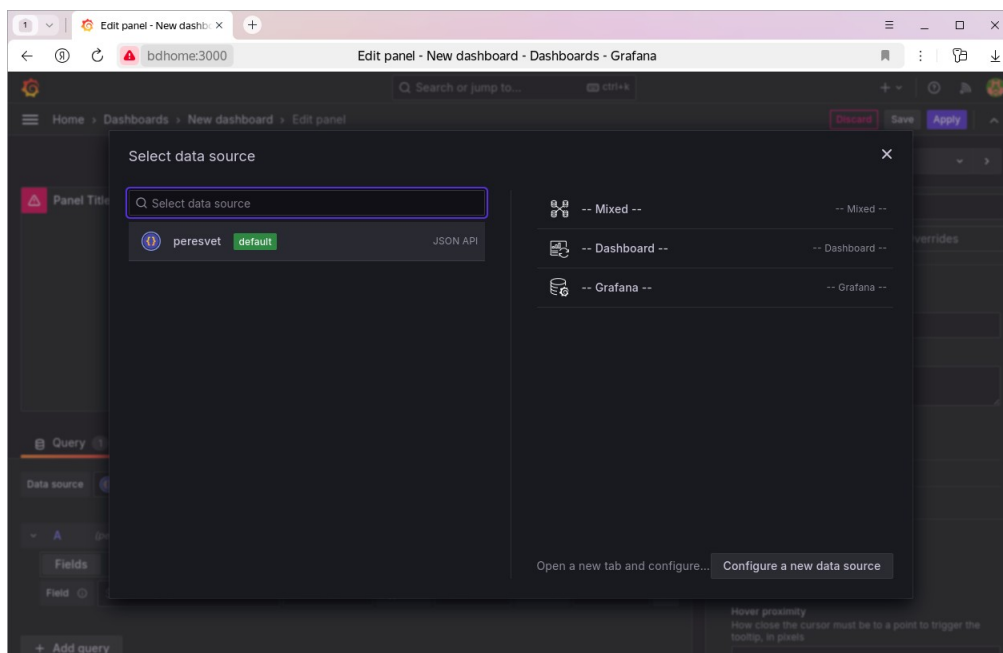


Рисунок 17: Выбор источника данных

Откроется окно настройки панели с выбранным источником данных и нужным нам типом панели - «Time Series»:

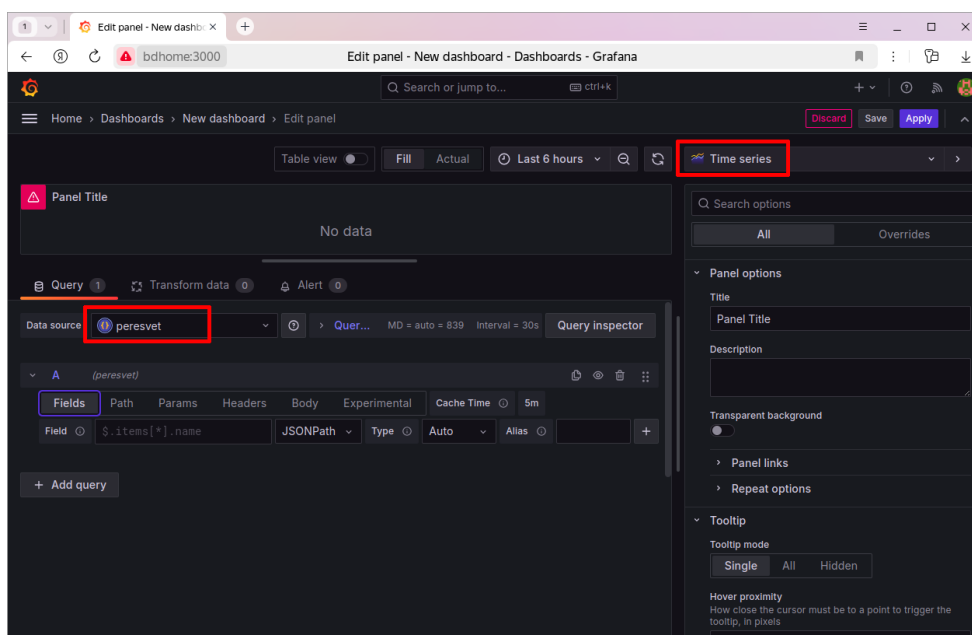


Рисунок 18: Выбор типа виджета

Настроим параметр «Path»:

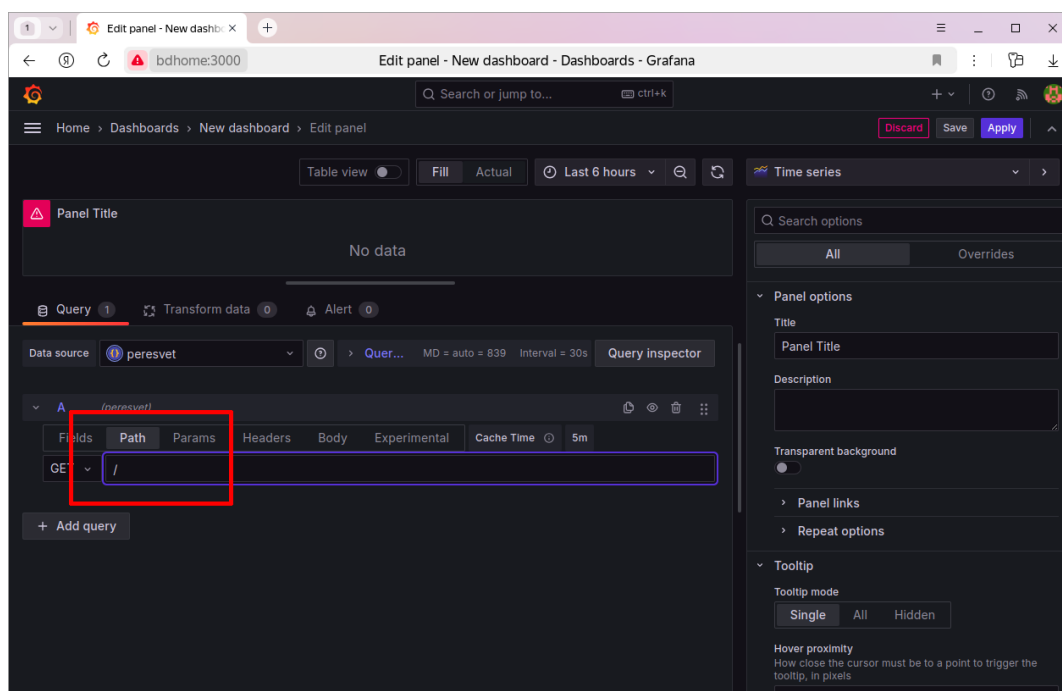


Рисунок 19: Настройка "Path"

Создадим новый параметр во вкладке «Params» (параметр, включаемый в URL запроса).

Назовём параметр q, а в качестве его значения укажем строку:

```
{"tagId":"58993cd8-f3e9-103e-97a9-c54148266c5f","format":true,"start":"$__isoFrom()","finish":"$__isoTo()"} 
```

Заметим, что в запросе указаны функции Grafana - «`__isoFrom()`» и «`__isoTo()`». Эти функции возвращают, соответственно, начало и конец выбранного пользователем на dashboard'е периода.

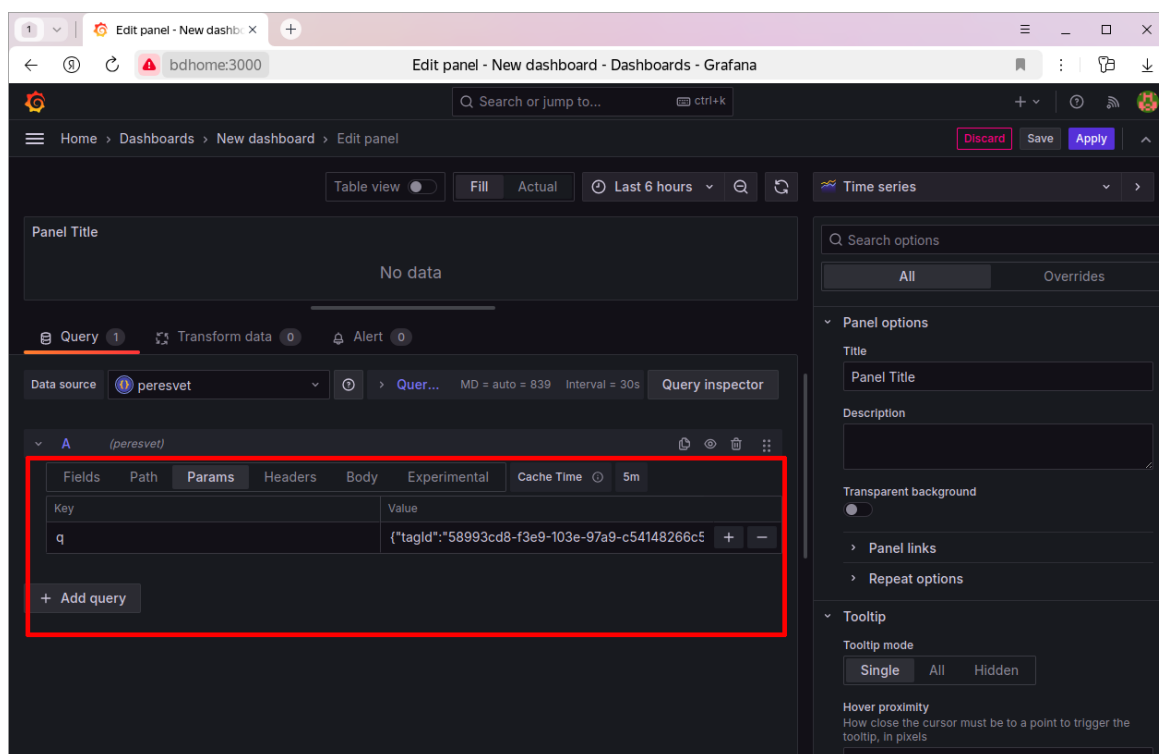


Рисунок 20: Формирование запроса

Для проверки того, что платформа возвращает данные, нажмём кнопку «Query inspector»:

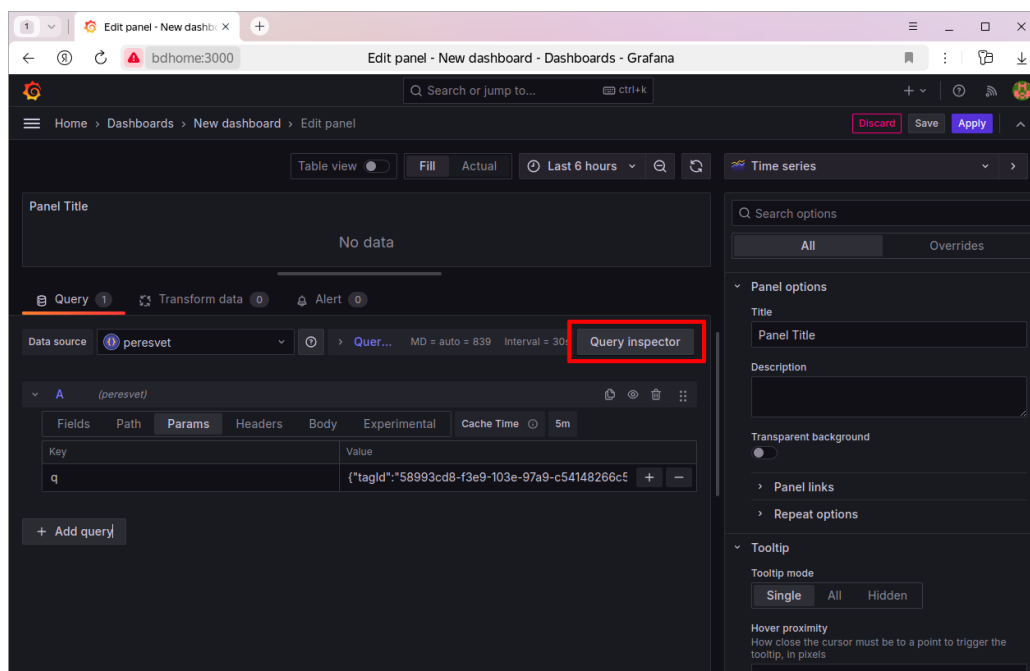


Рисунок 21: Проверка запроса

...затем Refresh:

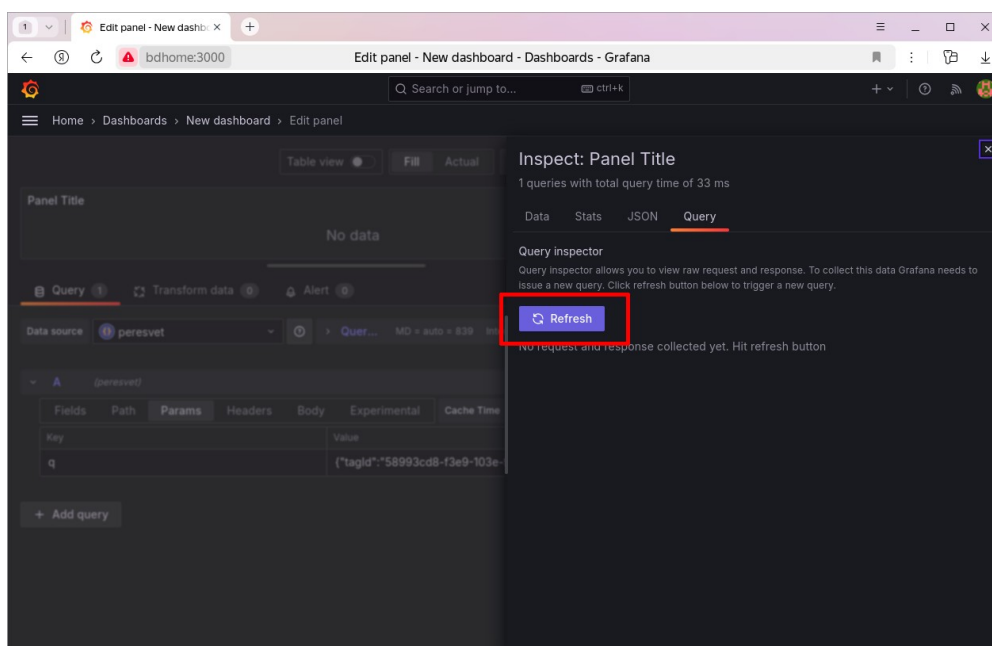


Рисунок 22: Обновление данных

...и убедимся, что платформа возвращает данные:

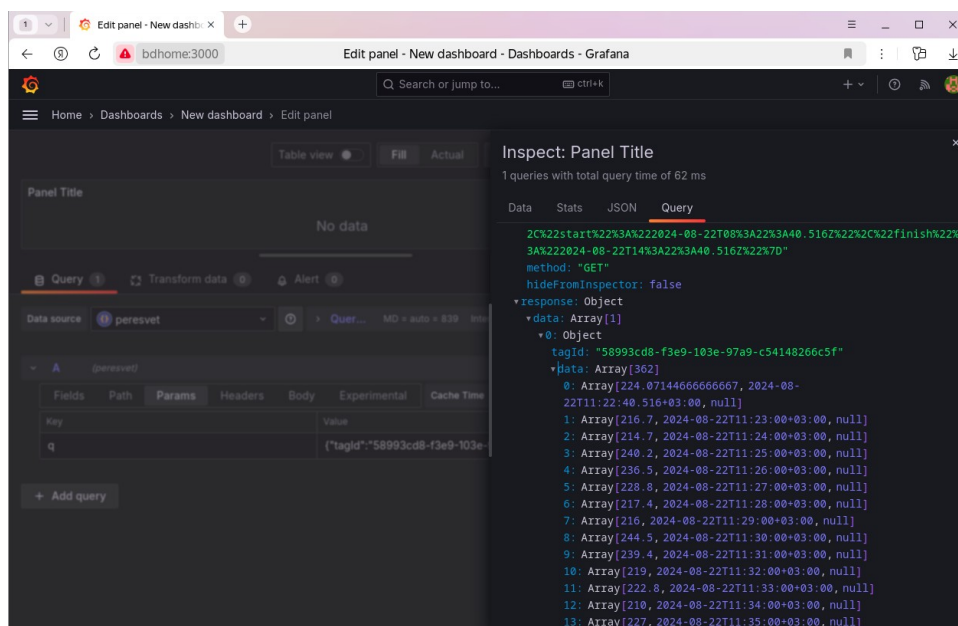


Рисунок 23: Данные из платформы

Теперь нам необходимо объяснить Grafan'е, как из полученного массива данных выбрать необходимые.

Для этого переходим в закладку «Fields» и настраиваем её так, как показано на рисунке:

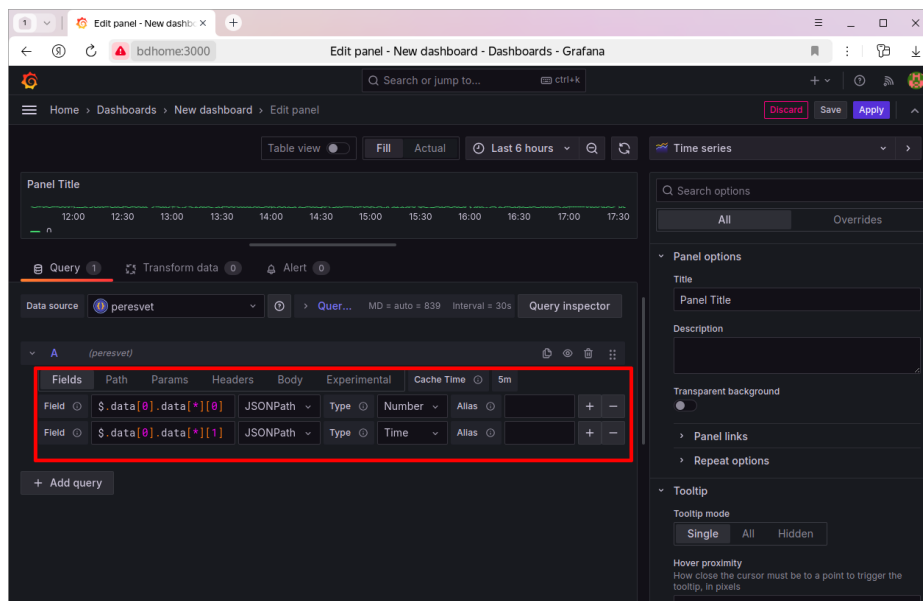


Рисунок 24: Настройка полей

Здесь верхнее поле — это значения тега, которые необходимо отобразить (синтаксис JSONPath), нижнее — метка времени.

Нажимаем кнопку «Apply» и видим тренд с данными тега:

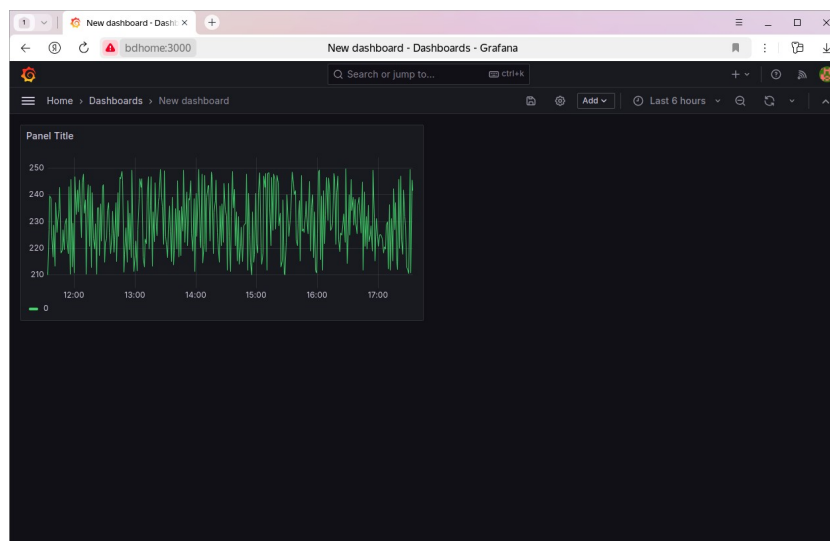


Рисунок 25: Тренд тега

Можно менять периоды, выбирать частоту обновления экрана, данные будут отображаться корректно:

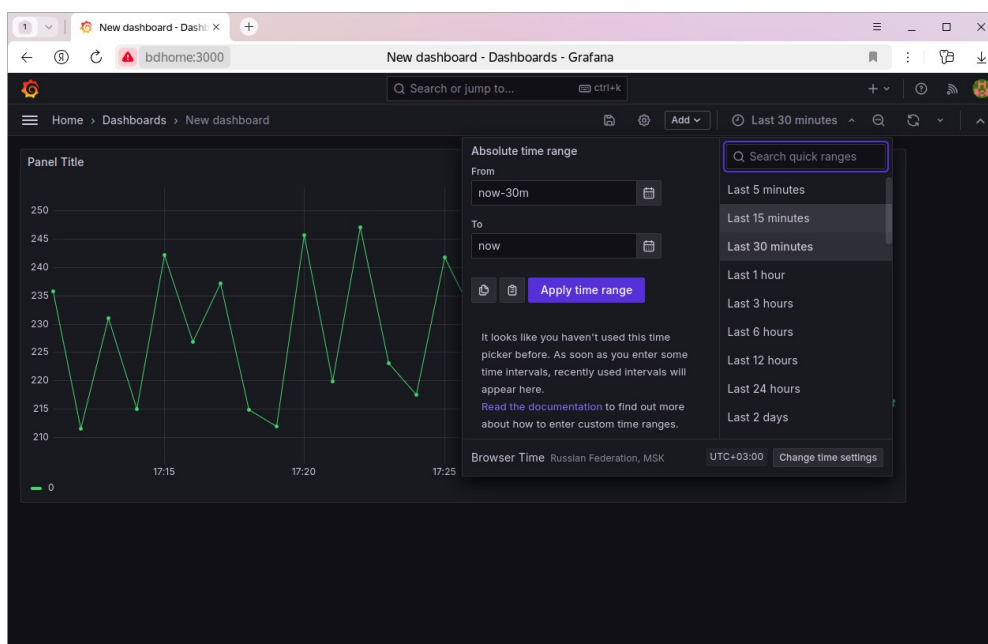


Рисунок 26: Изменение параметров

Аналогичным образом настраиваются другие виджеты в Grafana.

Ключевое знание — умение работать с историческими данными, которые предоставляет МПК-Пересвет. Работа с историческими данными подробно описана в руководстве пользователя.