

Регулярное обновление данных в проекте

Пользовательские сценарии

Стартовое состояние – пользователь находится в диспетчере данных, создан скрипт загрузки

Действие: Пользователь нажимает на кнопку "Обновление данных"

Результат: Открывается окно для создания или изменения задания на обновление данных с 5 возможными графиками: ежечасно, ежедневно, еженедельно, ежемесячно, CRON.

Действие: Пользователь выбирает график обновления нажатием на радиокнопку рядом с названием

Результат:

- Изменяются поля ввода данных согласно выбранному графику: можно ввести час для ежечасного, день для ежедневного и т.д. Всегда есть ещё одно поле ввода для времени. По сути, каждый график представляет собой вариант CRON-задания, у которого вручную задаётся только часть параметров
- Загружается иконка для выбора времени (только один раз)
- Теряются все внесенные изменения

Действие: Пользователь вводит данные для графика обновления

Результат:

- Изменения фиксируются в интерфейсе
- Изменения сбрасываются при выборе другого графика
- До подтверждения загрузки никаких взаимодействий с бэкендом не происходит

Действие: Пользователь нажимает на кнопку "Создать/Изменить"

Результат: Формируется задание на автообновление данных согласно выбранному графику и установленным пользователем параметрам

Стартовое состояние – пользователь находится в панели администратора на вкладке "Задания"

Действие: Пользователь нажимает на кнопку "Создать"

Результат: Открывается окно создания задания как в диспетчере данных, но есть возможность выбрать проект из потока, для которого будет создано задание

Действие: Пользователь нажимает на кнопку "Черновик" при активной кнопке "Потоки"

Результат: Становится доступен список черновиков пользователей, автоматически выбирается первый в списке, становится доступен список проектов для пользователя

Действие: Пользователь нажимает на кнопку "Потоки" при активной кнопке "Черновик"

Результат: Становится доступен список потоков стенда, автоматически выбирается первый в списке, становится доступен список проектов для проекта

Действие: Пользователь выбирает поток из списка потоков/черновиков

Результат: Становится доступен список проектов для потока

Настройка CRON-задания аналогична настройке из диспетчера данных

Необходимые изменения

Обеспечивать сохранность предыдущих данных при обновлении

Добавить сборщик событий

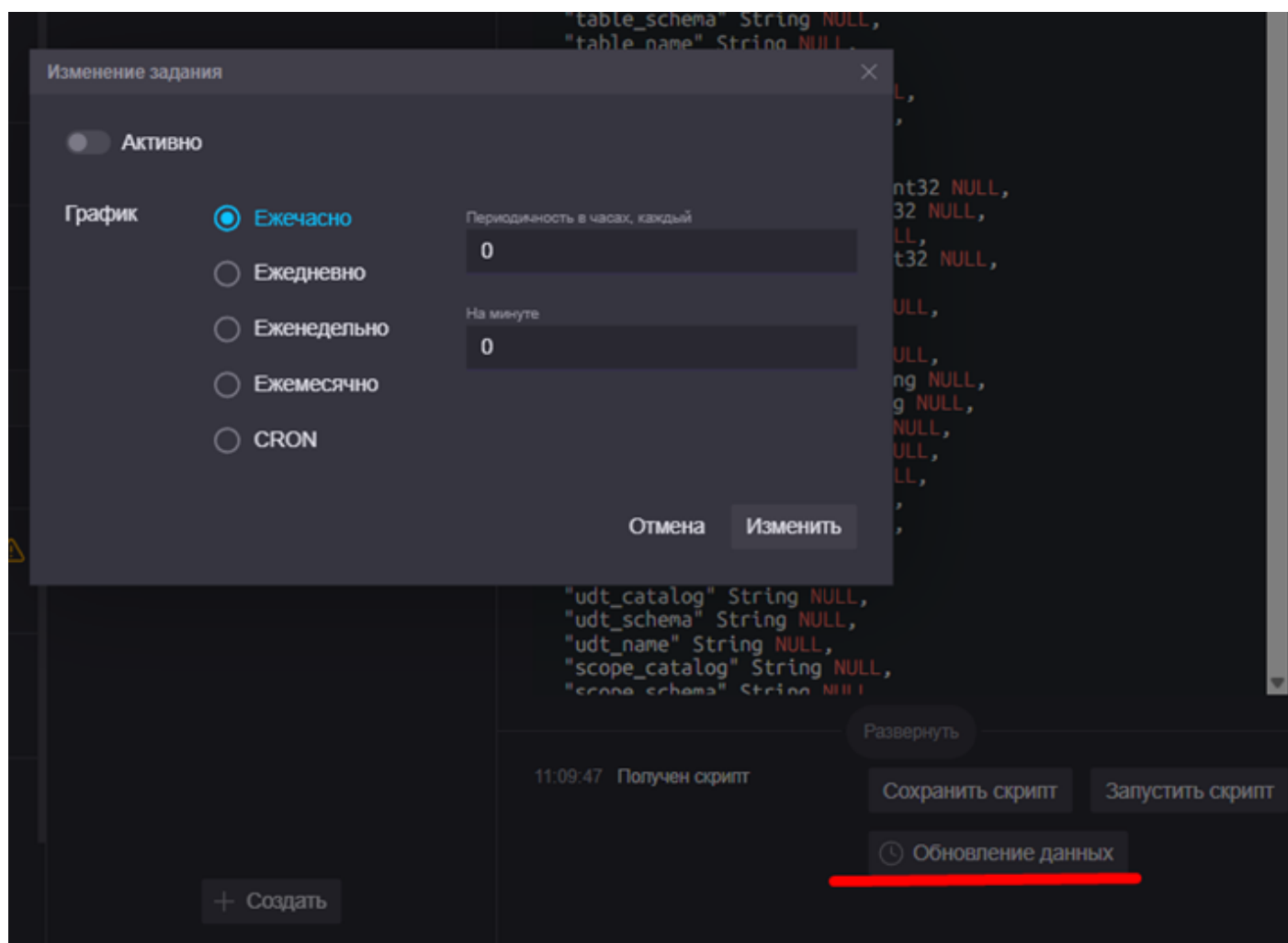
[Добавить таск-менеджер и очередь заданий](#)

[Добавить мониторинг обновлений](#)

Схема потока данных

Техническое описание

График обновления данных проекта устанавливается пользователем при нажатии на соответствующую кнопку на странице скрипта загрузки и подтверждении графика

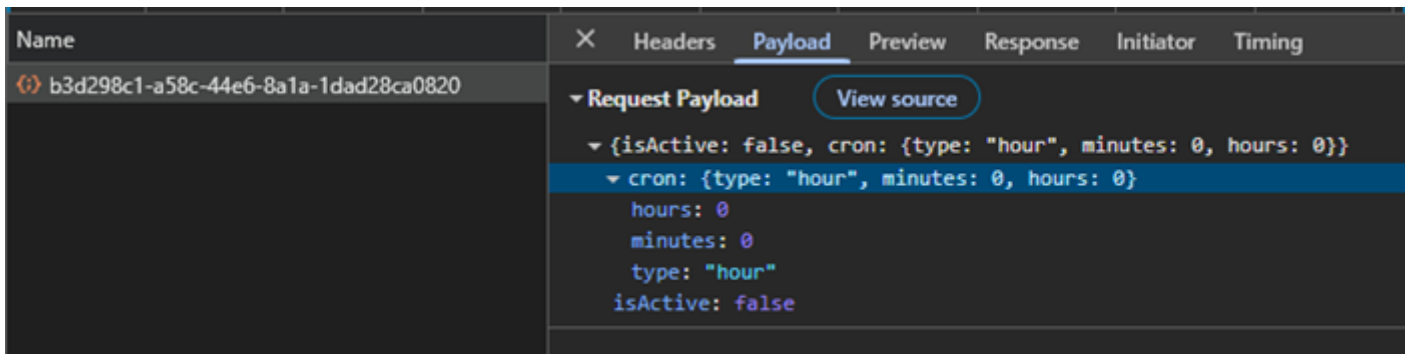


При подтверждении (кнопка "создать" или "изменить", если задание уже существует), на Nest.js сервис отправляется POST запрос, содержащий информацию по графику обновления данных и флаг (поле isActive), контролирующий активно ли обновление или нет.

Идентификатор задачи на обновление равен идентификатору проекта и добавляется как параметр в запросе

<https://fastbord-back-dev4.fb-dev.winsolutions.ru/api/v1/cronjob/{cronId}>, где cronId – идентификатор задачи (соответствует id проекта)

Тело запроса:



```
{
  "isActive": false,
  "cron": {
    "type": "hour",
    "minutes": 0,
    "hours": 0
  }
}
```

Ответ:

```
X Headers Payload Preview Response Initiator Timing
1 {
-   "kind": "cronjobItemWithId",
-   "cronjobItemWithId": {
-       "name": "111222113",
-       "cron": {
-           "type": "hour",
-           "hours": 0,
-           "minutes": 0
-       },
-       "isActive": false,
-       "type": "project_import",
-       "message": "Error",
-       "duration": "00:00:09",
-       "startedTask": "2025-06-04T14:07:34.048Z",
-       "finishedTask": "2025-06-04T14:07:43.510Z",
-       "status": "aborted",
-       "nextStartAt": "2025-06-19T09:00:00.000Z",
-       "id": "b3d298c1-a58c-44e6-8a1a-1dad28ca0820"
-   }
- }
```

```
{
  "kind": "cronjobItemWithId",
  "cronjobItemWithId": {
    "name": "111222113",
    "cron": {
      "type": "hour",
      "hours": 0,
      "minutes": 0
    },
    "isActive": false,
    "type": "project_import",
    "message": "Error",
    "duration": "00:00:09",
    "startedTask": "2025-06-04T14:07:34.048Z",
    "finishedTask": "2025-06-04T14:07:43.510Z",
    "status": "aborted",
    "nextStartAt": "2025-06-19T09:00:00.000Z",
    "id": "b3d298c1-a58c-44e6-8a1a-1dad28ca0820"
  }
}
```

Если по каким-либо причина проекта не существует, будет возвращена ошибка 604 (проект не найден)

Последовательность выполнения запроса

- инициация POST запроса от пользователя, с указанием периода обновления
- регистрация (прием) запроса в контроллере Nest.js (проверка лицензии, роли). Если роль не админ, разработчик или нет лицензии пользователю возвращается ошибка
- проверка существования проекта по id из запроса. Если проекта нет, возвращается ошибка
- отправка внутреннего запроса (в пределах сервера) от микросервиса обрабатывающего пользовательские запросы микросервису планировки задач (cron), через брокер (rabbitMQ). микросервис принимающий запросы клиентов → брокер (rabbitMQ) → микросервис планировки задач
- в микросервисе планировки (cron) создается задача и её метаданные сохраняются в базу данных postgres.

Если задача активна, она будет запускаться в соответствии с указанным временным периодом.

Обновление проекта – это функция cron, которая с помощью брокера отправляет в микросервис, отвечающий за скрипт загрузки, команду запускающую соответствующий проекту скрипт

Revision #13

Created 18 June 2025 11:00:49 by Артём

Updated 25 June 2025 13:03:59 by Артём