

# Программное обеспечение «Интеграционная платформа КЗ 2.0»

### РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

На 29 листах

Москва

2025 г.

## Содержание

| 1 | Термины и    | определения   | 2      |
|---|--------------|---|--------|
| 2 | Состав фун   | кций, реализуемых Платформой                                  | 4      |
| 3 | Описание ф   | рункций   | 5      |
|   | 3.1 Регист   | трация Пользователя   | 5      |
|   | 3.1.1        | Создание, изменение и удаление Пользователя                   | 5      |
|   | 3.2 Регист   | трация Абонентов  | 7      |
|   | 3.2.1 систем | Создание Абонента и настройка точек доступа к веб-сервисам ст | межных |
|   | 3.3 Настр    | ойка интеграционных сценариев                                 | 9      |
|   | 3.3.1        | Настройка и создание маршрута                                 | 9      |
|   | 3.3.2        | Настройка ресурсов для маршрутов                              | 15     |
|   | 3.3.3        | Настройка переменных для маршрута                             | 18     |
|   | 3.3.4        | Настройка библиотек   | 20     |
|   | 3.3.5        | Настройка компонентов   | 20     |
| 4 | Создание то  | естового сценария для демонстрации функциональности Платформы | 21     |
|   | 4.1 Созда    | ние маршрута  | 21     |
|   | 4.2 Выпол    | лнение тестового примера                                      | 24     |
| 5 | Возможные    | е ошибки  | 25     |
|   | 5.1 Польз    | овательские ошибки  | 25     |
|   | 5.1.1        | Ошибки при создании Пользователя                              | 25     |
|   | 5.1          | .1.1 Создание и изменение пользователя                        | 25     |
|   | 5.1.2        | Ошибки при создании Абонента                                  | 25     |
|   | 5.1          | .2.1 Создание и изменение абонента                            |        |
|   | 5.1.3        | Ошибки при работе с Маршрутами                                | 26     |
|   | 5.1          | .3.1 Изменение маршрута                                       | 26     |
|   | 5.1          | .3.2 Удаление маршрута  | 27     |
|   | 5.1.4        | Ошибки при работе с Переменными                               | 28     |
|   | 52 Систе     | мные ошибки   | 29     |

### 1 Термины и определения

| Термин        | Описание   |
|---------------|--|
|               | Apache Camel — открытый кроссплатформенный java-   |
|               | фреймворк, который позволяет проводить интеграцию  |
| Camel         | приложений в простой и понятной форме. Идеологически   |
| Callici       | основан на Шаблонах Интеграции Корпоративных Приложений.   |
|               | Документация фреймворка приведена на официальном портале   |
|               | https://camel.apache.org/  |
| API           | Интерфейс программирования приложений  |
| HTTP          | Протокол прикладного уровня передачи данных  |
| HTTPS         | Расширение протокола НТТР, поддерживающее шифрование   |
| ID            | Идентификационный номер  |
| Postman       | Инструмент для разработки, тестирования и документирования API (Application Programming Interface) |
| DECE          | Архитектурный стиль взаимодействия компонентов   |
| REST          | распределенного приложения в Сети  |
| COAD          | Простой протокол доступа к объектам, используемый для  |
| SOAP          | передачи сообщений в формате XML   |
|               | Инструмент для тестирования веб-сервисов, таких как SOAP   |
| SoapUI        | (Simple Object Access Protocol) и REST (Representational State                                     |
| •             | Transfer), а также других API  |
|               | Стандартный способ указания адреса ресурса в интернете. URL  |
|               | используется для идентификации и доступа к веб-страницам,  |
| URL           | файлам, изображениям, видео и другим ресурсам в сети.  |
|               | Документация приведена на официальном сайте стандарта  |
|               | https://www.w3.org/TR/xml/   |
| UUID          | Стандарт идентификации, позволяющий системам уникально   |
| UUID          | идентифицировать информацию  |
|               | XSL (eXtensible Stylesheet Language) — семейство   |
|               | рекомендаций консорциума W3C, описывающее языки  |
|               | преобразования и визуализации XML-документов.  |
| XSLT          | XSL несёт в себе информацию о том, как будет оформлен  |
|               | документ, где и как должны располагаться данные.   |
|               | Документация фреймворка приведена на официальном портале   |
|               | https://www.w3.org/TR/xslt20/  |
|               | Формат сериализации данных, концептуально близкий к языкам   |
|               | разметки, но ориентированный на удобство ввода-вывода  |
| YAML          | типичных структур данных многих языков программирования.   |
|               | Документация формата приведена на официальном сайте  |
|               | https://yaml.org   |
| Абонент       | Регистрируемая в Платформе система, принимающая участие в  |
|               | информационном взаимодействии  |
| Авторизация   | Процесс проверки учетных данных пользователя для   |
|               | предоставления доступа к системе   |
| Администратор | Роль пользователя с расширенными доступами для работы с  |
| 1             | Платформой   |
| Г. С          | Библиотека - сборник подпрограмм или объектов, используемых  |
| Библиотека    | в Платформе. В Платформе используются библиотеки в   |
| D 5 5         | формате JAR (Java ARchive — архив Java)  |
| Веб-браузер   | Программа для просмотра веб-страниц  |

| Термин                       | Описание   |  |  |  |
|------------------------------|--|--|--|--|
| Веб-интерфейс                | Способ взаимодействия пользователя с приложением, сервисом или системой через веб-браузер  |  |  |  |
| Компонент                    | Јаva-объекты, которые могут быть использованы в маршрутах для выполнения различных задач, таких как обработка данных, вызов методов, выполнение бизнес-логики и взаимодействие с внешними системами. Бины позволяют интегрировать пользовательский код в маршруты Camel, что делает их гибкими и мощными инструментами для создания сложных интеграционных решений |  |  |  |
| Корпоративная шина<br>данных | Связующее программное обеспечение, обеспечивающее централизованный и унифицированный событийно- ориентированный обмен сообщениями между различными информационными системами на принципах сервисориентированной архитектуры  |  |  |  |
| Маршрут                      | Концепция, используемая в интеграционных решениях, таких как Apache Camel, для описания маршрутов передачи данных между различными системами, приложениями или сервисами   |  |  |  |
| Переменная                   | Проименованная, адресованная область памяти Платформы, предназначенная для хранения значения, используемого при исполнении маршрута  |  |  |  |
| Платформа                    | Программное обеспечение «Интеграционная платформа КЗ 2.0» (версия 2.0), разработанное ООО «Ком3»: платформа межсистемного взаимодействия, обеспечивающая интеграцию информационных систем на основе технологий корпоративной сервисной шины  |  |  |  |
| Пользователь                 | Лицо или организация, являющееся пользователем Платформы   |  |  |  |
| Пользователь-                | Физическое и/или юридическое лицо, которая приобретает и   |  |  |  |
| лицензиат                    | эксплуатирует Платформу  |  |  |  |
| Ресурс                       | Механизм, который позволяет работать с файлами как с источниками данных для использования в маршрутах  |  |  |  |

### 2 Состав функций, реализуемых Платформой

- 1. Ведение перечня интеграционных сценариев:
  - реализация сценариев межсистемного взаимодействия в виде набора Camelмаршрутов;
  - настройка точек доступа к веб-сервисам смежных систем;
  - текстовый редактор с возможностью валидации маршрутов;
  - отображение логов межсистемного взаимодействия;
  - возможность старта и остановки интеграционных сценариев;
  - реализация адаптеров к смежным системам;
  - межсистемный обмен с использованием протоколов:
    - o REST;
    - o SOAP;
    - o FTP:
  - поддержка передачи и трансформации различных форматов данных:
    - о текстовый;
    - o XML;
    - o JSON;
    - o YAML.
- 2. Исполнение интеграционных сценариев для типовых API источников и получателей данных:
  - передача данных по настроенным маршрутам;
  - настраиваемое логирование событий интеграционного взаимодействия.
- 3. Регистрация пользователей ведение перечня пользователей для работы с вебинтерфейсом Платформы.
- 4. Регистрация Абонентов ведение перечня систем источников и получателей данных для работы с АРІ Платформы.
- 5. Возможность добавления сторонних библиотек.

### 3 Описание функций

### 3.1 Регистрация Пользователя

### 3.1.1 Создание, изменение и удаление Пользователя

При первичной развёртке дистрибутива вход доступен под пользователем «Администратор». Для пользования основными функциями системы необходимо пройти первичную авторизацию под пользователем «Администратор» и создать нового пользователя. Пользователь нужен для работы с Пользовательским интерфейсом, созданием маршрутов, ресурсов, абонентов, добавления библиотек, переменных и компонентов.

При первичном входе в систему открывается страница входа, где вводятся логин и пароль для пользователя «Администратор».

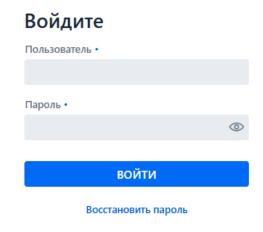


Рисунок 1. Страница входа

Для создания нового пользователя требуется зайти в раздел «Пользователи» и нажать кнопку «Создать».

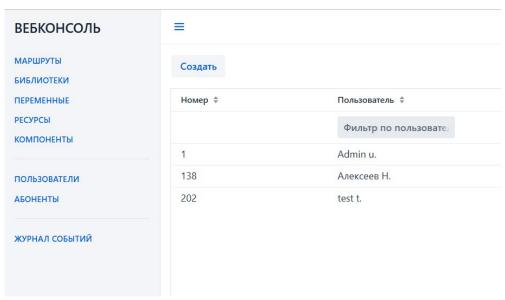


Рисунок 2. Создание Пользователя

Заполнить обязательные поля «Логин», «Фамилия», «Имя», «Еmail» и «Пароль» и нажать кнопку «Сохранить». При создании пользователя система присваивает ему UUID и дату создания. Эти данные хранятся в БД. После этого новый пользователь сможет зайти в веб-интерфейс, используя логин и пароль. Для возвращения в общий список пользователей, нужно нажать кнопку «Назад».

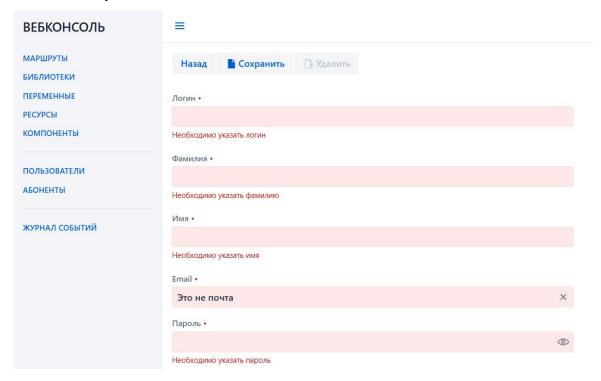


Рисунок 3. Страница создания Пользователя

Изменять данные пользователя или удалить пользователя может либо пользователь с ролью «Администратор», либо сам пользователь. Удаление других пользователей или изменение данных возможно только для пользователя с ролью «Администратор». Для этого требуется нажать кнопку «Изменить» или «Удалить» в карточке пользователя.

| ВЕБКОНСОЛЬ             |                            |  |  |
|------------------------|----------------------------|--|--|
| МАРШРУТЫ<br>БИБЛИОТЕКИ | Назад 📝 Изменить 🕒 Удалить |  |  |
| ПЕРЕМЕННЫЕ             | Логин•                     |  |  |
| РЕСУРСЫ                | test                       |  |  |
| компоненты             | Фамилия •                  |  |  |
| ПОЛЬЗОВАТЕЛИ           | test                       |  |  |
| АБОНЕНТЫ               | • кмИ                      |  |  |
|                        | test                       |  |  |
| журнал событий         | Email •                    |  |  |
|                        | test@test.ru               |  |  |
|                        | Пароль •                   |  |  |
|                        | ••••••                     |  |  |

Рисунок 4. Изменение и удаление Пользователя

### 3.2 Регистрация Абонентов

Для того, чтобы Платформа могла обрабатывать обращения к сервисам Абонентов, необходимо выполнить регистрацию Абонента и настроить данные аутентификации.

# 3.2.1 Создание Абонента и настройка точек доступа к веб-сервисам смежных систем

Настройка Абонента выполняется через веб-интерфейс Платформы в разделе «Абоненты».

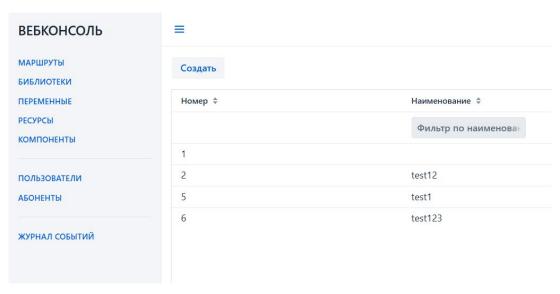


Рисунок 5. Расположение раздела «Абоненты» и кнопки «Создать»

Базовый перечень настроек Абонентов, участвующих в интеграционном взаимодействии, настраивается в веб-интерфейсе.

В открывшемся окне заполняются обязательные поля, для дальнейшей работы Абонента с системой.

Для корректного создания абонента требуется заполнить обязательные поля — «Код», «Наименование», «Пароль» и нажать кнопку «Сохранить». Данные в поле «Код» должны быть уникальными.

Поле «UUID», заполняется автоматически после нажатия кнопки «Создать». Поле «Дата создания», заполняется автоматически после нажатия кнопки «Создать» (Рисунок 6).

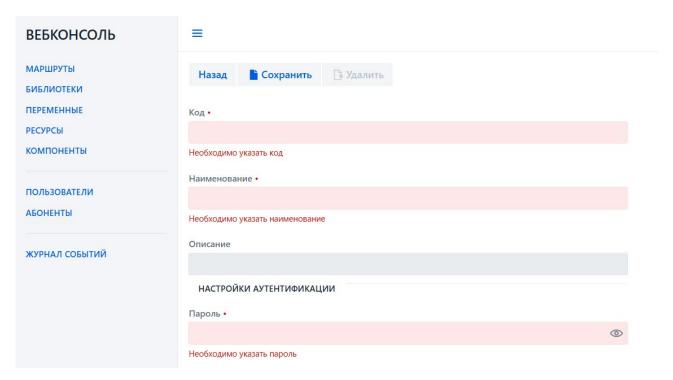


Рисунок 6. Страница создания Абонента

Для удаления и изменения Абонента требуется нажать дважды на созданного Абонента и нажать кнопку «Изменить» для внесения изменений или кнопку «Удалить» для удаления Абонента.

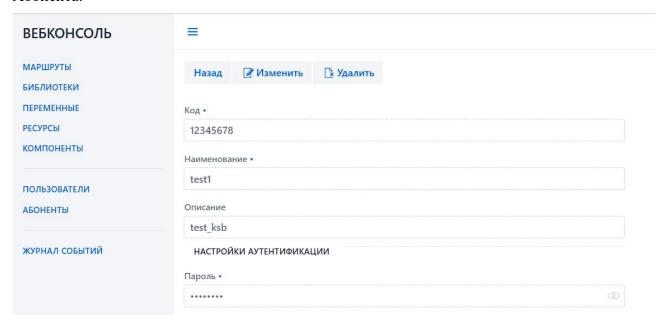


Рисунок 7. Изменение и удаление Абонента

### 3.3 Настройка интеграционных сценариев

### 3.3.1 Настройка и создание маршрута

Маршруты используются для взаимодействия систем по заданным правилам в самом Маршруте.

Настройка и создание маршрута производится в веб-интерфейсе Платформы в разделе «Маршруты» (Рисунок 8).

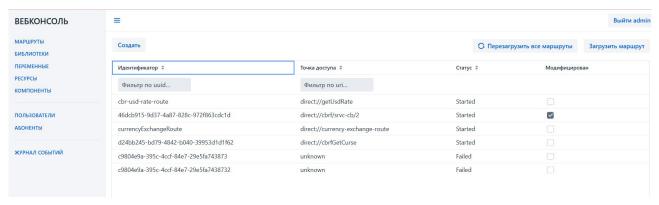


Рисунок 8. Расположение раздела «Маршруты»

Создать маршрут можно нажав кнопку «Создать». После этого в списке появится новый маршрут со статусом «Draft» и с проставленным чек-боксом в столбце «Модифицирован» (Рисунок 9).

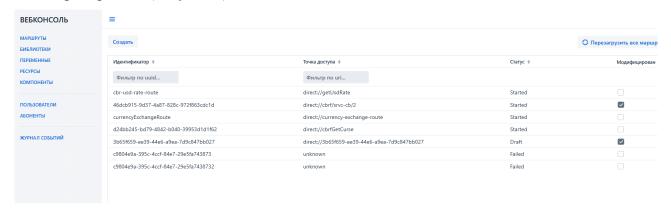


Рисунок 9. Создание нового Маршрута

Для настройки созданного маршрута нужно двойным кликом зайти в него, нажать кнопку «Изменить» и в окне редактирования прописать требуемые параметры и условия для Маршрута. При создании нового Маршрута будет автоматически сгенерирован шаблонный код (Рисунок 10).

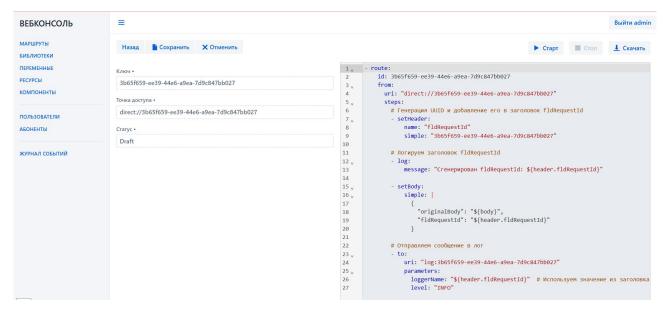


Рисунок 10. Редактирование нового Маршрута

Для добавления готового маршрута в веб-интерфейсе используется кнопка «Загрузить маршрут» (Рисунок 11).

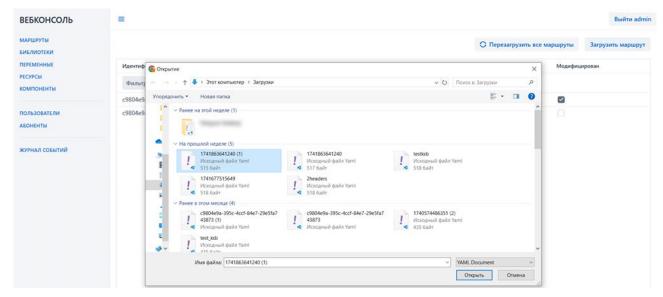


Рисунок 11. Загрузка Маршрута

Чтобы запустить загруженный маршрут требуется двойным кликом зайти в требуемый маршрут и нажать кнопку «Старт», а чтобы отключить маршрут – кнопку «Стоп». Для скачивания маршрута – кнопка «Скачать». Для возвращения к общему списку Маршрутов – кнопка «Назад» (Рисунок 12).

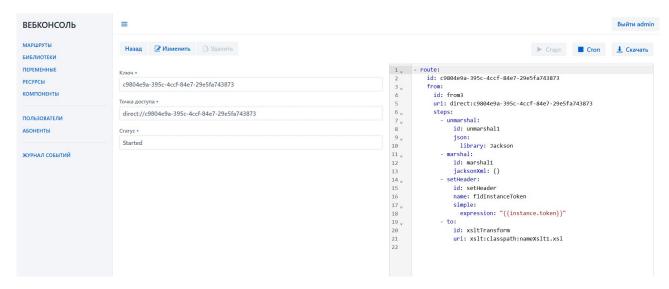


Рисунок 12. Запуск, остановка и скачивание Маршрута

Реализована возможность редактировать загруженный в веб-интерфейс маршрут. В окне редактирования отображаются подсказки по синтаксису и авто дополнения. Комбинация клавиш «ctrl» + «space» отображает возможные варианты объектов для написания в маршруте (Рисунок 13).

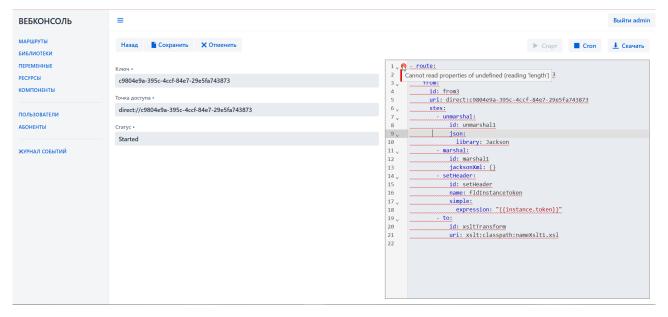


Рисунок 13. Редактирование маршрута в Пользовательском интерфейсе

Маршрут — это центральная концепция, которая определяет, как сообщения будут обрабатываться и перемещаться между различными компонентами системы. Маршрут представляет собой последовательность шагов (этапов), через которые проходит сообщение от точки входа (источника) до точки выхода (назначения). Каждый шаг может выполнять определенные действия, такие как преобразование данных, фильтрация, маршрутизация, логирование и т.д. Создание camel-маршрута происходит по шаблону, описанному ниже:

```
- route:
id: 3b65f659-ee39-44e6-a9ea-7d9c847bb027
```

```
from:
      uri: "direct://3b65f659-ee39-44e6-a9ea-7d9c847bb027"
        # Генерация UUID и добавление его в заголовок fldRequestId
        - setHeader:
            name: "fldRequestId"
            simple: "3b65f659-ee39-44e6-a9ea-7d9c847bb027"
        # Логируем заголовок fldRequestId
            message: "Сгенерирован fldRequestId: ${header.fldRequestId}"
        - setBody:
            simple: |
              {
                "originalBody": "${body}",
                "fldRequestId": "${header.fldRequestId}"
              }
        # Отправляем сообщение в лог
            uri: "log:3b65f659-ee39-44e6-a9ea-7d9c847bb027"
            parameters:
              loggerName: "${header.fldRequestId}" # Используем значение из
заголовка
              level: "INFO"
```

Ниже приведена таблица, содержащая основные элементы при создании маршрута и их описание (Таблица 1).

Таблица 1. Основные элементы, используемые при создании маршрута

| Уровень | Параметр | Описание   | Тип<br>данных           | Обяз. | Родительский<br>параметр |
|---------|----------|--|-------------------------|-------|--------------------------|
| 1       | route    | Служебное поле. Является заголовком для каждого добавляемого маршрута  | Элемент списка, словарь | Да    |                          |
| 2       | id       | Идентификатор маршрута. Должен быть уникальным в рамках всех системы.  | Строка                  | Да    | route                    |
| 2       | from     | Точка входа в маршрут, куда передаются входящие сообщения  | Объект                  | Да    | route                    |
| 3       | uri      | Унифицированный идентификатор ресурса, используемый для вызова маршрута из других маршрутов внутри Системы.                      | Строка                  | Да    | from                     |
| 3       | steps    | Список, определяющий последовательность шагов camel-маршрута, выполняемых для трансформации сообщений и передачи адресату внутри | Список                  | Да    | from                     |

| 4 | setHeader | Платформы или внешним системам, в зависимости от сценария и точки обработки ядром системы. Далее в качестве дочерних элементов описаны некоторые наиболее часто используемы шаблоны (операторы), используемые в качестве шагов Сатеl-маршрута  Устанавливает значение заголовка в сообщении. Используется для добавления или изменения заголовков в сообщении. В данном случае устанавливается заголовок fldRequestId с определенным значением. | Объект                  | Да  | steps     |
|---|-----------|---|-------------------------|-----|-----------|
| 5 | name      | Имя заголовка, который будет установлен. Определяет, какой именно заголовок будет изменен или добавлен.   | Строка                  | Да  | setHeader |
| 5 | simple    | Простое выражение, которое определяет значение заголовка. Указывает значение, которое будет присвоено заголовку.  | Строка или<br>выражение | Да  | setHeader |
| 4 | log       | Шаг, который логирует сообщение. Используется для вывода информации в лог. Это полезно для отладки или отслеживания состояния сообщения в процессе обработки.   | Объект                  | Нет | steps     |
| 5 | message   | Сообщение, которое будет выведено в лог. Определяет, какая информация будет записана в лог. В данном случае выводится значение заголовка fldRequestId.  | Строка или<br>выражение | Нет | log       |
| 4 | to        | Шаг, который отправляет сообщение в указанное место. Используется для передачи сообщения в другой компонент или систему. В данном случае сообщение отправляется в лог.  | Объект                  | Да  | steps     |
| 5 | uri       | Унифицированный идентификатор ресурса (URI), который указывает, куда будет отправлено сообщение. Определяет место назначения сообщения. В данном случае используется log, что означает, что сообщение будет записано в лог.   | Строка                  | Да  | to        |

| 5 | parameters        | Параметры, которые передаются в компонент, указанный в uri. Используется для настройки поведения компонента, в который отправляется сообщение. В данном случае задаются параметры для логгера. | Объект | Нет | to         |
|---|-------------------|--|--------|-----|------------|
| 6 | loggerName        | Имя логгера, которое будет использоваться для записи сообщения. Определяет, под каким именем будет записано сообщение в лог. В данном случае используется значение из заголовка fldRequestId.  | Строка | Да  | parameters |
| 6 | level             | Уровень логирования. Определяет уровень важности сообщения, которое будет записано в лог. В данном случае используется уровень INFO.   | Строка | Да  | parameters |
| 4 | removeHeade<br>rs | Удаляет все заголовки, чтобы очистить сообщение перед отправкой  | Объект | Да  | steps      |
| 4 | toD               | Шаг, который отправляет сообщение в указанное место. Используется для динамической передачи сообщения в другой компонент или систему.  | Объект | Да  | steps      |
| 5 | xpath             | Извлекает значение курса валюты, если есть соответствующая библиотека в КШД.   | Строка | Нет | setBody    |
| 4 | transform         | Оборачиваем извлечённое значение в тег.  | Объект | Да  | steps      |

### 3.3.2 Настройка ресурсов для маршрутов

XSL-преобразования позволяют выполнять более сложные преобразования, выходящие за рамки возможностей функциональности Camel-маршрутов. Такие преобразования добавляются в разделе «Ресурсы».

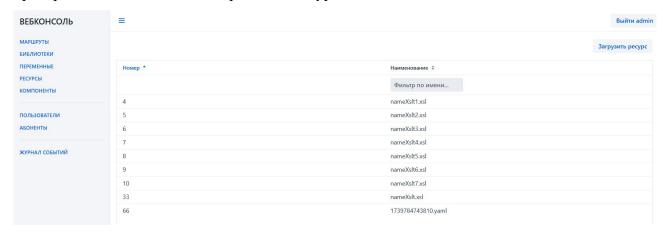


Рисунок 14. Расположение раздела «Ресурсы»

Новый ресурс можно добавить в раздел с помощью кнопки «Загрузить ресурс». Так же в Пользовательском интерфейсе есть возможность его изменить или удалить.

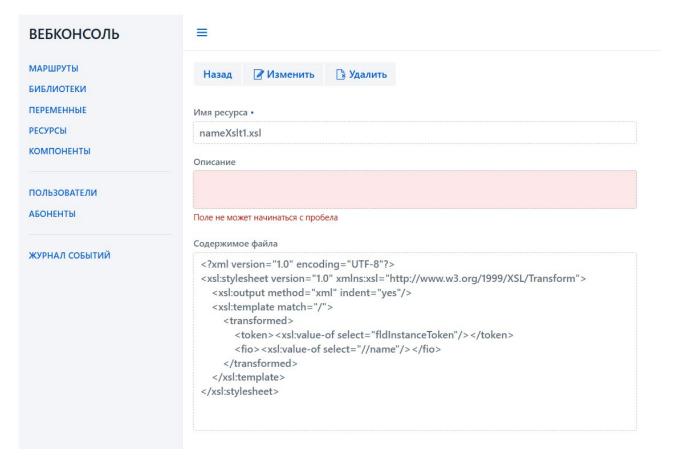


Рисунок 15. Изменение и удаление Ресурса

Вызвать преобразование для маршрута можно прописав его в самом Маршруте, указав название файла с преобразованием, который расположен в разделе «Ресурсы». Пример:

```
- route:
   id: 98187799-54a6-4bc4-bf05-dbe546195d1f
   from:
        uri: direct:98187799-54a6-4bc4-bf05-dbe546195d1f
        steps:
        - unmarshal:
            json:
                library: Jackson
        - marshal:
                jacksonXml: {}
        - to:
                # ниже ссылка на файл с преобразованием
                id: xsltTransform
                 uri: xslt:classpath:filename.xsl
```

Файл преобразования filename.xsl, в данном случае, отвечает за выделение дополнительных Header в ответе сервиса, к которому обращаемся, а именно отображение token из поля fldInstanceToken, которое мы получаем благодаря добавлению Переменной instance.token, из раздела «Переменные» и Header name (ФИО), из базы данных сервиса. Пример содержимого файла преобразования:

Таблица 2. Описание основных элементов, используемых в файле преобразования

| Уровень | Параметр   | Описание  |
|---------|--|---|
| 1       | xml version="1.0" encoding="UTF-8"   | Декларация XML, указывающая, что документ является XML-документом версии 1.0 и использует кодировку UTF-8.  |
| 2       | xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Tr ansform" | Корневой элемент XSLT-документа. Он определяет, что документ является таблицей стилей XSLT версии 1.0. Атрибут xmlns:xsl указывает на пространство имен, используемое для элементов XSLT. |
| 3       | xsl:output method="xml" indent="yes"/  | Элемент определяет, как будет выглядеть выходной документ. В  |

|   |   | данном случае метод xml указывает, что выходной документ будет в формате XML, а атрибут indent="yes" указывает, что выходной XML будет отформатирован с отступами для удобства чтения.   |
|---|---|--|
| 3 | xsl:template match="/"  | Элемент определяет шаблон, который будет применяться к корневому элементу входного XML-документа. Символ / указывает на корень документа.  |
| 4 | transformed   | Пользовательский элемент, который будет создан в выходном документе. Он служит контейнером для других элементов, которые будут добавлены в выходной XML.   |
| 5 | <token><xsl:value-of<br>select="\$token"/&gt;</xsl:value-of<br></token> | Элемент создает элемент <token> в выходном документе. Внутри него используется <xsl:value-of>, чтобы извлечь значение переменной \$token. Переменная \$token должна быть определена где-то в XSLT-коде (в данном коде она не определена, что может привести к ошибке при выполнении).</xsl:value-of></token> |
| 6 | <fio><xsl:value-of select="//name"></xsl:value-of></fio>                | Элемент создает элемент <fio> в выходном документе. Внутри него используется <xsl:value-of>, чтобы извлечь значение элемента <name> из входного XML-документа. Выражение //name ищет все элементы <name> в документе, независимо от их местоположения.</name></name></xsl:value-of></fio>                    |

Полная спецификация, описывающая возможности XSL, доступна на сайте стандарта <a href="https://www.w3.org/TR/xslt20/">https://www.w3.org/TR/xslt20/</a>.

### 3.3.3 Настройка переменных для маршрута

Переменные играют важную роль в управлении данными, логикой и состоянием процессов. Они используются для хранения и передачи информации между различными компонентами системы.

Пример использования переменной был описан выше в разделе 3.3.2.

Для создания и использования переменной требуется нажать кнопку «Создать» и заполнить обязательные поля или загрузить свою Переменную с помощью кнопки «Загрузить настройки».

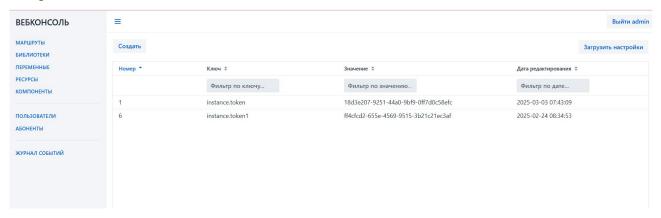


Рисунок 16. Создание и загрузка Переменной

Поле «Ключ» отвечает за идентификацию данных. Он позволяет сервисам и процессам находить нужные переменные в шине данных. Поле обязательно для заполнения и значение в поле должно быть уникальным для корректной работы маршрута. Поле «Значение» отвечает за хранение актуальных данных, которые используются для выполнения бизнес-логики, передачи между сервисами или принятия решений. Поле не может быть пустым. Поле «Комментарий» заполняется при необходимости для удобства использования. Для сохранения Переменной нужно нажать кнопку «Сохранить», для возвращения в общий список — кнопка «Назад». Поля «UUID» и «Дата создания» создаются автоматически при создании Переменной.

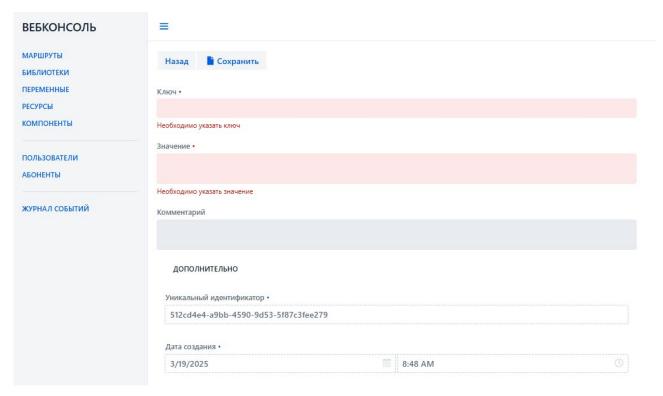


Рисунок 17. Страница создания Переменной

Для внесения изменений или удаления Переменной используются соответствующие кнопки «Изменить» и «Удалить» (Рисунок 18).

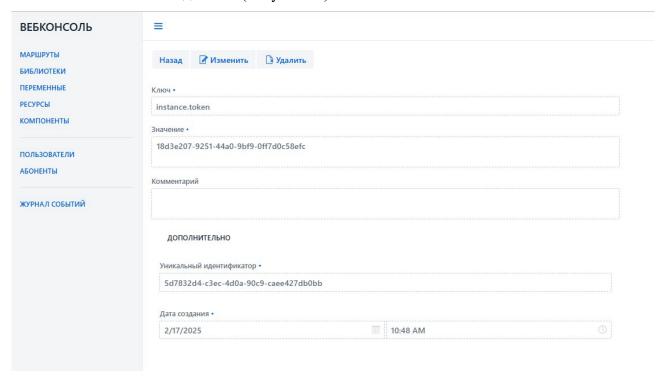


Рисунок 18. Страница редактирования Переменной

### 3.3.4 Настройка библиотек

В Платформе предусмотрена возможность загрузки сторонних библиотек в формате jar для расширения возможностей Платформы.

Для получения консультаций по возможностям данного раздела обратитесь в службу поддержки Производителя.

### 3.3.5 Настройка компонентов

Компоненты, или бины — это Java-объекты, которые можно использовать в маршрутах для выполнения внутренних прикладных технических задач. Компоненты предназначены для обработки данных, выполнения вычислений или взаимодействия с внешними системами. В Camel-маршруте предусмотрена возможность вызова Компонентов для выполнения в ходе обработки маршрутов.

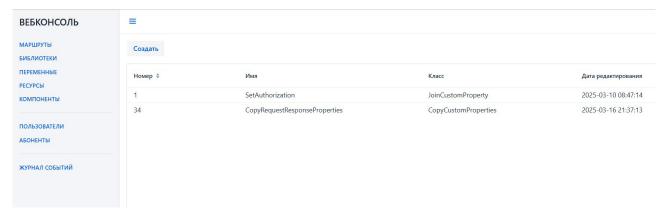


Рисунок 19. Раздел «Компоненты»

Для получения дополнительных консультаций по возможностям данного раздела обратитесь в службу поддержки Производителя.

### 4 Создание тестового сценария для демонстрации функциональности Платформы

Тестовый сценарий демонстрирует обращение пользователя через точку доступа развернутого экземпляра Платформы к АРІ ЦБ РФ как к источнику данных для получения курса валюты на дату, указанную в запросе. Описание сервиса приведено на странице https://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx?op=GetCursOnDate.

Пример демонстрирует возможности Платформы:

- 1) Создание маршрута, включающего выполнение преобразования входящего сообщения к формату, требуемому источником данных, а также преобразования ответного сообщения, полученного от источника данных с целевыми данными к отображению в интерфейсе браузера пользователя, облегчающему восприятие данных пользователем.
- 2) Обращение к точке доступа платформы выполняется через браузер пользователя или при помощи средства эмуляции запросов SoapUI или Postman;

Диаграмма последовательности, демонстрирующая шаги в ходе выполнения тестового сценария, показана на рисунке ниже.

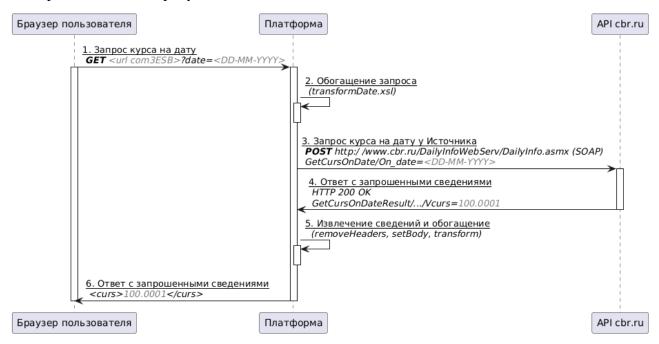


Рисунок 20. Диаграмма последовательности маршрута контрольного примера

Тестовый сценарий требует установки и запуска дистрибутива Платформы на сервере пользователя. Порядок установки Платформы на сервере (виртуальной машине) пользователя приведен в документе Инструкция по установке.

#### 4.1 Создание маршрута

Для выполнения тестового примера требуется создать необходимые для этого файлы и загрузить их в веб-интерфейсе Платформы. Требуется создать следующие файлы: srvc-cbrf.yaml, transformDate.xsl, unpacking.xsl. Названия даны для примера и не являются

обязательными к использованию, но их имена используются при вызове в коде маршрута. Так же необязательно создавать файлы «unpacking.xsl» если тело ответа от сервиса небольшое. В маршруте можно прописать преобразование, что будет рассмотрено в тестовом примере ниже.

1. Создать текстовый файл с названием «transformDate.xsl» с наполнением ниже и загрузить его в разделе «Ресурсы». Данный файл трансформации предназначен для преобразования входящего сообщения к формату, требуемому источником данных.

```
<xsl:stylesheet version="2.0"</pre>
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:web="http://web.cbr.ru/"
                exclude-result-prefixes="">
    <xsl:output method="xml" omit-xml-declaration="yes" indent="yes"</pre>
encoding="UTF-8"/>
    <xsl:strip-space elements="*"/>
    <xsl:param name="dateValue"/>
    <xsl:template match="@* | node()">
        <soapenv:Envelope</pre>
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://web.cbr.ru/">
            <soapenv:Header/>
            <soapenv:Body>
                <web:GetCursOnDate>
                     <web:On date><xsl:value-of</pre>
select="$dateValue"/></web:On date>
                </web:GetCursOnDate>
            </soapenv:Body>
        </soapenv:Envelope>
    </xsl:template>
</xsl:stylesheet>
```

2. В разделе «Маршруты» в веб-интерфейсе Платформы загрузить созданный файл маршрута «srvc-cbrf.yaml» с наполнением ниже (новые объекты, которые не встречались в примере маршрута из пункта 3.3.1 отображены в таблице (Таблица 1. Основные элементы, используемые при создании маршрута), нажав на кнопку «Загрузить маршрут» или создать новый, нажав кнопку «Создать».

```
- route:
    id: d4b44300-d3cd-4543-a80d-ccb5d8516ae4
    from:
        uri: direct:currency-exchange-route
        steps:
        # Устанавливаем заголовок dateValue, используя значение из заголовка
date
    - setHeader: { name: dateValue, simple: "${header.date}" }

# Устанавливаем тело сообщения в виде константы "<item/>"
        - setBody: { constant: "<item/>" }
```

```
# Применяем XSLT-преобразование к телу сообщения, используя файл
transformDate.xsl (возможно использования файла из носителя или ссылаясь на
другой веб ресурс)
        - to: "xslt:classpath:transformDate.xsl"
        # Удаляем все заголовки, чтобы очистить сообщение перед отправкой
        - removeHeaders: { pattern: "*"}
        # Устанавливаем заголовок SOAPAction, необходимый для SOAP-запроса
        - setHeader: { name: SOAPAction, constant:
"http://web.cbr.ru/GetCursOnDate" }
        # Устанавливаем заголовок Content-Type для указания типа содержимого
(XML)
        - setHeader: { name: Content-Type, constant: "text/xml" }
        # Логируем текущее тело сообщения для отладки (не обязательно)
        - log: ${body}
        # Отправляем сообщение на SOAP-сервер с указанием bridgeEndpoint и
отключением исключений при ошибках
        - toD:
"http://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx?bridgeEndpoint=true&throwEx
ceptionOnFailure=false"
        # Удаляем все заголовки, чтобы очистить сообщение перед ответом
клиенту
        - removeHeaders: { pattern: "*" }
        # Извлекаем значение курса валюты с помощью XPath (для USD, код 840)
(можно через файл если ответ большой, а можно через встроенную библиотеку КШД
если ответ небольшой)
        - setBody:
            xpath: "//ValuteCursOnDate[Vcode='840']/Vcurs/text()"
        # Оборачиваем извлечённое значение в тег <curs>
        - transform:
            simple: "<curs>${body}</curs>"
        # Устанавливаем заголовок Content-Type для указания типа содержимого
(XML) (charset=utf8 если в ответе присутствуют русские символы)
        - setHeader: { name: Content-Type, constant: "application/xml" }
```

- 3. Запустить его по нажатию на кнопку «Старт».
- 4. Создать и загрузить файл «unpacking.xsl» с содержимым ниже. Преобразование извлекает из ответного сообщения данные валюты «Доллар США» и обогащает запрос для представления в браузере. В этом шаге необязательно создавать файл, если в ответе менее трёх значений от сервиса. Данный файл может заменить команда в маршруте:

«- setBody: xpath: "//ValuteCursOnDate[Vcode='840']/Vcurs/text()"».

### 4.2 Выполнение тестового примера

Добавленные файлы позволят запросить у сервиса cbr.ru/DailyInfoWebServ/DailyInfo.asmx курс выбранной валюты на требуемую дату.

Порядок действий:

- 1. Открыть SoapUI или Postman (далее описываются шаги про SoapUI).
- 2. Создать новый REST-проект.
- 3. Создать POST-запрос с данными:
  - а. Данные авторизации (кнопка Auth в окне запроса слева снизу): Authorisation: «Basic», Username: «IPKOM3», Password: «1qaz@WSX#EDC»;
  - b. Тело запроса: Media Type: «text/xml», тело запроса дата, на которую запрашивается курс валюты в формате ГГГГ-ММ-ДД, например «2020-01-01» (без кавычек).
- 4. Отправить запрос нажать Alt+Enter.
- 5. Дождаться обработки запроса (до нескольких секунд).
- 6. Открыть тело полученного ответа, для этого переключиться на вкладку XML на правой части экрана. При корректной обработке цепочки маршрутов и преобразований должно отобразиться значение курса валюты за запрошенную дату в формате <curs>61.9057</curs>.

<u>Комментарий</u>. Для того, чтобы контрольный пример можно было выполнять без средства эмуляции запросов SoapUI или Postman, необходимо в настройках основного маршрута закомментировать строку «- setHeader: { name: dateValue, simple: "\${body}"}», поставив в начале строки символ «#». Тогда к сервису можно будет обращаться прямо из адресной строки браузера, а параметр даты нужно будет передавать, приписывая к адресной строке запроса «?dateValue=2020-01-01», где вместо «2020-01-01» указать требуемую дату, а также параметр авторизации в формате «&Authorization=Basic%20dcGFzc3dvcmQ». Тогда ответное сообщение появится в окне браузера при отправке запроса.

### 5 Возможные ошибки

#### 5.1 Пользовательские ошибки

В этом разделе будут разобраны частые ошибки при работе с созданием Абонента, Пользователя, Переменной, Ресурса и созданием маршрута, и работы в Postman.

### 5.1.1 Ошибки при создании Пользователя

Здесь будут разобраны ошибки, связанные с создание, изменением и удалением пользователя.

#### 5.1.1.1 Создание и изменение пользователя

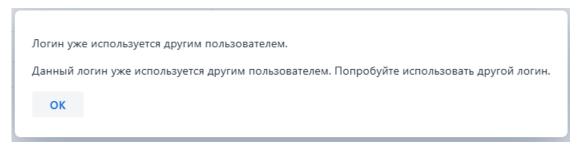
При создании и изменение пользователя могут быть следующие ошибки:

1) Ошибка при использовании пробела в поле «Логин», «Фамилия», «Имя» и «Пароль»:



Чтобы ошибка не возникала – в этих полях не нужно использовать пробел.

2) Ошибка при использовании не уникального значения в поле «Логин» или «Email»:



При возникновении данной ошибки необходимо убедиться, что логин и email введены правильно и ранее не использовались. Чтобы устранить ошибку — нужно использовать логин и email, которые ранее не использовались.

### 5.1.2 Ошибки при создании Абонента

Здесь будут разобраны ошибки, связанные с создание, изменением и удалением абонента.

#### 5.1.2.1 Создание и изменение абонента

При создании и изменение абонента могут быть следующие ошибки:

1) Ошибка при использовании пробела в поле «Код», и «Пароль»:

Код • Нельзя использовать пробел в этом поле

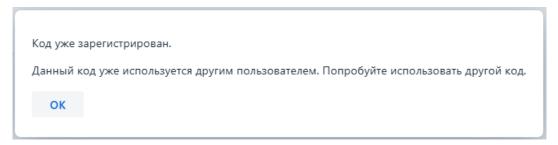
Чтобы ошибка не возникала – в этих полях не нужно использовать пробел.

2) Ошибка при использовании пробела как первого символа в поле «Наименование»:



Чтобы исправить данную ошибку, необходимо удалить пробел как первый символ.

3) Ошибка при использовании не уникального значения в поле «Код»:



При возникновении данной ошибки необходимо проверить уникальность значения в поле. Для устранения ошибки нужно использовать уникальное значение.

### 5.1.3 Ошибки при работе с Маршрутами

Здесь будут разобраны ошибки, связанные с загрузкой маршрута, изменением и удалением маршрута.

### 5.1.3.1 Изменение маршрута

При изменении маршрута через веб-интерфейс или при загрузке изменённого маршрута в веб-интерфейс могут возникнуть следующие ошибки:

1) Ошибка при нарушении синтаксиса во время редактирования маршрута в окне редактирования:

```
1 v
       - route:
    Cannot read properties of undefined (reading 'length') [7
 2
3 <sub>v</sub>
4
            id: from7
 5
            uri: direct:79b7e48e-f46e-4698-92e3-4e1086cda0a7
 6 ,
            step:
            - unmarshal:
7 <sub>v</sub>
8
                   id: unmarshal4
            json:
9 ,
             library: Jackson
10
11 v
            - marshal:
            id: marshal4
12
13
                jacksonXml: {}
```

При возникновении данной ошибки появляется значок уведомления в окне редактирования маршрута, в котором описывается ошибка и её расположение. Для устранения ошибки, требуется правильно указать название объекта или добавить пропущенный символ.

2) Ошибка при нарушении синтаксиса после добавления нового маршрута в Вебконсоль или загрузки обновлённого маршрута:

```
Unsupported field-> stes in 79b7e48e-f46e-4698-92e3-4e1086cda0a7.yaml, line 6, column 9-> - unmarshal-> ^
```

При возникновении данной ошибки следует ознакомиться с её описанием в всплывающем окне и исправить название объекта или добавить пропущенный символ. После этого заново загрузить маршрут.

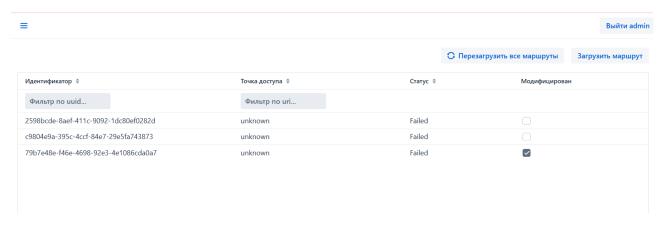
3) Ошибка в случаях, когда неправильно указан путь к переменной или ресурсу:

```
Failed to create route 79b7e48e-f46e-4698-92e3-4e1086cda0a7 at->
>>> To[xslt->classpath->nameXs.xsl] <<< in route-> Route(79b7e48e-
f46e-4698-92e3-4e1086cda0a7)[From[direct->79b7... because of Failed
to resolve endpoint-> xslt->//classpath->nameXs.xsl due to->
javax.xml.transform.TransformerException->
java.io.FileNotFoundException-> Cannot find resource-> classpath-
>nameXs.xsl for URI-> classpath->nameXs.xsl
```

В данном случае требуется ознакомиться с описанием ошибки в всплывающем окне и исправить название или путь к переменной. После этого повторно загрузить маршрут.

### 5.1.3.2 Удаление маршрута

В некоторых случаях, связанных с синхронизацией маршрута с Camel, может возникать такая ошибка:

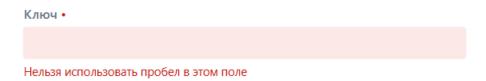


В столбце «Точка доступа» отображается «unknown» и в столбце «Статус» отображается «Failed». В этом случае требуется написать пользователю с ролью «Администратор» запрос на перезагрузку Платформы.

### 5.1.4 Ошибки при работе с Переменными

Здесь будут рассмотрены частые ошибки при работе с переменными – Создание и Изменение.

1) Ошибка при использовании пробела в поле «Ключ»:



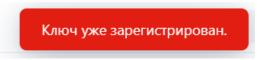
Чтобы ошибка не возникала – в этом поле не нужно использовать пробел.

2) Ошибка при использовании пробела как первого символа в поле «Значение» и «Комментарий»:



Чтобы исправить данную ошибку, требуется удалить пробел как первый символ.

3) Ошибка при использовании не уникального значения в поле «Ключ»:



При возникновении данной ошибки требуется проверить уникальность значения в поле. Для устранения ошибки нужно использовать уникальное значение.

### 5.2 Системные ошибки

Посмотреть ошибки можно в разделе «Журнал событий», скачав файл логов.



Рисунок 21. Скачивание логов в "Журнале событий"

Кроме того, логи доступны для просмотра в веб-интерфейсе Платформы.