

Конфигурация отправки оповещений во внешние источники

В рамках данной лабораторной работы мы расскажем как выполнить конфигурацию отправки оповещений из системы во внешние источники на примере Telegram и Jira. Длительность выполнения лабораторной работы — 1 час.

Отправка оповещений в Telegram

Для того чтобы отправлять оповещения в Telegram, необходимо выполнить три глобальных шага — создать Telegram-бота для приема оповещений, сконфигурировать интеграцию канала в LogIQ и добавить запрос, по которому будут отправляться оповещения. Рассмотрим этот процесс более подробно.

В рамках данной лабораторной работы мы будем выполнять интеграцию с ботом, созданным с помощью приложения Telegram BotFather, но вы можете создать бота через другие источники. Для этого вам понадобится доступ к API и токenu авторизации вашего бота.

Шаг 1. В Telegram откройте приложение BotFather введя в строку поиска имя приложения — **@BotFather**.

Шаг 2. В поле ввода сообщения введите команду **/start**, затем **/newbot**.

Шаг 3. Задайте имя бота, затем задайте его username. Например, **Оповещения из LogIQ** и **logiq_alerts_bot** соответственно.

Шаг 4. Сохраните **Bearer Token**, который вам будет предоставлен для управления ботом через API-запросы.

Шаг 5. Запустите своего бота, перейдя в него и введя команду **/start**.

Шаг 6. С помощью ПО Postman выполните GET-запрос в следующем формате:

GET https://api.telegram.org/bot<bearer-токен_полученный_в_Telegram>/getUpdates

Шаг 7. В полученном ответе найдите идентификатор вашего чата с именем поля **chat_id**.

Шаг 8. В веб-интерфейсе LogIQ перейдите в раздел **Мониторинг → Внешние запросы** и нажмите **+** чтобы **Создать канал** для отправки оповещений.

Шаг 9. Во всплывающем модальном окне укажите название и описание для канала оповещений.

Шаг 10. Укажите адрес для доступа к API вашего бота в виде:

https://api.telegram.org/bot<bearer-токен_полученный_в_Telegram>

Шаг 11. Выберите **No Auth** в качестве метода аутентификации.

Шаг 12. Установите ограничение запросов под ваши требования, например, не больше 60 сообщений в час.

Шаг 13. Нажмите **Создать**, чтобы сохранить новый канал оповещений.

Шаг 14. Нажмите **+** и в выпадающем списке выберите **Создать запрос**.

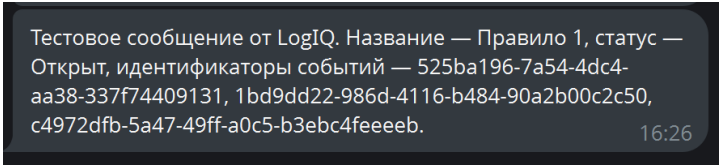
Шаг 15. В открывшемся модальном окне добавьте название и описание для запроса, а также выберите ранее созданный канал оповещений в поле **Канал**.

Шаг 16. Установите таймаут и количество повторений при ошибке запроса.

Шаг 17. В блоке **Настройки сообщения** установите **Тип контента** application/json и задайте тело запроса. В нашем примере мы просто отправим тестовое сообщение. Для этого введите следующий запрос:

```
{
  "chat_id": "ваш_идентификатор",
  "text": "Тестовое сообщение от LogIQ. Название — **name**, статус — **status**, идентификаторы событий — **events.uuid**."
}
```

После выполнения запроса, в ваш Telegram-бот будет отправлено сообщение следующего вида:



Тестовое сообщение от LogIQ. Название — Правило 1, статус —
Открыт, идентификаторы событий — 525ba196-7a54-4dc4-
aa38-337f74409131, 1bd9dd22-986d-4116-b484-90a2b00c2c50,
c4972dfb-5a47-49ff-a0c5-b3ebc4feeeeb.

Вы можете создавать любой запрос под свои нужды. Подробное описание переменных для запросов доступно в руководстве разработчика.

Заведение задач в Jira

Для того чтобы превращать оповещения в новые тикеты в Jira, необходимо выполнить три глобальных шага — получить

токен для подключения к API вашего проекта Jira, сконфигурировать интеграцию канала в LogIQ и добавить запрос, по которому будут отправляться оповещения для создания тикетов. Рассмотрим этот процесс более подробно.

Шаг 1. В Jira перейдите в ваш профиль, затем откройте вкладку **Персональные токены доступа** и нажмите **Создать токен**.

Шаг 2. Введите имя токена и установите окончание срока его действия (при необходимости).

Шаг 3. Сохраните **Bearer Token**, который вам будет предоставлен для отправки API-запросов.

Шаг 4. С помощью ПО Postman выполните GET-запрос в следующем формате:

GET https://<IP-адрес_Jira>/rest/api/2/project

Выберите тип авторизации **Bearer Token** и введите токен, полученный в **шаге 3**.

Шаг 5. В полученном ответе найдите проект, для которого вам необходимо создавать тикеты и скопируйте его идентификатор.

Шаг 6. С помощью ПО Postman выполните GET-запрос в следующем формате:

GET https://<IP-адрес_Jira>/rest/api/2/issue/createmeta/<имя_проекта>/issuetypes

Выберите тип авторизации **Bearer Token** и введите токен, полученный в **шаге 3**.

Шаг 7. В полученном ответе найдите тип задачи, которая будет создаваться и скопируйте его идентификатор.

Шаг 8. В веб-интерфейсе LogIQ перейдите в раздел **Мониторинг → Внешние запросы** и нажмите + чтобы **Создать канал** для отправки оповещений.

Шаг 9. Во всплывающем модальном окне укажите название и описание для канала оповещений.

Шаг 10. Укажите адрес для доступа к API Jira в виде:

https://<IP-адрес_Jira>/rest/api/2/issue

Шаг 11. Выберите **Bearer Token** в качестве метода аутентификации и укажите его значение в поле **Токен**.

Шаг 12. Установите ограничение запросов под ваши требования, например, не больше 60 сообщений в час.

Шаг 13. Нажмите **Создать**, чтобы сохранить новый канал оповещений.

Шаг 14. Нажмите + и в выпадающем списке выберите **Создать запрос**.

Шаг 15. В открывшемся модальном окне добавьте название и описание для запроса, а также выберите ранее созданный канал оповещений в поле **Канал**.

Шаг 16. Установите таймаут и количество повторений при ошибке запроса.

Шаг 17. В блоке **Настройки сообщения** установите **Тип контента** application/json и задайте тело запроса. В нашем примере мы создадим тестовую задачу. Для этого введите следующий запрос:

```
{
  "fields": {
    "summary": "Тест интеграции с LogIQ",
    "description": "Выявлена проблема! **status** — актуальный статус",
    "project": {
      "id": "<идентификатор_проекта>"
    }
  },
  "issuetype": {
    "id": "<идентификатор_типа_задачи>"
  }
}
```

После этого на вашей доске в Jira от вашего имени будет заведена задача с названием **Тест интеграции с LogIQ** и описанием **Выявлена проблема! Открыт — актуальный статус**.

Вы можете создавать любой запрос под свои нужды. Подробное описание переменных для запросов доступно в руководстве разработчика. Также вы можете обогатить свой запрос приоритетом, назначенными исполнителями и так далее. Подробное описание доступных полей описано в [официальной документации Atlassian](https://docs.axel.pro/entry/1363/).