InfoWatch RocketChat Adapter. Руководство по установке, конфигурированию и запуску

13/02/2025

© АО “ИнфоВотч”

Тел./Факс +7 (495) 229-00-22

http://www.infowatch.ru

СОДЕРЖАНИЕ

1 Введение 3

2 Общие сведения и особенности работы InfoWatch RocketChat Adapter 4

2.1 Особенности работы 4

2.1.1 Перехват сообщений 4

2.1.2 Перехват файлов 5

2.1.3 Перехват истории 5

2.1.4 Ограничения 5

3 Программные требования 6

4 Состав Адаптера и схема взаимодействия модулей 7

4.1 Connection Manager 7

4.2 Rocket Cache 8

4.3 Chats Manager 8

4.4 Push-API Sender 8

4.5 File Storage 9

5 Установка InfoWatch RocketChat Adapter 10

5.1 Подготовка на сервере Traffic Monitor 10

5.2 Подготовка на сервере Rocket.Chat 10

5.3 Установка Адаптера 12

6 Настройка Адаптера после установки 14

6.1 Состав конфигурационного файла 14

7 Запуск, остановка и удаление Адаптера 19

7.1 Адаптер, установленный из пакета 19

7.2 Адаптер, развернутый в Docker-контейнере 20

1 Введение

В настоящем руководстве представлены сведения по установке, настройке и запуску Infowatch RocketChat Adapter.

В настоящем руководстве **не** рассматриваются вопросы установки и настройки сервера Rocket.Chat. За получением необходимых сведений вы можете обратиться к официальной документации Rocket.Chat (доступна на английском языке).

Аудитория

Руководство предназначено для администраторов InfoWatch Traffic Monitor, которым требуется перехватывать тип события "Рокетчат" в сервисе "Мессенджер".

Руководство рассчитано на пользователей, имеющих навыки администрирования в среде операционных систем семейств MS Windows и Linux.

2 Общие сведения и особенности работы InfoWatch RocketChat Adapter

InfoWatch RocketChat Adapter (далее - Адаптер) - программный модуль, предназначенный для контроля внутренних коммуникаций сотрудников компаний, которые используют мессенджер RocketChat.

Адаптер позволяет извлекать из мессенджера RocketChat новые текстовые сообщения и вложения из групповых и личных чатов, чтобы передавать их в Traffic Monitor. Адаптер не имеет собственного графического интерфейса пользователя.

2.1 Особенности работы

Адаптер использует учетную запись, созданную на сервере мессенджера Rocket.Chat. Для перехвата сообщений Адаптер добавит эту учетную запись во все закрытые группы.

Для взаимодействия с Адаптером на сервер Traffic Monitor должен быть добавлен специальный плагин. Добавление плагина в Traffic Monitor регистрирует:

- новый тип события "Рокетчат" в сервисе "Мессенджер" (код: `im_rocket`);
- новый тип контакта "Контакт Рокетчат" (код: `im_rocketId`).

Для отправителя и получателя Адаптер записывает все контакты, извлеченные при перехвате:

- тип контакта e-mail;

  > **Примечание:**
  >
  > Если e-mail отсутствует, будет добавлен Контакт Рокетчат.

- отображаемое имя пользователя: ФИО, псевдоним (nickname);
- логин в Rocket.Chat.

RocketChat Adapter запрашивает данные по:

- Realtime API - для открытых и закрытых комнат/групп, на которые подписана учетная запись. Адаптер получает данные в реальном времени;
- REST API - для личных переписок. Адаптер запрашивает данные с заданной периодичностью.

2.1.1 Перехват сообщений

Адаптер перехватывает сообщения в личных и групповых чатах, в открытых и закрытых группах.

Если при перехвате фрагмента чата будет выполнено одно из условий формирования событий, Адаптер создаст событие с теневой копией фрагмента чата.
Таким образом, при перехвате один чат может быть разделен на несколько теневых копий, и на каждую теневую копию должно быть сформировано отдельное событие.

Пересылка сообщений в другие чаты является отправкой новых сообщений.

При перехвате цитат событие будет содержать только текст, добавленный отправителем.

Отредактированное сообщение Адаптер перехватывает как новое.

Адаптер игнорирует удаление в чате отправленных ранее и перехваченных им сообщение.

2.1.2 Перехват файлов

При перехвате файлов в качестве получателей будут указаны все участники чата.

Если к отправке файла отправитель добавляет текст (описание), Адаптер при перехвате сформирует два события:

- событие с файлом;
- событие чата с текстом.

У обоих событий будет одно время формирования. Событие чата обрабатывается согласно настройке формирования событий чата. То есть в событии с текстом, добавленным при отправке файла, могут быть другие сообщения чата.

> **Важно!**
>
> При пересылке файла из одного чата в другой Адаптер не формирует событие.

2.1.3 Перехват истории

Адаптер не перехватывает всю историю переписки до момента его установки. При запуске Адаптер запрашивает из чатов последние сообщения. Количество запрашиваемых сообщений определяет параметр `initcount` в секции `rooms` конфигурационного файла `rocketconfig.json`. Из каналов Адаптер не запрашивает прошлые сообщения, только из p2p-чатов (Peer-to-Peer).

2.1.4 Ограничения

Адаптер не поддерживает:

- перехват сообщений и файлов в следующих типах комнат Rocket.Chat:
  - Teams;
  - Discussions;
  - Threads;
  - Omnichannels;
  - Livechannels.
- перехват аудио- и видеозвонков.

Информация о каналах/группах и составе их участников обновляется в Адаптере по таймауту, указанному в конфигурационном файле.
Если в процессе работы Адаптера был добавлен новый канал, сообщения не будут перехвачены до тех пор, пока адаптер не получит данные о новом канале. Перехвачены будут только те сообщения, которые будут отправлены после того, как учетная запись Адаптера подпишется на новый канал.
Состав участников на момент формирования событий может быть неактуальным, если Адаптер еще не обновил информацию.

3 Программные требования

Адаптер поддерживает установку на ОС семейства Linux:

- Red Hat Enterprise Linux 7.x;
- РЕД ОС 7.3.x;
- Ubuntu 22.x.

Вы можете установить Адаптер, используя поставляемый пакет, или развернуть его в Docker-контейнере.

Развернуть в Docker-контейнере вы можете на любой ОС, которая поддерживает установку контейнера с Ubuntu.

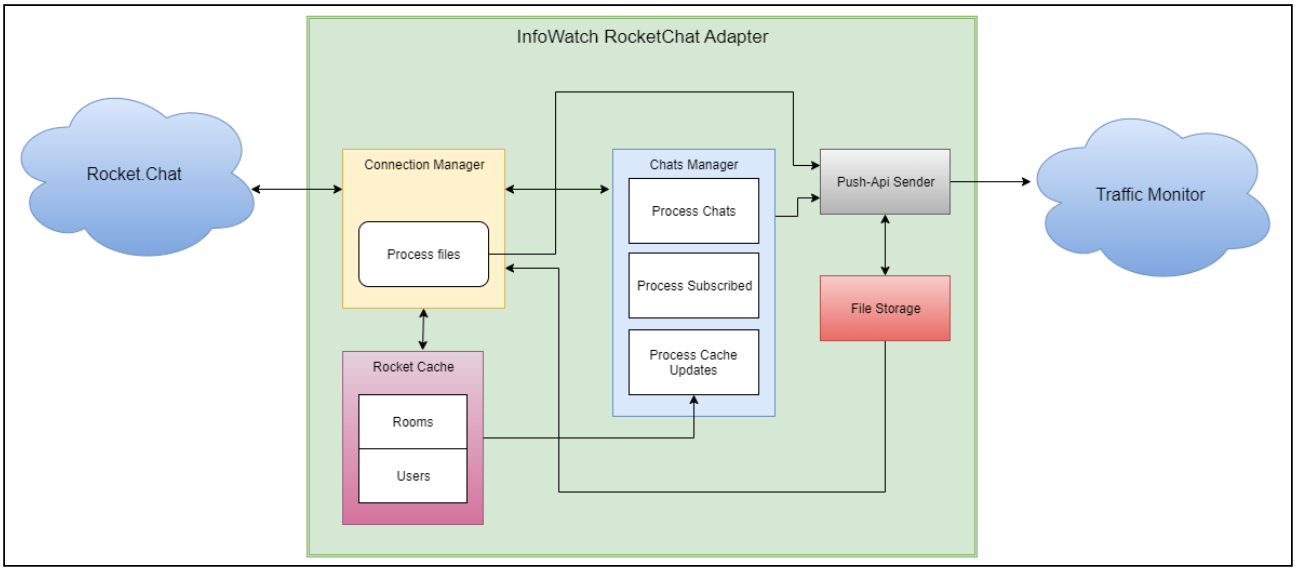
> **Примечание:**
>
> Для развертывания в Docker-контейнере обратитесь к сотрудникам компании InfoWatch.
>
> Для этого способа установки на сервере должен быть установлен Docker 20.10.23 или более новый.

4 Состав Адаптера и схема взаимодействия модулей

Адаптер InfoWatch RocketChat Adapter реализован в виде нескольких основных взаимодействующих модулей:

- Connection Manager;
- Rocket Cache;
- Chats Manager;
- Push-API Sender;
- File Storage.

Модули Адаптера взаимодействуют по следующей схеме:



4.1 Connection Manager

Компонент отвечает за взаимодействие с сервером Rocket.Chat:

- реализует весь функционал доступа по Realtime API и по REST API;
- содержит данные учетной записи для соединения с сервером Rocket.Chat;
- выполняет открытие соединений, вход пользователя и запросы любых данных от Rocket.Chat.

Все компоненты, запрашивающие данные у сервера Rocket.Chat, работают через Connection Manager.

Для обработки сообщений, содержащих файлы, Connection Manager имеет один отдельный поток выполнения `ProcessMessagesWithFiles`.

Если сообщение от групп содержит файлы, Адаптер формирует два события:

1. генерирует объект-файл для Push-API sender;
2. формирует текстовое сообщение, содержащее название файла и его описание, и отправляет его в модуль Chats manager.

Сообщения с файлами из личных переписок (Direct Messages) передаются в модуль Chats Manager. После обработки файлы из этих сообщений отправляются в поток `ProcessMessagesWithFiles`.

4.2 Rocket Cache

Модуль хранит кэш данных из Rocket.Chat.

Хранит и с заданной периодичностью обновляет:

- список всех пользователей Rocket.Chat с контактной информацией;
- список всех комнат (Direct Messages, Channels, Private Groups) и их участников;
- список всех закрытых групп (Private Groups), на которые подписана служебная учетная запись Rocket.Chat.

Модуль имеет отдельный поток выполнения `ProcessUpdates`, в котором по таймауту, указанному в конфигурационном файле, производит обновление списков пользователей, комнат и их участников.

4.3 Chats Manager

Из отдельных сообщений Rocket.Chat модуль формирует чаты по разным комнатам.

Chats manager работает в три потока:

- `ProcessSubscribed` - обработка сообщений Rocket.Chat, полученных от Connection Manager по Realtime API. Когда количество сообщений в чате достигает значения, заданного в конфигурационном файле, отправляет собранные данные в Push-API sender;
- `ProcessCacheUpdates` - обработка обновлений комнат от Rocket Cache (добавление, удаление);
- `ProcessChats` - основной обработчик. По таймауту, заданному в конфигурационном файле, отправляет собранные чаты для всех типов комнат. Отправляет сообщения с файлами из личных переписок (Direct Messages) в поток `ProcessMessagesWithFiles` (Connection Manager).

4.4 Push-API Sender

Модуль отправляет данные в Traffic Monitor.
Push-API Sender состоит из:

- `ProcessMessages` - основной обработчик. Получает от Chats Manager собранные чаты, формирует из них события в формате push-api. События типа "Файл" всегда сразу отправляются в очередь обработки модуля File Storage. Текстовые события передаются в очередь на отправку в Traffic Monitor. Основной обработчик запускает второстепенные;
- `ProcessConnection` - второстепенные обработчики. Их количество задает параметра "`count`" секции "`tmserver`" конфигурационного файла адаптера.
  `ProcessConnection` вычитывают события из очереди на отправку и отправляют их в Traffic Monitor.
  Когда событие, сохраненное в File Storage, успешно отправлено в Traffic Monitor, обработчик отправляет это событие на удаление в очередь обработки File Storage. Каждый обработчик независимо от других отправляет свой статус в File Storage через выделенный канал.
  Также `ProcessConnection` управляют соединением с Traffic Monitor и в случае проблем повторно

подключаются по возрастающему таймауту: 1 сек, 5 сек, 20 сек, 2 мин, 10 мин, 30 мин. Если восстановить соединение не удалось, обработчик будет повторять попытку соединения раз в 30 минут.

4.5 File Storage

Этот модуль - хранилище данных. File Storage сохраняет в файловой системе файлы и сформированные события push-api в формате JSON.

Отдельное событие представлено папкой с `event.json`. Если событие содержит файлы, они также будут в папке события. Имя папки события - уникальный id. Путь для хранения указан в конфигурационном файле Адаптера.

File Storage создает в хранилище две папки:

- `Out` - хранилище;
- `Error` - события, при отправке которых возникли ошибки, связанные с самим событием (например, неверный формат). В папках событий сохраняется файл `error.txt` с описанием ошибки.
  События с ошибками можно обработать только вручную. После исправления ошибки переместите папку с событием в папку `Out` и перезапустите Адаптер.

File Storage состоит из:

- `ProcessEvents` - основной обработчик. Он следит за статусами обработчиков Push-API Sender. При изменении статуса проверяет, запущен ли загрузчик данных. Если соединение потеряно, останавливает загрузчик. Если соединение восстановлено, запускает его. Также при изменении статуса `ProcessEvents` уведомляет второстепенные обработчики.
- `loadToStorage` - загрузчик данных. Читает каталог хранилища Out, для каждого события создает объект на загрузку и помещает его в очередь-кэш. Размер очереди-кэша задает параметр "`cachesize`" секции "`filestorage`" конфигурационного файла Адаптера.
- `ProcessWork` - второстепенные обработчики. Вычитывают события из двух очередей: от Push-API Sender и из кэша загрузки. Push-API Sender и загрузчик выставляют операции обработчикам `ProcessWork`. В зависимости от выставленных операций, обработчики выполняют действия с событиями.
  Количество второстепенных обработчиков `ProcessWork` определяет параметр "`count`" секции "`filestorage`" конфигурационного файла Адаптера.

Типы операций:

- `SaveEvent` - сохранение события в хранилище. При установленном соединении с Traffic Monittor событие помещается в очередь на отправку Push-API Sender.
- `AddFiles` - обработчик создает папку для события в хранилище и загружает туда файлы из Rocket.Chat через Connection Manager. После событие будет передано обратно в очередь на отправку Push-API Sender.
- `DelEvent` - удаление события из хранилища.
- `LoadEvent` - при установленном соединении с Traffic Monitor событие загружается из хранилища и передается в очередь на отправку Push-API sender.

Если при отправке событие вызвало ошибку, оно будет перемещено в папку `Error`.

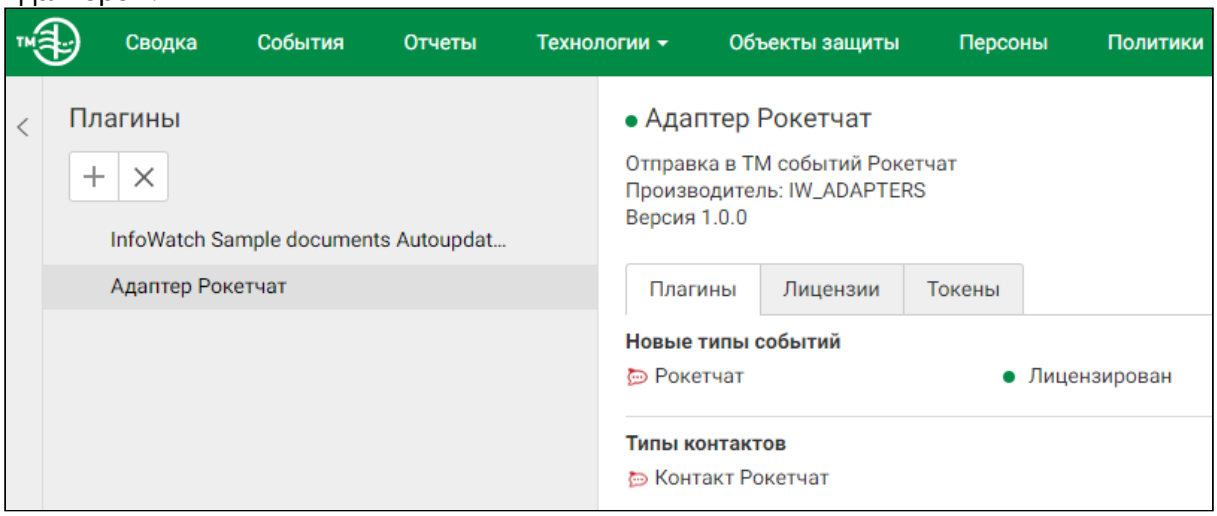
5 Установка InfoWatch RocketChat Adapter

Чтобы развернуть InfoWatch RocketChat Adapter, выполните следующие шаги:

1. Подготовка на сервере Traffic Monitor.
2. Подготовка на сервере Rocket.Chat.
3. Установка Адаптера.

5.1 Подготовка на сервере Traffic Monitor

1. В консоли управления Traffic Monitor перейдите в раздел Лицензии и добавьте активную лицензию на использование Адаптера.
2. Перейдите в раздел Плагины и добавьте специальный плагин для взаимодействия с Адаптером.



3. Выберите добавленный плагин Адаптер Рокетчат и перейдите на вкладку Токены.
4. Выберите активный токен, скопируйте и сохраните это значение.
   Оно будет использовано позже для настройки подключения Адаптера к серверу Traffic Monitor.

5.2 Подготовка на сервере Rocket.Chat

1. Создайте в Rocket.Chat нового пользователя или используйте существующего.

> **Примечание:**
>
> При выборе имени учитывайте, что оно будет отображаться в служебных сообщениях при добавлении пользователя в закрытые группы.

2. Для возможности перехвата чатов, выдайте пользователю права доступа.
   Рекомендуем выдать пользователю права администратора.

> **Примечание:**

Для перехвата всех поддерживаемых Адаптером чатов пользователю должны быть выданы следующие права доступа:

- Просматривать административную информацию (view-room-administration). Обязательное разрешение;
- Добавить пользователя к любому приватному каналу (add-user-to-any-p-room);
- Добавить пользователя к любому публичному каналу (add-user-to-any-c-room);
- Присоединяться к публичным чатам без кода входа (join-without-join-code);
- Смотреть публичные каналы (view-c-room);
- Просматривать личные сообщения (view-d-room);
- Просмотр полной информации о других пользователях (view-full-other-user-info);
- Просматривать историю (view-history);
- Просматривать LiveChat комнаты (view-livechat-rooms);
- Просматривать закрыты чаты (view-p-room);
- Просмотр привилегированных настроек (view-privileged-setting);
- Просматривать статистику (view-statistics);
- Снять ограничения по количеству запросов к REST API (api-bypass-rate-limit).

Если отсутствует возможность выдать полные права доступа, Адаптер может перехватывать сообщения только из открытых каналов. Для этого потребуются следующие права доступа:

- Создание токенов персонального доступа (create-personal-access-tokens);
- Смотреть публичные каналы (view-c-room);
- Просмотр полной информации о других пользователях (view-full-other-user-info);
- Просматривать историю (view-history).

Роли и права можно настроить в разделе *Администрирование → Права доступа* (Administration → Permissions).
Для поиска прав доступа вводите их названия на английском языке.

Для перехвата файлов из p2p-чатов перейдите в *Администрирование → Настройки → Загрузка файлов* (*Administration → Settings → File Upload*) и отключите следующие ограничения:

- Restrict files to rooms' members
- Restrict files to users who can access the room

3. Перейдите в раздел *Моя учетная запись → Токены для личного доступа* (*My Account → Personal Access Token*).
4. Добавьте пользователю новый токен.
   На экране отобразятся токен и ID пользователя. **Обязательно** сохраните их, они будут использованы позже для настройки подключения Адаптера к серверу Rocket.Chat.

> **Важно!**
>
> Токен и ID пользователя будут показаны только один раз при добавлении. Просмотреть их позже будет невозможно.

5. Перейдите в раздел *Администрирование → Настройки → Общие настройки* (*Administration → Settings → General*) и разверните пункт REST API.
6. Обязательно активируйте опции:
    - *Включить конечную точку истории личных сообщений* (*Enable Direct Message History Endpoint)*;
    - *Разрешить получить все (Allow getting everything)*.

5.3 Установка Адаптера

Чтобы установить Адаптер из пакета:

1. Скопируйте на сервер поставляемый файл:
    - `iwtm-rocketchat-7.11.1.300-release.x86_64.rpm` - пакет для установки Адаптера на Red Hat Enterprise Linux и РЕД ОС;
    - `iwtm-rocketchat_7.11.1.300-release_amd64.deb` - пакет для установки Адаптера на Ubuntu.
2. Выполните **одну** из команд установки:
    - Для Red Hat Enterprise Linux и РЕД ОС:
        - `yum install iwtm-rocketchat-7.11.1.300-release.x86_64.rpm`
        - `dnf install iwtm-rocketchat-7.11.1.300-release.x86_64.rpm`
        - `rpm -ivh iwtm-rocketchat-7.11.1.300-release.x86_64.rpm`
    - Для Ubuntu:
        - `apt-get install iwtm-rocketchat_7.11.1.300-release_amd64.deb`
        - `dpkg -i iwtm-rocketchat_7.11.1.300-release_amd64.deb`

Чтобы развернуть Адаптер в Docker-контейнере:

1. Скопируйте на сервер в одну директорию поставляемые файлы:
    - `Dockerfile.ubuntu22_04` - Dockerfile, набор инструкций для создания образа, из которого будет собираться контейнер. Файл также содержит комментарии разработчика;
    - `iwtm-rocketchat_7.11.1.300-release_amd64.deb` - пакет для установки Адаптера в операционных системах, основанных на Debian;
    - `iwtm-rocketchat-7.11.1.300-release.x86_64.rpm` - пакет для установки Адаптера в операционных системах, основанных на Red Hat.
2. Перейдите в директорию, в которую скопировали файлы.
3. Откройте на редактирование `Dockerfile.ubuntu22_04` .
4. Если для установки Адаптера вы будете использовать:
    - **ваш репозиторий**, содержащий необходимый пакет - в строке, начинающейся с `ADD` , замените указанный адрес на адрес вашего репозитория.
    - **поставляемый пакет установки**:
        a. Закомментируйте строку, начинающуюся с `ADD` .

        > **Примечание:**
        >
        > Чтобы закомментировать строку, добавьте в начале символ `#` . Чтобы раскомментировать строку, удалите в начале символ `#` .

        b. Раскомментируйте строку вида:
        `COPY iwtm-rocketchat_7.11.1.300-release_amd64.deb /sandbox` .

c. В строке из предыдущего шага укажите имя пакета, который вы будете использовать.

5. Для сборки образа контейнера с Адаптером выполните команду:
   ```
   docker buildx build -f Dockerfile.ubuntu22_04 . -t iw_rocketchat
   ```

После установки InfoWatch RocketChat Adapter обязательно выполните первичную настройку.

6 Настройка Адаптера после установки

Перед запуском InfoWatch RocketChat Adapter выполните настройку:

1. Если Адаптер развернут в Docker-контейнере:
   a. Создайте в какой-либо директории папки:
      - `etc` - для конфигурационного файла Адаптера;
      - `log` - для лог-файлов Адаптера;
      - `fstorage` - для файлового хранилища Адаптера.

      > **Примечание:**
      >
      > Пользователь, от имени которого будет запускаться Адаптер, должен обладать правами на чтение и запись в созданных папках.

   b. Скопируйте прилагаемый конфигурационный файл `rocketconfig.json` в созданную директорию `etc`.
2. Отредактируйте конфигурационный файл `rocketconfig.json:`

   > **Примечание:**
   >
   > Если Адаптер установлен из пакета, конфигурационный файл - `/opt/iw/tm5/etc/rocketconfig.json`.

   a. В секции `"rocketserver"` укажите в параметрах:
      i. `"host"` - IP-адрес или доменное имя сервера Rocket.Chat.
      ii. `"port"` - порт для подключения.
   b. В секции `"rocketuser"` укажите в параметрах:
      i. `"id"` - ID пользователя Rocket.Chat.
      ii. `"token"` - токен пользователя Rocket.Chat.
   c. В секции `"tmserver"` укажите в параметрах:
      i. `"host"` - IP-адрес или доменное имя сервера Traffic Monitor.
      ii. `"port"` - порт для подключения.
      iii. `"token"` - токен плагина Адаптер Рокетчат.
3. Сохраните изменения в конфигурационном файле.

> **Примечание:**
>
> Остальные параметры имеют значения по умолчанию. При необходимости вы можете их отредактировать.
> Состав конфигурационного файла с описанием параметров представлен в таблице ниже.

6.1 Состав конфигурационного файла

В таблице для параметров указаны значения по умолчанию или примерные.

Содержимое	Значение по умолчанию	Описание
`{ "logger": {`	**Настройки логирования**	
`"level": "trace",`	`Warning`	Уровень логирования. Возможные значения: • *trace* - наиболее подробное логирование (включаются все сообщения с уровнями *debug, info* и выше) • *debug* - вывод отладочной информации, обычно для диагностики ошибок и общения с разработкой/поддержкой • *info* - информационные сообщения, могут быть отключены на стабильной системе • *warning* - предупреждения. • *error* - ошибки. • *fatal* - наиболее важные сообщения, приводящие к завершению работы или отказу.
`"logfile": "/opt/iw/tm5/log/rocketchat.log" },`	`/opt/iw/tm5/log/rocketchat.log`	Файл, в который будут записаны логи работы адаптера внутри контейнера. Содержимое зависит от уровня логирования. Если Адаптер развернут в контейнере: • лог-файл, доступный для взаимодействия, будет создан в директории указанной при запуске Адаптера; • не редактируйте этот параметр.
`"rocketserver": {`	**Настройки подключения к серверу Rocket.Chat**	
`"protocol": "http",`	`http`	Протокол, по которому Адаптер взаимодействует с сервером Rocket.Chat. Возможны значения: • `http` • `https`
`"host": "10.60.44.69",`		IP-адрес или доменное имя сервера, на котором установлен Rocket.Chat
`"port": "3000" },`	`3000`	Порт для подключения к серверу Rocket.Chat. Если вы оставите значение параметра `"port"` пустым:

Содержимое	Значение по умолчанию	Описание
		• по протоколу `http` Адаптер будет подключаться по портам `80` и `8080`; • по протоколу `https` Адаптер будет подключаться по порту `443`.
`"rocketuser": {`	**Учетная запись для подключения к серверу Rocket.Chat**	
`"id": "w89HSvsCs9whxTfWA",`		ID пользователя Rocket.Chat, от имени которого Адаптер будет получать данные
`"token": "Sr-ID83GbeQChzNNg-cC_FoEu9SkI4p0CnBFGXZLEG0",`		Токен пользователя Rocket.Chat, от имени которого адаптер будет получать данные
`"name": "fr1end" },`		Имя пользователя Rocket.Chat, от имени которого адаптер будет получать данные. Необязательный параметр
`"tmserver": {`	**Настройки подключения к серверу Traffic Monitor**	
`"host": "10.60.25.73",`	`localhost`	IP-адрес или доменное имя сервера Traffic Monitor
`"port": "9101",`	`9101`	Порт для подключения к серверу Traffic Monitor. Порт, по которому работает сервис iw_xapy_puppy
`"token": "o3had01xw6tysugktatw",`		Токен доступа из плагина Адаптер Рокетчат
`"company": "IW_ADAPTERS",`	`IW_ADAPTERS`	Идентификатор компании-разработчика адаптера. Должен соответствовать значению в плагине. **Внимание!** Не редактируйте этот параметр
`"count": 4 },`	`4`	Количество обработчиков `ProcessConnection`, которые будут созданы в модуле Push-API Sender для отправки данных в Traffic Monitor. Каждый обработчик создает отдельное соединение с сервером Traffic Monitor. Укажите целое число, больше нуля

Содержимое	Значение по умолчанию	Описание
`"cache": {`	**Настройка работы адаптера**	
`"timeout": 10 },`	`30`	Таймаут обновления данных. При первом подключении Адаптер получает данные обо всех текущих пользователях и каналах. Затем Адаптер обновляет данные с заданной периодичностью. В минутах
`"rooms": {`	**Настройки формирования событий**	
`"initcount": 100`	`100`	Количество последних сообщений, которые запросит Адаптер из каждого p2p-чата при старте. Адаптер запросит последние сообщения только из p2p-чатов, из каналов будут получены только новые сообщения. Укажите целое число, больше нуля
`"timeout": 10,`	`10`	Частота формирования событий чата по времени. Время, по достижении которого Адаптер формирует событие для отправки в Traffic Monitor. В минутах
`"count": 30 },`	`20`	Частота формирования событий чата по количеству сообщений. Количество сообщений, по достижении которого Адаптер формирует событие для отправки в Traffic Monitor. Укажите целое число, больше нуля. **Примечание:** Параметр влияет на отправку только каналов и групп. Адаптер отправляет личные переписки (Direct Messages) только по времени (параметр `"timeout"`)
`"filestorage": {`	**Настройки файлового хранилища адаптера**	
`"path": "/opt/iw/tm5/fstorage",`	`/opt/iw/tm5/fstorage`	Путь к хранилищу. Если Адаптер развернут в контейнере: • хранилище, доступное для взаимодействия, будет создано в директории указанной при запуске Адаптера; • не редактируйте этот параметр.

Содержимое	Значение по умолчанию	Описание
`"count": 4,`	`4`	Количество обработчиков `ProcessWork`, которые будут созданы в модуле File Storage для обработки объектов хранилища. Укажите целое число, больше нуля
`"cachesize": 1100 } }`	`1000`	Размер кэша. Количество объектов из хранилища, которые одновременно могут быть размещены в памяти адаптера. Укажите целое число, больше нуля

7 Запуск, остановка и удаление Адаптера

Перед запуском Адаптера обязательно выполните настройку.

Вы можете запустить Адаптер в одном из режимов:

- **Интерактивном** - команда будет выполняться, пока Адаптер работает.
  Терминал будет занят, а при отключении от него Адаптер прекратит работу.
- **Фоновом** - команда будет выполнена и не будет своим состоянием отображать работу Адаптера.
  Терминал будет свободен, а при отключении от него Адаптер продолжит работу.

В зависимости от типа установки, из пакета или в контейнере, будут отличаться команды для работы с Адаптером

7.1 Адаптер, установленный из пакета

Запуска Адаптера выполните команду:

- `/opt/iw/tm5/bin/rocketchat` - в интерактивном режиме;
- `/opt/iw/tm5/bin/rocketchat &` - в фоновом режиме.

Для остановки Адаптера, запущенного в фоновом режиме:

1. Определите идентификатор процесса (PID) Адаптера с помощью команды:
   `ps -ax | grep rocket`

   > **Примечание:**
   >
   > В выводе команды найдите строку, в которой указан процесс `rocketchat`, слева в начале строки будет указан его идентификатор.
   >
   > **Пример вывода команды**
   >
   > ```
   > 19862 pts/0    Sl     0:00 ./rocketchat
   > 19885 pts/0    S+     0:00 grep --color=auto rocket
   > ```
   >
   > В нашем примере PID: `19862`.

2. Прервите выполнение процесса командой:
   `kill -SIGINT <PID>`
   В нашем примере:
   `kill -SIGINT 19862`

Для остановки Адаптера, запущенного в интерактивном режиме, прервите выполнение процесса комбинацией клавиш CTRL + C.

Для удаления Адаптера выполните одну из команд:

- Дл Red Hat Enterprise Linux и РЕД ОС:
  - `yum remove iwtm-rocketchat-7.11.1.300-release.x86_64.rpm`
  - `dnf remove iwtm-rocketchat-7.11.1.300-release.x86_64.rpm`
  - `rpm -e iwtm-rocketchat-7.11.1.300-release.x86_64.rpm`
- Для Ubuntu:

- `apt-get remove iwtm-rocketchat_7.11.1.300-release_amd64.deb`
- `dpkg -r iwtm-rocketchat_7.11.1.300-release_amd64.deb`

7.2 Адаптер, развернутый в Docker-контейнере

Для запуска Адаптера:

1. Перейдите в директорию, в которой в процессе первичной настройки были созданы папки `etc`, `log` и `fstorage`.
2. Для запуска контейнера в интерактивном режиме выполните команду:
   `docker run --name iw_rocketchat -it -v $(pwd)/etc:/opt/iw/tm5/etc -v $(pwd)/log:/opt/iw/tm5/log -v $(pwd)/fstorage:/opt/iw/tm5/fstorage iw_rocketchat`

   > **Примечание:**
   >
   > Если контейнер запущен в интерактивном режиме, при отключении терминала, он будет остановлен.
   >
   > Если контейнер запущен в фоновом режиме, при отключении терминала он продолжит работу.
   >
   > Для запуска контейнера в фоновом режиме выполните команду:
   > `docker run --name iw_rocketchat -d -v $(pwd)/etc:/opt/iw/tm5/etc -v $(pwd)/log:/opt/iw/tm5/log -v $(pwd)/fstorage:/opt/iw/tm5/fstorage iw_rocketchat`
   >
   > Команды запуска, приведенные в инструкции, предполагают, что вы использовали все параметры по умолчанию: не меняли имя образа (`iw_rocketchat`), а также запускаете адаптер из директории с папками `etc` , `log` и `fstorage` .
   > Если вы хотите запустить контейнер не из директории с папками, вместо `pwd` укажите путь до директории, содержащей нужную папку.

   В результате будет запущен контейнер с именем `iw_rocketchat`.

Для остановки Адаптера выполните команду:
`docker stop iw_rocketchat`

Для удаления Адаптера остановите контейнер и удалите образ с помощью команды:
`docker image rm iw_rocketchat`