Личный кабинет

- Шаблоны писем
  - Редактор кода
  - Редактор Stripo
- Рассылки
  - Создание массовой рассылки
- API
  - Изменение статуса мультилида
  - Изменение статуса заказа
  - Общая информация
  - Передача бонусных баллов
- Аналитика
  - Виджеты
- Оповещения
  - Рассылка оповещений на webhook

Шаблоны писем

Редактор кода

Редактор кода поддерживает HTML и MJML разметку.

Функции

Функции для вызова списка товаров

Общая информация

Эти функции используются в циклах **for** для передачи данных о товарах из фида в шаблон.

Начало цикла:

```
{% for %}
```

Конец цикла:

```
{% endfor %}
```

Вызов функции возвращает список товаров, каждая итерация цикла возвращает один товар, чтобы ограничить число итераций и соответственно количество товаров в конкретном блоке, можно использовать метод slice с помощью квадратных скобок [начало:конец], например [0:2] означает, что нужно запустить цикл по первым двум товарам, [2:4] - по второму и третьему, отсчёт начинается с 0.

Пример цикла по двум первым товарам из списка товаров в корзине:

```
{% for item in get_cart_items() [0:2] %}
{% endfor %}
```

Товары из корзины

```
get_cart_items()
```

Просмотренные товары

```
get_viewed_items()
```

Товары из сервиса рекомендаций

Нужно указать алгоритм сервиса рекомендаций через параметр **service_name**

```
get_recommendations(service_name='top_offers')
```

Этот список необходимо определить в отдельную переменную, чтобы избежать дублей товаров, так как каждый вызов функции возвращает новый список товаров:

```
{% set rec_offers = get_recommendations(service_name='top_offers') %}
```

Не все **алгоритмы рекомендаций** можно использовать в рассылках, доступные к использованию отмечены значком конверта.

Случайные товары из фида

Можно указать необходимое количество товаров в списке через параметр **count**

```
get_random_offers(count=3)
```

Этот список необходимо определить в отдельную переменную, чтобы избежать дублей товаров, так как каждый вызов функции возвращает новый список товаров:

```
{% set random_offers = get_random_offers(count=10) %}
```

Пример

Создаём переменную **random_offers** со списком случайных товаров, указываем количество - 10

```
{% set random_offers = get_random_offers(count=10) %}
```

Запускаем цикл по товарам в горизонтальном блоке с товарами из созданной переменной со списком и ограничиваем количество товаров до двух:

```
<mj-section>
  {% for item in random_offers[0:2] %}
  <mj-column>
```

```
    <mj-image src="{{item.picture}}" />
    <mj-text>
      {{item.name}}
    </mj-text>
    <mj-text>
      {{item.price}}
    </mj-text>
    <mj-button href="{{item.url}}">Купить</mj-button>
  </mj-column>
  {% endfor %}
</mj-section>
```

Получаем 2 товара в ряд с изображением, названием, ценой и кнопкой, ведущей на

items_row.png"

страницу товара: Image not found or type unknown

Дополнительные функции

Получить случайный промокод из импортированного списка

```
{{get_promocode()}}
```

Отменить формирование шаблона

```
{{exit()}}
```

Важно: функция exit не работает в массовых рассылках

Переменные товаров

Эти переменные используются в циклах товарных функций, названия соответствуют тегам в фиде

```
{{item.picture}}
{{item.name}}
```

```
{{item.price}}
{{item.oldprice}}
{{item.url}}
{{item.type_prefix}}
{{item.vendor}}
{{item.model}}
```

Переменные лида

Переменная	Значение
{{lead.name}}	Имя
{{lead.last_name}}	Фамилия
{{lead.position}}	Должность
{{lead.email}}	Адрес электронной почты
{{lead.phone}}	Номер телефона
{{lead.region}}	Регион
{{lead.country}}	Страна
{{lead.city}}	Город
{{lead.address}}	Адрес
{{lead.company}}	Компания
{{lead.birthday}}	День рождения
{{lead.sex}}	Пол

Остальные переменные

Ссылка для открытия письма в браузере (веб-версия письма)

```
{{webview_url}}
```

Переменная используется в качестве значения атрибута **href**

```
href='{{webview_url}}'
```

Ссылка для отписки от рассылок

```
{{unsubscribe_url}}
```

Переменная используется в качестве значения атрибута **href**

```
href='{{unsubscribe_url}}'
```

Редактор Stripo

Помимо HTML и MJML верстки можно воспользоваться редактором Stripo - визуальный редактором писем, который не требует знаний языков разметки, письмо создаётся путём перетаскивания уже готовых элементов из меню и их стилизацией.

77.png

Image not found or type unknown

У Stripo есть своя инструция по созданию писем, ознакомиться с ней можно по **этой ссылке**.

Рассылки

Создание массовой рассылки

Чтобы отправить массовую рассылку, нужно перейти в раздел "Рассылки" в личном кабинете и нажать кнопку "Создать рассылку".

88.png

Image not found or type unknown

Адресат

Кому

11.png

В выпадающем меню можно выбрать "одиночный адресат" или "список клиентов" Image not found

При выборе одиночного адресата можно отправить рассылку по одному адресу, например, для теста. При выборе списка клиентов можно отправить рассылку сразу всей базе клиентов или какому-либо сегменту.

Адрес

Это поле используется для указания почтового адреса для отправки рассылки одиночному адресату. При рассылке по списку клиентов это поле остаётся пустым.
22.png

Image not found or type unknown

Имя списка

В выпадающем меню будут отображены стандартные списки и все созданные вами списки

33.png

из раздела "Списки клиентов". Image not found or type unknown

Для отправки рассылки по всей базе нужно выбрать список "Все клиенты".

Выбор шаблона рассылки

В выпадающем списке нужно выбрать шаблон письма, который хотите отправить. Можно воспользоваться поиском по шаблонам.

44.png

Image not found or type unknown

Параметры рассылки

Имя рассылки

Название вашей рассылки, которое будет отображено в разделе "Рассылки".

Почта отправителя

Адрес электронной почты, который будет использоваться в качестве отправителя, в почтовом клиенте этот адрес будет указан в поле "От кого".

Важно: используйте только корпоративный адрес электронной почты вашего сайта, для которого были настроены DNS записи.

Имя отправителя

Имя отправителя, которое будет отображаться в почтовом клиенте в поле "От кого".

Тема письма

Тема письма

Время отправки

Дата и время отправки рассылки. Для выбора даты и времени нужно нажать на иконку календаря. Если оставить поле пустым, рассылка отправится сразу после нажатия кнопки "Отправить".

Размер сегмента и Интервал между сегментами

Если необходимо отправить рассылку частями, то можно указать размер сегмента и интервал между их отправками в минутах, например 10000 писем с интервалом в 30 минут.

По умолчанию указано 1000 писем с интервалом 0 минут, что значит, что рассылка будет отправлена сразу по всей базе без задержки.

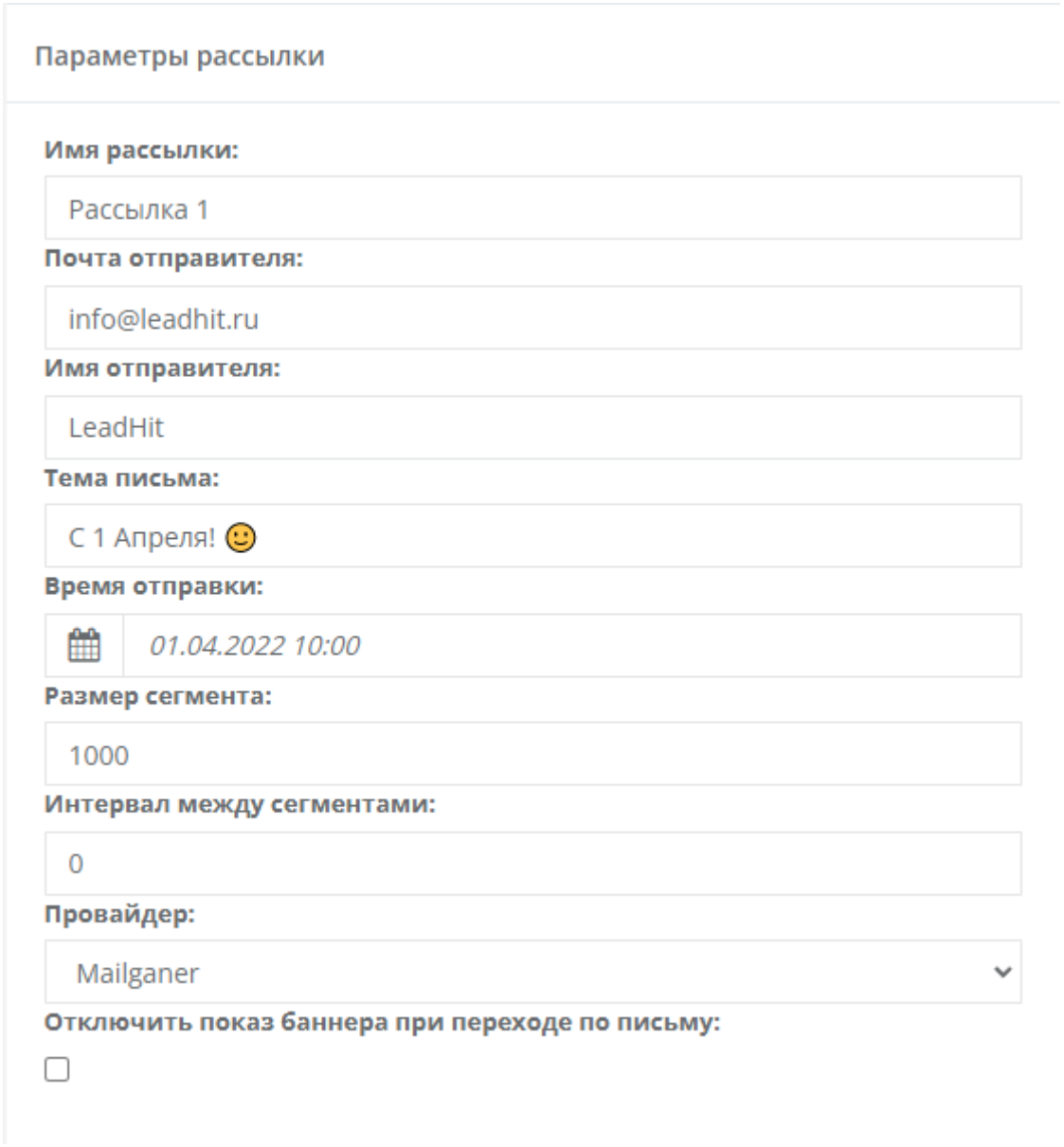
Провайдер

Mailganer

Отключить показ баннера при переходе по письму

При выборе этой опции после перехода из письма на сайт не будет показываться виджет "Смартоффер".

Пример заполненных параметров рассылки



Отправка рассылки

После заполнения всех полей нажмите кнопку "Отправить". Рассылка начнёт формироваться и вас перенаправит на страницу "Рассылки".

API

Изменение статуса мультилида

Адрес: **https://track-api.leadhit.io/client/update_multilead/**

Описание

Изменение статуса мультилида и всех привязанных к нему лидов.

Может пригодиться, если нужно ограничить отправку писем отпределенным лидам, таким лидам можно выставлять отличный от дефолтного статус, а в условиях рассылок запретить/разрешить отправку писем лидам с данным статусом.

Параметры

Метод - POST

В заголовках нужно передать "api-key" и "leadhit-site-id".
Параметры - json с полями "email", "status", "site_id".

Статус должен быть одним из списка: **"new", "contacted", "customer", "manager", "regular", "wholesale"**.
По умолчанию у мультилида установлен статус "new".

Api-key и site-id предоставляем мы.

Пример использования

```
POST https://track-api.leadhit.io/client/update_multilead/
```

Headers (application/json):

```
api-key: <api-key>
leadhit-site-id: <your_site_id>
```

Body (application/json):

```
{
  "email": "example@mail.ru",
  "status": "contacted",
  "site_id": "<your_site_id>"
}
```

Response 200 (application/json):

```
{
  "status": "success",
  "message": "Multilead with email example@mail.ru was updated. New status - contacted."
}
```

Возможные ошибки

Если покупатель с таким email не найден:

Response 404 (application/json):

```
{
  "detail": "Multilead not found"
}
```

Пример запроса с помощью CURL:

```
curl --request POST \
  --url https://track-api.leadhit.io/client/update_multilead/ \
  --header 'Content-Type: application/json' \
  --header 'api-key: <api-key>' \
  --header 'leadhit-site-id: <your_site_id>' \
  --data '{
  "email": "example@mail.ru",
  "status": "contacted",
  "site_id": "<your_site_id>"
}'
```

Изменение статуса заказа

Адрес: **https://service.leadhit.ru/api/v1/orders**
Метод: **set_order_status**

Описание

Изменение статусов заказов можно использовать для настройки триггерных рассылок.

Параметры

Метод: **POST**
Заголовки: **api-key**
Параметры:

- **JSON** с полями **"order_id"**, **"status"**;
- Версия JSON-RPC: **2.0**;
- ID запроса: **0**.

Статус должен быть одним из списка: **"submitted", "paid", "delivered", "cancelled"**.
После получения нами заказа, ему выставляется статус "submitted" - оформлен.

API-key запросите у нас.

Пример использования

```
POST https://service.leadhit.ru/api/v1/orders
```

Headers (application/json):

```
api-key: <api-key>
```

Body (application/json):

```
{
  "method": "set_order_status",
  "params": {
    "order_id": "f168148717",
    "status": "paid"
```

```
  },
  "jsonrpc": "2.0",
  "id": 0
}
```

Response 200 (application/json):

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": "ok",
    "message": "Order status changed"
  },
  "id": 0
}
```

Возможные ошибки

- Если номер заказа не найден:
  Response 200 (application/json):

```
{
  'message': "Order with this id doesn't exist"
}
```

- Если неправильно указан статус заказа:
  Response 200 (application/json):

```
{
  'message': 'Possible statuses: cancelled, delivered, paid, submitted'
}
```

Пример запроса с помощью CURL:

```
Curl:
```sh
$ curl -i \
    -H "Accept: application/json" \
    -H "Content-Type: application/json" \
    -H "API-KEY: 4f8425fc77796b0266000009:TKW0i5FP5vkIrgMlnpoZbfOEagA" \
    -X POST -d
```
```

```
'{"method":"set_order_status","params":{"order_id":"f168148717","status":"paid"},"jsonrpc":"2.
0","id":0}' \
    https://service.leadhit.ru/api/v1/orders/

>>> {"jsonrpc": "2.0", "result": {"status": "ok", "order_status": "paid"}, "id": 0}
```

Общая информация

Терминология

- **Узел** - конечный URL, к которому отправляются запросы. Например - `https://service.leadhit.ru/api/v1/orders/`.
- **Методы** - методы, которые принимают узлы, являются обычными питоновскими функциями.

API

API располагается по адресу https://service.leadhit.ru/api/v1/

- Основной протокол - JSON-RPC 2.0
- Content-type - application/json

Карта методов

У каждого узла API есть специальная страница - карта методов. На этой странице можно увидеть все существующие методы данного узла, а также дополнительную информацию о них.

Ее можно найти, если выполнить GET-запрос (или зайти через браузер), добавив к интересующему вас URL узла в конце `/map`.

Пример карты методов:

`https://service.leadhit.ru/api/v1/orders/map`

Авторизация

Чтобы пользоваться API, сначала необходимо получить API key. Обратитесь к своему менеджеру за ним.

После того, как вы получили ключ, вам необходимо передать его в заголовке POST-запроса к API. Он должен иметь название API-KEY (или api-key).

В случае, если API не получило авторизационный ключ в заголовке, вы получите ответ 403.

Базовый запрос

Общение с API происходит по протоколу JSON-RPC 2.0, через POST-запросы.
Content-type должен быть `application/json`.
Не забывайте об авторизации (заголовок API-KEY).

Помимо этого, каждый запрос к API должен содержать в себе три обязательных параметра - метод, версия JSON-RPC, id запроса. JSON-RPC всегда равен 2.0, а id может быть любым, такой же id вернется в ответе на запрос:

```
"method": "example_method"
"jsonrpc": "2.0"
"id": 0
```

Помимо обязательных параметров, методы принимают аргументы, котороые передаются через "params".
Могут быть либо списком, либо словарем.

В качестве примера - метод **set_order_status**.

```
Запрос:


{
  "method": "set_order_status",
  "params": {
    "order_id": "f168148717",
    "status": "paid"
  },
  "jsonrpc": "2.0"
  "id": 0
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": "ok",
    "message": "Order status changed"
```

```
  },
  "id": 0
}
```

Ошибки

В случае, если вы не авторизованы, сервер вернет статус 403.
Если вы ошиблись в параметре, или произошла какая-то другая ошибка, сервер вернет статус 200, а в теле ответа будет содержаться ключ **error**, в котором описана причина ошибки.

```
{
  "jsonrpc": "2.0",
  "id": 0,
  "error": {
    "message": "Server error",
    "code": -32000,
    "data": {
      "message": "Possible statuses: paid, submitted",
      "args": [
        "Possible statuses: paid, submitted"
      ],
      "type": "ValidationError"
    }
  }
}
```

Передача бонусных баллов

Передача данных

Данные о количестве бонусов покупателя передаются клиентом через API по адресу https://track-api.leadhit.io/client/multilead_bonus

Метод - POST

В **заголовках** нужно передать "api-key" и "leadhit-site-id" со значениями API ключа и id сайта. Значение для **api-key** запросите у своего аккаунт-менеджера.

Параметры - json с полями "email", "bonus". В поле `bonus` передается количество бонусов покупателя, в формате `float`.

Заголовок запроса:

```
api-key: <api-key>
leadhit-site-id: <your_site_id>
```

Тело запроса:

```
[{
 "email": "exists_lead@mail.ru",
 "bonus": 123.00
},
{
 "email": "not_exists_lead@mail.ru",
 "bonus": 10.00
}]
```

Примеры ответов на запрос

Response 200 (application/json)

При успешной передаче списка будет выдаваться статус 200 с ответом в виде списка со статусами по каждому обновлению бонусов у лида. Все обновления, которые не были сделаны будут в конце списка со статусом “error”:

```
[{
 "status": "success",
 "message": "Bonuses for multilead with email exists_lead@mail.ru  updated."
},
{
 "status": "error",
 "message": "multilead with email not_exists_lead@mail.ru not found"
}]
```

Response 400 (application/json):

```
{
   "detail": Body is empty"
}
```

При передачи списка с любыми невалидными данными будет выдаваться ошибка 422 с описанием полей, где данные некорректны

Response 422 (application/json)

```
{
       "detail": [
    {
      "loc": ["string"],
      "msg": "string",
      "type": "string"
    }
       ]
}
```

Response 500 (application/json)

При получении ошибки 500 необходимо проверить данные на корректность и повторить запрос через несколько минут. Если ошибка повторяется, то уведомить с примерами данных, с которыми возникает ошибка для последующего анализа

```
{
“detail”: “Internal error”
 }
```

Аналитика

Виджеты

Описание

Виджеты отправляют данные по событиям, на основе которых формируется аналитика по виджетам в личном кабинете.

События виджетов доступны для отслеживания на сайте через ивент `widgetStatsSended`. Если нужно отправлять данные о событиях виджетов в систему аналитики (Google Analytics, Яндекс Метрика), то можно добавить JS обработчик на нужные события, указав id виджета и название события, например:

```
document.addEventListener('widgetStatsSended', function (e) {
  if (e.detail.wid == 'cb509' && e.detail.etype == 'popup_view') {
    // cb509 - id виджета; popup_view - название события
    ... вызов функции системы аналитики (reachGoal, gtag, ga) для отправки события ...
  }
});
```

Идентификаторы виджетов можно посмотреть в личном кабинете в разделе Аналитика > По виджетам, кликнув на виджет откроется страница с аналитикой по виджету, последняя часть url страницы и есть идентификатор виджета.

Вы можете настроить отслеживание событий самостоятельно или запросить у своего менеджера настройку средствами LeadHit. Для этого нам потребуются названия целей и соответствующая им система аналитики.

Полезные ссылки:

- Настройка целей в Яндекс Метрике
- Настройка целей в Google Analytics [UA]
- Настройка целей в Google Analytics 4 [GA4]

Типы событий по виджетам

Название виджета:

- описание события | название события для отслеживания

Смартоффер (десктоп)

- Показ виджета | popup_view
- Показ виджета более 3 секунд | popup_view_3s
- Показ виджета более 10 секунд | popup_view_10s
- Заполнение виджета | fill
- Закрытие виджета | close

Смартоффер (мобильные устройства)

- Показ ярлыка виджета | view
- Показ окна виджета | popup_view
- Заполнение виджета | fill
- Закрытие виджета | close
- Сворачивание виджета | roll_up

Колесо фортуны (десктоп)

- Показ виджета | popup_view
- Показ виджета более 3 секунд | popup_view_3s
- Показ виджета более 10 секунд | popup_view_10s
- Заполнение виджета | fill
- Закрытие виджета | close

Колесо фортуны (мобильные устройства)

- Показ ярлыка | view
- Показ окна | popup_view
- Показ окна более 3 секунд | popup_view_3s
- Показ окна более 10 секунд | popup_view_10s
- Заполнение формы | fill
- Закрытие | close

Scratch

- Показ виджета | popup_view
- Показ виджета более 3 секунд | popup_view_3s
- Показ виджета более 10 секунд | popup_view_10s
- Заполнение виджета | fill
- Закрытие виджета | close

Вишлист

- Показ ярлыка виджета | view

- Показ окна виджета | popup_view
- Показ виджета более 3 секунд | popup_view_3s
- Показ виджета более 10 секунд | popup_view_10s
- Заполнение виджета | fill
- Закрытие виджета | close

Подписка на товары

- Показ виджета | popup_view
- Заполнение виджета | fill

Social Proof

- Показ виджета | view

Купон

- Показ виджета | popup_view
- Закрытие виджета | close

Notification

- Показ виджета | view
- Закрытие виджета | close

Брошенная корзина

- Показ виджета | view
- Клик | click
- Закрытие виджета | close

Последние просмотренные товары

- Показ виджета | view
- Клик | click
- Закрытие виджета | close

Бустер среднего чека

- Показ виджета | popup_view
- Показ окна с мотивацией | show_motivation
- Показ финального окна | show_promo
- Сворачивание | roll_up

Рекомендации

- Показ виджета | view

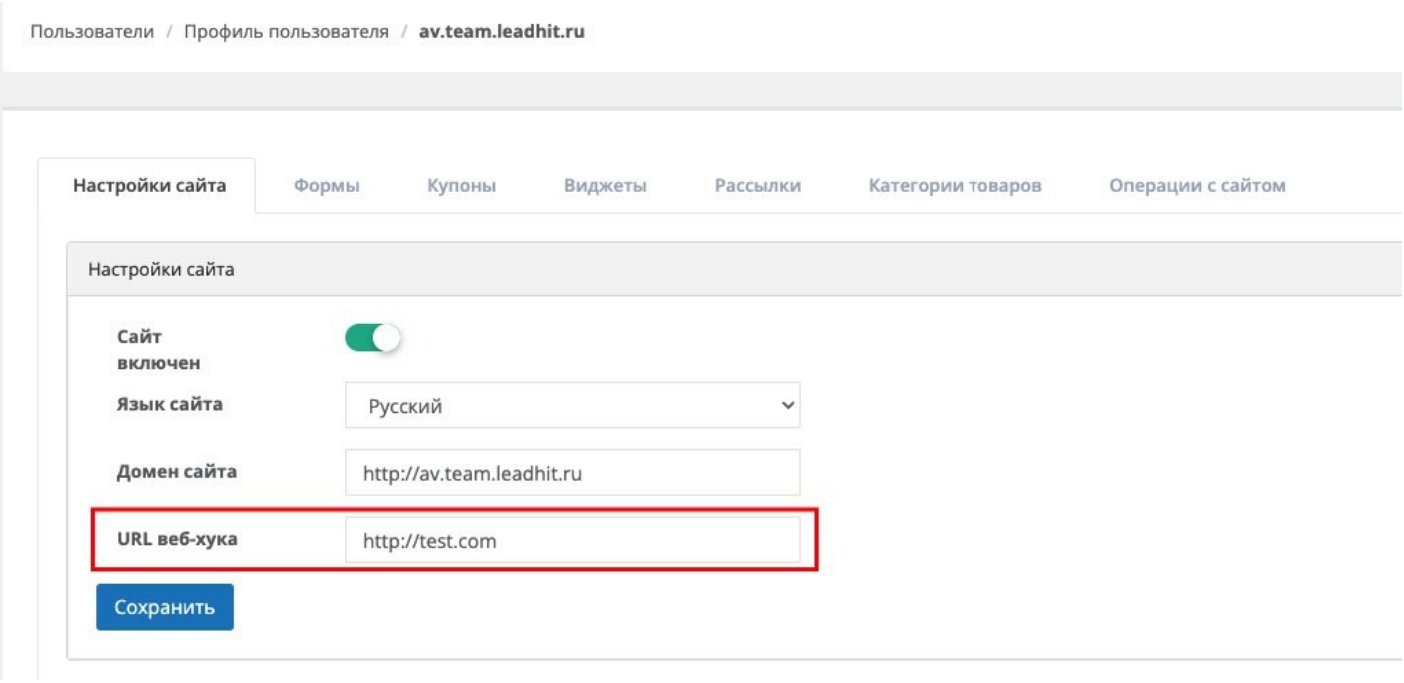
- Клик | click

Оповещения

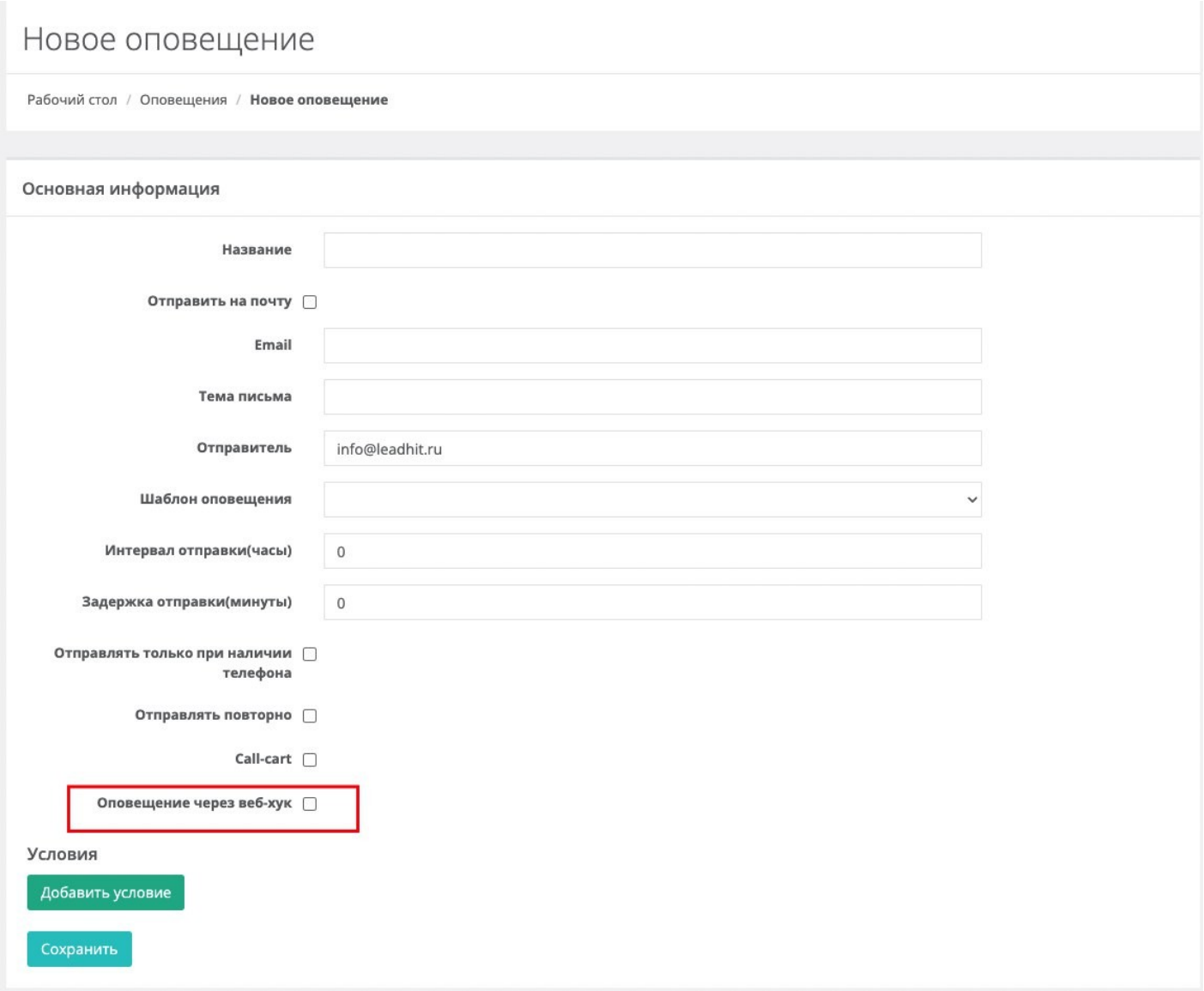
Рассылка оповещений на webhook

Данные из оповещений можно получать через webhook.

Для этого в настройках сайта нужно указать **URL веб-хука**, на который будет приходить webhook.



Включить параметр **оповещение через веб-хук** в настройках оповещения.



Как только оповещение сработает, на указанный url придёт post-запрос с данными.

Пример

```
{'alert_name': 'Alert Name',
 'domain': 'http://your_site.ru',
 'items': [
            {
               'available': true,
               'category': 'Product Category',
               'category_id': '4478932',
               'curr': 'руб.',
               'description': 'Some Description',
               'id': '62bbfe9f07651d3e10919cda',
               'min_url': 'url',
```

```
                'model': 'Product Model',
                'name': 'Product Name',
                'offer_id': '999999999',
                'oldprice': '',
                'picture': 'https://media.leadhit.io/picture.jpg',
                'price': '99999.99',
                'stocks': {},
                'type_prefix': 'Product Prefix',
                'url': 'https://your_site.ru/collection/product/your_product',
                'vendor': 'Vendor Name'
            }
          ],
 'last_form_with_phone': 'Корзина',
 'last_form_with_phone_date': '2022-06-29 12:17',
 'last_form_with_phone_url': 'https://your_site.ru/new_order',
 'lead_channel': 'direct',
 'lead_email': 'Н/Д',
 'lead_form': 'Корзина',
 'lead_form_add1': '',
 'lead_form_add2': '',
 'lead_form_url': 'https://your_site.ru/new_order',
 'lead_id': '62bbe358eec3047b1b05cbd1',
 'lead_index': '',
 'lead_ip': '999.999.999.999',
 'lead_name': 'Lead Name',
 'lead_phone': '8(999)999-99-99',
 'lead_region': '',
 'lead_time_added': '29.06.2022 12:17:28',
 'lead_visits': '1'}
```