

Интеграция сервиса

- Добавление кода на сайт
 - Установка счетчика
 - Установка кода заказа
 - Готовые модули для CMS
- Импорт товаров интернет-магазина
 - Подготовка фида (файла с товарами интернет-магазина)
 - Подготовка регионального фида
- Сервис рекомендаций
 - Алгоритмы сервиса
 - Спецификация получения рекомендаций по RSAPI
- DNS записи
 - Переход на новый транспорт для рассылок
- Дополнительные настройки отдельных инструментов
 - Виджет "Бустер среднего чека"
 - Рассылка "Брошенный просмотр категории товаров"
 - Виджет "Брошенная корзина"

Добавление кода на сайт

Добавление кода на сайт

Установка счетчика

Счетчик необходимо добавить на каждую страницу сайта.

При установке счетчика в HTML-код сайта, разместите код как можно ближе к началу страницы, в пределах тегов `<head></head>` или `<body></body>`.

Пример кода счетчика:

```
<script type="text/javascript" language="javascript">
  var _lh_params = {
    "popup": false
  };

  // Идентификатор личного кабинета
  lh_clid = 'CLIENT_ID_HERE';

  (function() {
    var lh = document.createElement('script');
    lh.type = 'text/javascript';
    lh.async = true;
    lh.src = ('https:' == document.location.protocol ? 'https://' : 'http://') +
' track.leadhit.io/track.js?ver=' + Math.floor(Date.now() / 100000).toString();
    var s = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(lh, s);
  })();
</script>
```

Идентификатор личного кабинета

Это идентификатор личного кабинета для вашего сайта, вы найдете его в инструкции по интеграции, которую вам предоставят наши менеджеры.

Добавление кода на сайт

Установка кода заказа

Код заказа необходимо разместить на финальную страницу заказа, где пользователя информируют об успешном оформлении заказа.

В код нужно передать данные о номере заказа, сумме и товарах.

Пример кода заказа:

```
<script type="text/javascript" class="lh-order-script">
(function () {

    // EDIT VALUES BELOW
    var order_id = 'ORDER_ID_HERE'; // Номер заказа
    var cart_sum = 'CART_SUM_HERE'; // Сумма заказа
    var order_offers = []; // Товары в заказе
    // END OF EDITABLE VALUES

    function readCookie(name) {
        if (document.cookie.length > 0) {
            let offset = document.cookie.indexOf(name + "=");
            if (offset != -1) {
                offset = offset + name.length + 1;
                let tail = document.cookie.indexOf(";", offset);
                if (tail == -1) tail = document.cookie.length;
                return unescape(document.cookie.substring(offset, tail));
            }
        }
        return null;
    }

    function genUID() {
        return 'xxxxxxxxxxxxxxxxxxxxxxxx'.replace(/x/g, function () {
            var rand;
            try {
                rand = crypto.getRandomValues((new Uint8Array(1)))[0] % 16;
            } catch (e) {
                rand = Math.floor(Math.random() * 16);
            }
        });
    }
});
```

```

    }
    return rand.toString(16);
  });
}

var site_id = 'SITE_ID_HERE'; // Идентификатор личного кабинета
var lead_id = readCookie('_lhtm_u');
var url = window.location.href;
var form_id = genUID();
var form_fields = {
  'orderid': order_id,
  'cart_sum': cart_sum
};

var request_body_form = {
  'site_id': site_id,
  'lead_id': lead_id,
  'form_id': form_id,
  'url': url,
  'referrer': 'direct',
  'action': 'lh_orderid',
  'form_fields': form_fields
};

var xhr_form = new XMLHttpRequest();
xhr_form.open('POST', 'https://track-api.leadhit.io/save/lead/form', true);
xhr_form.setRequestHeader('Content-type', 'application/json; charset=utf-8');
xhr_form.send(JSON.stringify(request_body_form));

if (Array.isArray(order_offers) && order_offers.length > 0) {
  var request_body_order = {
    'site_id': site_id,
    'lead_id': lead_id,
    'order_id': order_id,
    'cart_sum': cart_sum,
    'items': order_offers
  };

  var xhr_order = new XMLHttpRequest();
  xhr_order.open('POST', 'https://track-api.leadhit.io/save/lead/order', true);
  xhr_order.setRequestHeader('Content-type', 'application/json; charset=utf-8');
  xhr_order.onreadystatechange = function () {

```

```
    if (this.readyState != 4) return;
    console.log(' order sended' );
};
xhr_order.send( JSON.stringify(request_body_order));
}
})();
</script>
```

Идентификатор личного кабинета

Это идентификатор личного кабинета для вашего сайта, вы найдете его в документе с инструкцией по интеграции, которую вам предоставят наши менеджеры. Идентификатор уже будет указан в переменной `site_id`, его нужно оставить без изменений.

Номер заказа

Номер заказа укажите в переменной `order_id`, тип данных - string (строка)

```
var order_id = 'ORDER_ID_HERE'
```

Сумма заказа

Сумму заказа укажите в переменной `cart_sum`, тип данных - string (строка)

```
var cart_sum = 'CART_SUM_HERE'
```

Формат суммы должен быть 200.00 - разделитель точка и 2 знака после точки. В случае несоблюдения этого правила, суммы заказа будут сохраняться некорректно, что приведет к ошибкам в данных.

Товары в заказе

Товары укажите в переменной `order_offers`

```
var order_offers = []
```

Товары представляют из себя массив объектов. Каждый объект товара содержит следующие атрибуты:

- url: URL товара.
URL должен быть полным, с указанием префикса протокола (http://, https://) и доменом, а так же должен совпадать с URL этого же товара в YML файле (фиде).
Типа данных - строка (string)
- name: Название товара.
Тип данных - строка (string)
- price: Стоимость товара.
Тип данных - число (number) или строка (string)

- count: Количество купленных экземпляров товара.
Тип данных - число (number) или строка (string)
- currency: Код валюты в формате ISO 4217 (https://ru.wikipedia.org/wiki/ISO_4217).
Тип данных - строка (string)

Все атрибуты обязательны к заполнению.

Пример массива с товарами:

```
var order_offers = [  
  {  
    'url': 'https://medieval-shop.io/offers/broken_sword',  
    'name': 'Сломанный меч',  
    'price': 500,  
    'count': 1,  
    'currency': 'RUB'  
  },  
  {  
    'url': 'https://medieval-shop.io/offers/excellent_helmet',  
    'name': 'Превосходный шлем',  
    'price': 2000,  
    'count': 3,  
    'currency': 'USD'  
  }  
]
```

ГОТОВЫЕ МОДУЛИ ДЛЯ CMS

С помощью модулей (расширений для CMS) установка счётчика и кода заказа будет произведена автоматически.

CS-Cart

Инструкция по установке модуля CS-Cart:

1. После покупки модуля на указанную вами почту придёт письмо, в котором будет прикреплён архив для скачивания модуля.
2. После того, как вы скачаете модуль, в административном разделе необходимо перейти на вкладку "Модули" -> "Управление модулями" и нажать на значок "+" справа.
3. Необходимо загрузить модуль из папки, куда вы его скачали в виде архива и нажать на кнопку "Загрузить и установить". После успешной установки модуль LeadHit появится в списке ваших модулей.
4. При нажатии по ссылке в общем списке модулей будет выведено поле для ввода ID вашего магазина. Необходимо ввести ID, найти его можно в коде счётчика в переменной `lh_clid`.

Ссылка на модуль: <https://marketplace.cs-cart.com/leadhit-en.html>

Импорт товаров интернет-магазина

Подготовка фида (файла с товарами интернет-магазина)

Для товаров магазина мы используем файл формата xml по стандарту Яндекс.Маркета (YML), при создании файла следуйте [документации Яндекса](#).

Атрибуты элемента `<offer></offer>`, которые мы загружаем из фида:

- id
- name
- url
- price
- oldprice
- picture
- description
- typePrefix
- vendor
- model
- categoryId

Корректность формата файла можно проверить через [валидатор XML-фидов](#), во вкладке “Маркет”.

Подготовка регионального фида

Общее

Функционал региональности позволяет получать локальную информацию о товарах (например цену и наличие), актуальную для конкретного региона (страны / города).

В свою очередь это позволяет:

- отображать в рекомендациях товары, доступные в конкретном регионе со специфичной к этому региону ценой;
- в email-рассылках использовать информацию о товаре, специфичную для региона, к которому относится пользователь.

Изменения YML для поддержки региональности

Для передачи специфичной для региона информации о товаре через yml необходимо в тег `offer` добавить блок `stock`. Пример:

```
<offer id="1039485" available="true">
  <price>100.00</price>
  <url>http://example.com/pathToProduct</url>
  <stock id="14"> К примеру, этот stock отвечает за Екатеринбург
    <available>true</available> Наличие товара в Екатеринбурге
    <price>200.00</price> Цена товара в Екатеринбурге
    <url>http://ekb.example.com/pathToProduct</url> Ссылка на товар на поддомене для
Екатеринбурга
  </stock>
  <stock id="15"> К примеру, этот stock отвечает за Санкт-Петербург
    <available>false</available> Наличие товара в Санкт-Петербурге
    <price>250.00</price> Цена товара в Санкт-Петербурге
    <url>http://spb.example.com/pathToProduct</url> Ссылка на товар на поддомене для Санкт-
Петербурга
  </stock>
```

```
</offer>
```

В данный момент каталог товаров поддерживает кастомизацию следующих атрибутов:

- available
- price
- oldPrice
- url
- currencyId

Требования к описанию атрибутов региона:

- каждый элемент offer обязательно должен содержать все стоки;
- товары, у которых не указан ни один `stock`, должны быть не в наличии (`available="false"`);
- каждый элемент `stock` в рамках одного `offer` обязан иметь уникальные строковые идентификаторы, заданные атрибутом `id`;

Таким образом, при получении информации о товаре, привязанном к одному из объявленных регионов, данные о цене и наличии будут переопределены значениями элемента `stock`.

Передача stock id в локальное хранилище браузера

Кроме этого, на сайте необходимо создать ключ в локальном хранилище (**localStorage**) с именем `_lh_stock_id` и со значением, которое будет равно `stock id` в фиде.

То есть, если посетитель заходит на сайт `ekb.example.com`, а в фиде за Екатеринбург отвечает `stock id` со значением "2" то нужно, чтобы создавался ключ `_lh_stock_id` со значением "2", в этом случае сервис будет использовать данные о товарах из фида ориентируясь на элемент `stock`, у которого `stock id="2"`.

Сервис рекомендаций

Алгоритмы сервиса

Cross-sale

Рекомендации формируются на основе истории посещений и действий пользователей сайта.

Учитываются просмотры, добавления в корзину и покупки.

☒ Алгоритм доступен к использованию в письмах.

Cart-upsale

Рекомендации формируются за счёт товаров, которые находятся в корзине.

Проверяется история корзинного соседства этих товаров с другими и рекомендуются последние.

☒ Алгоритм доступен к использованию в письмах.

Offer-upsale

Рекомендации работают в том же ключе, что и Cart-upsale рекомендации.

Только в данном случае проверяется корзинное соседство просматриваемого в данный момент товара.

Search

Поисковые рекомендации формируются с учетом последнего поискового запроса пользователя сайта.

Например, человек в поисковом поле ввёл слово "футболка". В этом случае в качестве рекомендованных товаров будут появляться товары со словом "футболка" в названии или в описании.

В алгоритме также учитываются простые орфографические ошибки.

Например, при поиске слова "фтболка" сервис вернёт рекомендации по слову "футболка".

Top-offers

Рекомендуются **самые покупаемые товары** за последние 14 дней на основе добавлений в корзину и заказов с этими товарами (если подключен код заказ с товарами). Просмотры товаров не учитываются.

У этого алгоритма есть опция "интервал", который позволяет указать за какой период нужны рекомендации самых покупаемых товаров.

Интервал по умолчанию - 14 дней

Алгоритм доступен к использованию в письмах.

Buying-now

Покупают сейчас

Рекомендуются купленные товары за последние 2 часа.

Алгоритм доступен к использованию в письмах.

Similar-offers

Похожие товары

Рекомендации формируются на основе схожести по категории и цене (+- 20%).

Last Viewed

Последние просмотренные товары

Рекомендуются товары, которые пользователь просматривал последними.

Опционально можно исключить товары, которые пользователь уже купил.

Спецификация получения рекомендаций по RSAPI

Для того, чтобы получить рекомендации необходимо сделать HTTP запрос (GET или POST) по следующему адресу **https://rs.leadhit.io/get**, передав необходимые параметры.

Пример запроса:

```
https://rs.leadhit.io/get?clid=123abc&lead_uid=456def&service_name=top_offers
```

Список возможных параметров:

- **clid** - идентификатор сайта. Значение хранится в глобальной javascript переменной **lh_clid**.
- **lead_uid** - идентификатор лида. Значение хранится в куки **_lhtm_u**, домен которой такой же, как у сайта.
- **service_name** - название сервиса рекомендаций, к которому отправится запрос.
- **source** - для товаров (и их категорий) из какого источника следует применять бизнес-правила. 'source' может быть 3 видов: 'cart' (по умолчанию) - товары из корзины лида, 'view' - товары из 100 последних просмотренных страниц лида, 'request' - страницы товаров из параметра 'url_list[]'
- **offer_url** - ссылка на товар. Параметр может использоваться в алгоритме 'offer_upsale' для поиска товаров, с которыми товар по ссылке, кладут в корзину.
- **url_list[]** - список ссылок на страницы товаров. Данный список используется в бизнес-правилах, если source='request', а также может использоваться в алгоритме 'cart_upsale' в качестве текущей корзины лида.
- **search_query** - поисковый запрос. Актуально для сервиса с поисковым алгоритмом (search). В ответе будут содержаться товары, в названия или в описания которых будет входить значение параметра **search_query**.
- **lead_history_days_count** - актуально для сервисов, использующих алгоритм матричного разложения (cross_sale). Параметр определяет период (в днях), за который учитываются действия лида, на основе которых генерируются рекомендации.
- **category_id** - категория товаров, которые будут в ответе. Используется в сервисах с типом 'top_offers'.
- **stock_id** - идентификатор стока, с которым необходимо получить товары. Необходимо использовать, если подключен региональный фид.

- **limit** - лимит количества товаров. По умолчанию сервис возвращает 20 товаров, но с помощью этого параметра количество товаров можно увеличить до 1000. Работает только с алгоритмом "top_offers".
- **extra_fields_filter** - фильтр по мета-тегам, например параметр `extra_fields_filter={"new": "true", "bestseller": "true"}` вернет товары с мета-тегами "new" и "bestseller" со значением "true". Работает только с алгоритмом "top_offers".

Алгоритмы и используемые в них параметры:

- Для корректной работы алгоритма **cart_upsale** необходимо передать **url_list[]** или **lead_uid** для поиска корзины клиента. Если переданы оба параметра, то используется **url_list[]**.
- Если в сервис с алгоритмом **offer_upsale** не передан параметр **offer_url**, в качестве ссылки на товар используется 'Referrer' из 'headers' запроса.
- В алгоритме **similar_offers** в качестве ссылки на целевой товар используется 'Referrer' из 'headers' запроса.
- Бизнес-правила на данный момент используются в следующих алгоритмах: **cross_sale, top_offers, offer_upsale, cart_upsale, buying_now**.

Параметр \ Алгоритм	cross_sale	top_offers	offer_upsale	cart_upsale	similar_offers	buying_now	search
clid	v	v	v	v	v	v	v
lead_uid	v	v	v	v	v	v	
service_name	v	v	v	v	v	v	v
source	v	v	v	v		v	
offer_url			v				
url_list[]	v	v	v	v		v	
search_query							v
lead_history_days_count	v						
category_id		v					
stock_id	v	v			v	v	
limit		v					
extra_fields_filter		v					

Формат ответа:

Рекомендации отдаются в JSON формате, и представляют из себя список объектов следующего вида:

```
{
  "available": true,
  "category_id": "20102",
  "cities": [],
  "description": "Air Jordan Legacy 312 – Гибридная модель, использующая элементы Air Jordan 1, Air Jordan 3 и Alpha Force Low.",
  "extra_fields": {},
  "model": "Air Jordan",
  "name": "Кроссовки Nike Air Jordan Legacy 312 Low",
  "offer_id": "2145678",
  "oldprice": "15000",
  "picture": "https://media.leadhit.io/upload/_noimage.png",
  "picture_orig": "https://www.site.ru/product/123456/image.png",
  "price": "12500",
  "rating": 1.0,
  "stock_fields": [],
  "url": "https://www.site.ru/product/123456/",
  "url_hash": "4c13e8d0f4b0",
  "vendor": "Nike"
}
```

Аналитика по рекомендациям

Аналитика задействует статистику по виджетам рекомендаций. Для отдельного блока товарных рекомендаций, который вы используете на сайте, мы создаем виджет. Для формирования статистики, необходимо отправить запрос с событием с помощью JS функции `Leadhit.sendWidgetStats`, используя идентификатор виджета в качестве одного из параметров.

Вам необходимо согласовать с менеджером, какие алгоритмы сервиса рекомендаций вы хотите использовать и где разместить, чтобы мы запустили эти алгоритмы и создали к ним виджеты для сбора статистики. После этого менеджер сообщит вам идентификаторы виджетов, которые нужно использовать для отправки событий.

Как отправить событие

Показ (view) — это событие нужно отправить после загрузки блока с рекомендациями:

```
Leadhit.sendWidgetStats('123', 'view');
```

где `123` — идентификатор виджета, `view` — тип события.

Клик (click) — это событие нужно отправить после клика на товар из блока рекомендаций:

```
Leadhit.sendWidgetStats('123', 'click', {'offer_url':  
'https://www.site.ru/product/123456/?lhsource=recommendations'});
```

где `123` — идентификатор виджета, `click` — тип события, `https://www.site.ru/product/123456/` — URL товара, `lhsource=recommendations` — сервисный get-параметр для подсчета атрибуции заказов.

Атрибуция заказов

Для того, чтобы товарные рекомендации учитывались в атрибуции заказов, необходимо добавить в URL товаров параметр `lhsource=recommendations`, например `https://www.site.ru/product/123456/?lhsource=recommendations`.

DNS записи

Переход на новый транспорт для рассылок

Для перехода на новый транспорт необходимо внести изменения в DNS-записях вашего сайта.

Внести изменения нужно для поддомена, который сейчас используется для рассылок LeadHit: **lh** или **post**

Важно: Текущие записи удалять не нужно.

Какие изменения нужно внести?

1. Добавить новую DKIM запись

Тип записи:

Хост:

Значение:

```
v=DKIM1; k=rsa;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDCLdYr TXR9EHFuESawj kF3BNFN2eyr gayZ3u4qkr d4+Zy+s ZcBuQXk
5xcaKjNj 7Z9WDoHs cyp1oaN+hS3LMdCs ZLBC168VUaexQzG10XnTvCI luWEtQA hF5Hoi2u9VJzZtL YcF/QhYLU6ZCb ggnb
Jtdjq uIfIHs xVvMzIKuqWo4wIDAQAB
```

2. Дополнить SPF запись

Тип записи:

Хост:

Текущее значение:

```
v=spf1 include: mailgun.org ~all
```

Нужно добавить:

```
include: spf.mailganer.com
```

Итоговое значение:

```
v=spf1 include: mailgun.org include: spf.mailganer.com ~all
```

Дополнительные настройки отдельных инструментов

Виджет "Бустер среднего чека"

Для корректной работы этого виджета сервису необходимо иметь доступ к актуальной сумме корзины.

Для этого мы просим вас передавать значение суммы корзины пользователя в локальное хранилище данных браузера (LocalStorage) в виде ключ/значение. Это значение должно изменяться в зависимости от действий пользователя (добавления товаров в корзину, изменение их количества, удаление товаров) даже без перезагрузки страницы.

Имя ключа - `lh_cart_sum`.

Вот простой пример JavaScript кода, который устанавливает значение ключа LocalStorage:

```
localStorage.setItem('lh_cart_sum', cart_sum_variable_name);
```

где `cart_sum_variable_name` - сумма корзины пользователя.

[Подробнее про LocalStorage](#)

Рассылка "Брошенный просмотр категории товаров"

Для работы триггерной рассылки "Брошенный просмотр категории товаров" необходимо предоставить файл со списком категорий товаров.

Список должен содержать название, URL и ID категории.

ID категорий в списке должны совпадать с ID категорий в фиде, так как исходя из сопоставления ID категорий система будет понимать, из какой категории нужно запросить товары для добавления в рассылку. Категория товара указывается в параметре `<categoryId>`.

Пример списка:

Название	URL	ID
Одежда	https://example.com/catalog/clothes/	1
Обувь	https://example.com/catalog/shoes/	2

Пример фида:

```
<categories>
  <category id="1">Одежда</category>
  <category id="2">Обувь</category>
</categories>
<offers>
  <offer id="784" available="true">
    <name>Пальто</name>
    <url>https://example.com/catalog/clothes/item-784/</url>
    <categoryId>1</categoryId> <!-- категория товара -->
  </offer>
  <offer id="265" available="true">
```

```
<name>Свитер</name>
<url>https://example.com/catalog/clothes/item-265/</url>
<categoryId>1</categoryId> <!-- категория товара -->
</offer>
<offer id="591" available="true">
  <name>Ботинки</name>
  <url>https://example.com/catalog/shoes/item-591/</url>
  <categoryId>2</categoryId> <!-- категория товара -->
</offer>
<offer id="803" available="true">
  <name>Кроссовки</name>
  <url>https://example.com/catalog/shoes/item-803/</url>
  <categoryId>2</categoryId> <!-- категория товара -->
</offer>
<offers>
```

В данном примере, если визит пользователя закончится на просмотре категорий (и просмотра карточек товара не было за визит), а последняя просмотренная будет <https://example.com/catalog/shoes>, то пользователь получит письмо с товарами, которые относятся к категории с идентификатором "2" - `<categoryId>2</categoryId>`.

Если вы используете подкатегории товаров, то важно учитывать, что дочерние категории товаров не будут учитываться, если в `<categoryId>` товара указана родительская.

Пример фида:

```
<categories>
  <category id="1">Одежда</category>
    <category id="11" parentId="1">Верхняя одежда</category>
    <category id="12" parentId="1">Брюки</category>
  <category id="2">Обувь</category>
    <category id="21" parentId="2">Ботинки</category>
    <category id="22" parentId="2">Кроссовки</category>
</categories>
<offers>
  <offer id="784" available="true">
    <name>Пальто</name>
    <url>https://example.com/catalog/clothes/outerwear/item-784/</url>
    <categoryId>11</categoryId> <!-- категория товара -->
  </offer>
  <offer id="591" available="true">
```

```
<name>Ботинки</name>
<url>https://example.com/catalog/shoes/boots/item-591/</url>
<categoryId>21</categoryId> <!-- категория товара -->
</offer>
<offers>
```

В данном случае список категорий должен выглядеть следующим образом:

Название	URL	ID
Верхняя одежда	https://example.com/catalog/clothes/outerwear/	11
Брюки	https://example.com/catalog/clothes/trousers/	12
Ботинки	https://example.com/catalog/shoes/boots/	21
Кроссовки	https://example.com/catalog/shoes/sneakers/	22

Виджет "Брошенная корзина"

Для работы виджета "Брошенная корзина" нужно хранить актуальные данные о состоянии корзины пользователя в **local storage** в формате **JSON**, название ключа:

```
lh_abandoned_cart_widget_data.
```

Содержимое ключа:

- **Общая сумма корзины**, формат - число(number);
- **Список товаров**:
 - **название товара**, формат - строка(string);
 - **ссылка на изображение товара**, формат - строка(string);
 - **ссылка на товар**, формат - строка(string);
 - **количество товаров**, формат - число(number);
 - **сумма стоимости товаров**, формат - число(number).

Если корзина пользователя пуста, данный ключ должен быть удален из local storage.

Пример:

```
{
  "total_cart_sum": 5000,
  "cart_items": [
    {
      "name": "Смартфон",
      "img": ".../smartphone.png",
      "link": ".../catalog/smartphone/",
      "count": 1,
      "price": 3000
    },
    {
      "name": "Наушники",
      "img": ".../headphones.png",
      "link": ".../catalog/headphones/",
      "count": 2,
      "price": 2000
    }
  ]
}
```

```
}
```

```
1
```

```
}
```