Интеграция сервиса [бета версия]

- Добавление кода на сайт
  - Установка счетчика
  - Установка кода заказа
- Импорт товаров интернет-магазина
  - Подготовка фида (файла с товарами интернет-магазина)
- API
  - Использование API

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
  };
  lh_clid = "5c8a1ff073efc3407f9e4603";
  (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);
  })(); /* "PLEASE DO NOT MAKE ANY CHANGES IN THIS JS-CODE!"*/
</script>
```

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">
  (function () {
    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 order_id = 'ORDER_ID_HERE'; // String
    var cart_sum = 'CART_SUM_HERE'; // String
```

```
var order_offers = []; // Offers list here

var site_id = '5e8727c3ffce38858fd1bbc7';
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>
```

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

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'
  }
]
```

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**.

Не все элементы фида из стандарта YML используются в нашей системе, а так же присутствуют и те, которых нет в стандарте.

Список используемых системой и доступных к использованию в шаблонизаторе писем элементов:

Элемент	Описание
offer_id	идентификатор товара
picture	ссылка на изображение товара
name	название товара
price	текущая цена товара
oldprice	старая цена товара
economy	экономия (разница между текущей и старой ценой)
economy_percent	экономия в процентах
url	ссылка на товар
description	описание товара
type_prefix	тип / категория товара
vendor	производитель
model	модель
category_id	идентификатор категории товара
curr	валюта
available	доступность товара / наличие

Корректность формата файла можно проверить через **валидатор XML-фидов**, во вкладке “Маркет”.

API

Использование API

Документация

Документацию можно найти на сайте https://cp-documentation.leadhit.io
Логин и пароль для авторизации в документации можно получить у аккаунт-менеджера.

Авторизация и аутентификация

Чтобы использовать API личного кабинета вам необходимо получить токен авторизации.

Страница в документации: Получение токена авторизации.

Токен долгоживущий, можно получить его один раз и использовать до тех пор, пока не получите в ответе код `401 Unauthorized`, после этого нужно получить новый токен.

Использование токена для аутентификации производится путем его передачи в заголовке.

Страница в документации: Авторизация/аутентификация.

Помимо токена вам также необходимо передавать в заголовке ваш site id. Его вы можете получить у аккаунт-менеджера или через API в методе получения информации о пользователе.

Страница в документации: Получить информацию о пользователе.

Предостережение

Пожалуйста, не используйте ваш токен на фронтенде в открытом виде.
Делайте все запросы через бэкэнд, чтобы никто не мог завладеть вашим токеном или логином и паролем от ЛК Leadhit и использовать его в своих целях.