
Рамблер / топ-100

Руководство разработчика

Версия: 1.3

Оглавление

Базовая настройка счетчика	3
Создание счетчика	3
Добавление счетчика на сайт	5
Донастройка счетчика	7
Методы по работе со счетчиком	7
Атрибуты счетчика	8
Передача параметров визита	10
Передача идентификатора пользователя	11
Разметка страниц сайта для аналитики блоков	12
Разметка разделов сайта	14
Учет хешей	15
Обратная связь	16
Изменения	17

Базовая настройка счетчика

Создание счетчика

С помощью сервиса Топ-100 можно настроить счетчик для отслеживания действий пользователей на сайте и получить код для вставки на страницы сайта.

Для этого необходимо:

1. [Зарегистрироваться в Топ-100](#).
2. [Создать новый счетчик](#), указав:
 - URL сайта
Основной домен сайта (обязательно с указанием протокола сайта: http:// или https://).
 - Примечание:** Если при регистрации выдается сообщение вида «URL сайта не уникален», значит, указанный сайт уже был зарегистрирован, и счетчик для него уже создан. Если вы по каким-то причинам потеряли доступ к этому счетчику и хотите перерегистрировать сайт, [обратитесь в поддержку](#).
 - Название счетчика (сайта)
Указанное название будет отображаться [в перечне подключенных счетчиков](#). А также [в каталоге Топ-100](#), если в настройках счетчика будет проставлен признак участия в рейтинге.

Примечание: В результате Топ-100 сохранит счетчик и присвоит ему уникальный идентификатор.

3. Указать требуемые настройки счетчика:
 - Признак участия в рейтинге Топ-100
Для участия в рейтинге необходимо указать:
 - Перечень категорий – указать одну категорию или несколько (см. [рекомендации по выбору темы](#)).
 - Перечень типов – опциональный параметр (см. [рекомендации по выбору типа](#)).
 - Перечень регионов – опциональный параметр (см. [рекомендации по привязке к региону](#)).
 - Перечень ключевых слов – указать одно слово или несколько.
 - Отображение статистики: доступна всем или закрытая.
При раскрытии статистики, любому посетителю Топ-100 будет доступен отчет «Посещаемость», «Популярность» и отчет «Сегодня» с данными по посетителям, визитам и просмотрам.
- Внимание:** Новый сайт будет добавлен в рейтинг Топ-100 только после прохождения модерации.
- Настройки счетчика: отображать виджет или нет, цвет виджета
Если указано, что виджет нужен, задан его цвет, то для его отображения необходимо при инициализации счетчика задать [в его атрибутах](#) идентификатор элемента `element`, где должен отображаться виджет. Также можно на каждой странице сайта, где требуется отобразить виджет, добавить вызов метода [drawLogoTo](#), передавая туда идентификатор соответствующего элемента.

- Признак необходимости отображения в статистике данных только с указанных доменов
Для отображения в статистике необходимо указать:
 - Перечень доменов, данные с которых необходимо показывать в статистике

Внимание: В статистику будут попадать данные только с этих доменов, вне зависимости от того, на каких ещё доменах/страницах будет установлен счётчик.

- Признак необходимости исключить указанные домены
Для исключения данных из статистики необходимо указать:
 - Перечень доменов, которые необходимо исключить из статистики – указать один домен или несколько.

Примечание: Возможность исключить статистику может потребоваться, например, для тестовых стендов, где ведётся разработка сайта.

- Признак необходимости аналитики блоков страниц
Для сбора аналитики блоков страниц необходимо указать:
 - Перечень data-атрибутов, [используемых в разметке блоков страниц](#) для сбора данных по количеству показов и нажатий на эти блоки.
Подробнее – см. описание атрибута [attributes_dataset](#).

По факту завершения ввода данных Топ-100 автоматически сгенерирует код счетчика для вставки на сайт для асинхронной загрузки.

Код будет содержать атрибуты, определенные в соответствии с заданными настройками счетчика, и в том числе будет включать:

- Путь к библиотеке, отвечающей за работу счетчика Топ-100:

```
//st.top100.ru/top100/top100.js
```

- Объявление элемента `options` с перечнем атрибутов счетчика:

- `project` – обязательный атрибут, проставляется идентификатор счетчика,
- `attributes_dataset` – опциональный атрибут, представляет собой перечень data-атрибутов, указанных в настройках аналитики блоков страниц.

Пример объявления в коде элемента `options` с перечнем атрибутов счетчика:

```
var options = {  
  project: 'PROJECT_ID',  
  attributes_dataset: [  
    'attribute1',  
    'attribute2'  
  ]  
};
```

После каждого изменения настроек счетчика (виджет/пиксель, цвет) и data-атрибута для сбора аналитики блоков страниц автоматически сгенерированный код будет обновляться.

Добавление счетчика на сайт

Выполнив все необходимые настройки в Топ-100, можно добавить соответствующий счетчик на сайт. Для этого в HTML-код сайта следует вставить сгенерированный код.

Для корректного подсчёта статистики необходимо установить код на все страницы сайта.

Внимание: Для корректного подсчета событий для single-page application следует использовать метод [trackPageview](#).

Код можно встроить в любое место HTML-кода. Рекомендуется вставлять его в начало страницы, чтобы счётчик успел загрузиться при коротком визите.

Пример кода счетчика для асинхронной загрузки:

```
<!-- Top100 (Kraken) Counter -->
<script>
(function (w, d, c) {
    (w[c] = w[c] || []).push(function() {
        var options = {
            project: 'PROJECT_ID'
        };
        try {
            w.top100Counter = new top100(options);
        } catch(e) { }
    });

    var n = d.getElementsByTagName("script")[0],
        s = d.createElement("script"),
        f = function () { n.parentNode.insertBefore(s, n); };
    s.type = "text/javascript";
    s.async = true;
    s.src =
        (d.location.protocol == "https:" ? "https:" : "http:") +
        "//st.top100.ru/top100/top100.js";

    if (w.opera == "[object Opera]") {
        d.addEventListener("DOMContentLoaded", f, false);
    } else { f(); }
})(window, document, "_top100q");
</script>
<noscript></noscript>
<!-- END Top100 (Kraken) Counter -->
```

Примечание: Можно добавить на страницу сайта код счетчика для синхронной загрузки. Но следует иметь в виду, что при синхронной загрузке, пока счётчик не инициализируется полностью, все остальные скрипты не будут загружены.

Пример кода счетчика для синхронной загрузки:

```
<!-- Top100 (Kraken) Counter -->
<script src="//st.top100.ru/top100/top100.js"></script>
<script>
    var options = {
        project: 'PROJECT_ID'
    };
    try {
```

```
        window.top100Counter = new top100(options);
    } catch(e) { }
</script>
<noscript></noscript>
<!-- END Top100 (Kraken) Counter -->
```

После подключения счётчика на сайт начнет собираться базовая статистика, доступная для просмотра в отчетах:

- Посещаемость,
- Технологии,
- Аналитика блоков (данные будут собираться, если в настройках счетчика стоит признак необходимости аналитики блоков страниц, и указаны data-атрибуты, использованные в разметке блоков страниц сайта),
- Содержание,
- Аудитория,
- Источники трафика,
- UTM метки.

Внимание: Статистика перестанет собираться, если на вашем сайте не будет ни одного посещения в течение девяти календарных дней. При этом в случае участия в рейтинге сайт оттуда пропадёт, и соответствующая настройка счетчика будет сброшена. Если в дальнейшем будет зафиксировано посещение, то сбор статистики возобновится. Участие в рейтинге автоматически восстановится, и сайт будет возвращен в рейтинг. Также будет выполнена повторная проверка [метаданных сайта](#).

Донастройка счетчика

Через UI Топ-100 выполняется настройка счетчика с базовым набором отслеживаемых событий.

Состав отслеживаемых событий можно при необходимости расширить (например, добавить отслеживание кликов на интересующие элементы интерфейса). Для этого необходимо скорректировать сгенерированный при помощи Топ-100 код счетчика, дополнив его вызовом интересующих функций и определением требуемых атрибутов.

В общем случае для каждого экземпляра счетчика могут вызываться разные методы и переопределяться различные атрибуты из числа поддерживаемых Топ-100.

Методы по работе со счетчиком

Поддерживаются следующие методы по работе со счетчиком:

Метод	Описание
<code>sendCustomVars</code>	Метод используется для передачи пользовательских переменных: принимает на вход объект <code>{param1:value1, ..., paramN:valueN}</code> и отправляет эти данные в счетчик. Подробнее про вызов метода – см. «Передача параметров визита» .
<code>trackPageview</code>	Метод используется для single-page application для реализации корректного подсчёта статистики. В код страницы необходимо добавить вызов метода <code>trackPageview</code> для созданного экземпляра счетчика в те моменты, когда необходимо отправлять события показа страницы.
<code>drawLogoTo</code>	Метод следует вызывать на страницах сайта с указанием места, где нужно разместить виджет. На вход необходимо передавать идентификатор элемента на странице, в который будет отрисован виджет (значение идентификатора <code>element</code> может быть также указано в атрибутах счетчика при инициализации). Вызов метода необходимо добавить в код каждой из страниц, где требуется разместить виджет.
<code>syncUserId</code>	Метод используется для передачи идентификатора пользователя: принимает на вход идентификатор и отправляет его в счетчик. Таким образом можно переопределить значение <code>user_id</code> , указанное в настройках счётчика при инициализации. Подробнее про передачу идентификатора пользователя – см. «Передача идентификатора пользователя» .

Внимание: Перед вызовом методов следует убедиться, что экземпляр счетчика создан и доступен. Данные не будут собираться, если счётчик ещё не создан, т.к. их не с чем связать.

Пример вызова в коде страницы метода `sendCustomVars`:

```
<script
  // ...
  Var customVars = {
    param1: value1,
    param2: value2
  };
  top100Counter.sendCustomVars(customVars);
  // ...
</script>
```

Пример вызова в коде страницы метода `sendCustomVars` с проверкой существования счетчика:

```
// ...
top100Counter && top100Counter.sendCustomVars(customVars);
// ...
```

Пример вызова в коде страницы метода `trackPageview`:

```
// ...
window.top100Counter.trackPageview();
// ...
```

Пример вызова в коде страницы метода `drawLogoTo`:

```
// ...
window.top100Counter.drawLogoTo(element_id);
// ...
```

Пример вызова в коде страницы метода `syncUserId`:

```
// ...
window.top100Counter.syncUserId(user_id);
// ...
```

Атрибуты счетчика

Возможно переопределение следующих атрибутов счетчика:

Атрибут	Описание
project	Обязательный атрибут. Идентификатор счетчика. Указывается в коде скрипта счетчика в качестве параметров атрибута <code>options</code> . Проставляется автоматически при генерации кода счетчика с помощью Топ-100.
element	Опциональный атрибут. Идентификатор элемента на странице, в который будет отрисован виджет. Указывается в коде скрипта счетчика в качестве параметров атрибута <code>options</code> .

Атрибут	Описание
user_id	<p>Опциональный атрибут. Идентификатор пользователя.</p> <p>Указывается в коде скрипта счетчика в качестве параметров атрибута options.</p> <p>Подробнее про передачу атрибута – см. «Передача идентификатора пользователя».</p>
trackHashes	<p>Опциональный атрибут. Флаг, характеризующий необходимость подсчета хешей, используемых для навигации по страницам.</p> <p>Указывается в коде скрипта счетчика в качестве параметров атрибута options.</p> <p>Подробнее про передачу атрибута – см. «Учет хешей».</p>
attributes_dataset	<p>Опциональный атрибут. Перечень data-атрибутов, используемых в разметке блоков страниц для сбора данных по количеству показов и нажатий на эти блоки.</p> <p>Указывается в коде скрипта счетчика в качестве параметров атрибута options.</p> <p>Проставляется автоматически при генерации кода счетчика с помощью Топ-100, если через UI были заданы настройки аналитики блоков страниц.</p> <p>Подробнее про передачу атрибута – см. «Разметка страниц сайта для аналитики блоков».</p>
chapters	<p>Опциональный атрибут. Перечень идентификаторов страниц сайта, в разрезе которых требуется учитывать события. Если атрибут не задан, то события учитываются по сайту в целом.</p> <p>Указывается в коде скрипта счетчика в качестве параметров атрибута options.</p> <p>Подробнее про передачу атрибута – см. «Разметка разделов сайта».</p>

Пример кода экземпляра счетчика с определением атрибута element:

```

<!-- Top100 (Kraken) Counter -->
  // ...
  var options = {
    // ...
    element: element_id,
    // ...
  };
  // ...
<!-- END Top100 (Kraken) Counter -->

```

Передача параметров визита

Можно расширить состав собираемой статистики по визитам пользователей, добавив ряд дополнительных параметров.

Для этого в код страницы необходимо добавить вызов метода [sendCustomVars](#) для созданного экземпляра счетчика и передать параметром объект с данными, которые надо отправить в счетчик. Это следует проделать со всеми страницами сайта, для которых необходимо отслеживать соответствующие события.

Накопленная статистика будет доступна для просмотра в отчете «Параметры визитов».

Пример отправки данных о шагах реализуемого сценария:

```
window.top100Counter.sendCustomVars({'<partner>::<product>::<utm_campaign>::<step_id>': 1})
```

В примере выше необходимо заменить переменные `<partner>`, `<product>`, `<utm_campaign>`, `<step_id>` на нужные вам.

Пример фрагмента сценария:

Шаг 1: Выбор услуги

➔ Никаких действий не нужно.

Шаг 2: Оформление услуги

➔ После загрузки формы необходимо вызвать функцию счётчика:

```
window.top100Counter.sendCustomVars({'ivan_and_partners::service::summercampaign::step_1': 1})
```

Шаг 3: Ознакомление с формой электронного сертификата на получение услуги

➔ После загрузки формы необходимо вызвать функцию счётчика

```
window.top100Counter.sendCustomVars({'ivan_and_partners::service::summercampaign::step_2': 1})
```

Примечание: Состав отслеживаемых шагов может быть различным в зависимости от специфики реализуемых на сайте пользовательских сценариев.

Передача идентификатора пользователя

Для сбора статистики в разрезе пользователей необходимо присвоить пользователям уникальные идентификаторы и включать их в данные, отправляемые в счетчик Топ-100. Это позволит соотносить в Топ-100 активности на сайте (просмотры, клики и т.п.) с заданными пользователями.

Передавать идентификатор пользователя в Топ-100 нужно следующим образом:

- Указывая идентификатор пользователя в качестве параметра `user_id` в коде счетчика при инициализации.

Пример:

```
<!-- Top100 (Kraken) Counter -->
// ...
var options = {
  // ...
  user_id: <USER_ID> || null
};
// ...
<!-- END Top100 (Kraken) Counter -->
```

- Либо передавая идентификатор через метод `syncUserId`.

Добавив в код страницы вызов `syncUserId`, можно переопределить значение `user_id`, указанное в настройках счётчика при инициализации.

Пример:

```
// top100Counter -- экземпляр ранее созданного счётчика
top100Counter.syncUserId(<USER_ID>);
// ...
```

Внимание: Если планируется использование метода `syncUserId`, то обязательно в коде счетчика необходимо указать параметр `user_id`. Если на момент инициализации счётчика пользователь неизвестен, то в качестве `user_id` нужно указать `null`. Без указания параметра `user_id` при вызове `syncUserId` будет напечатано предупреждение в консоли.

Так можно учитывать в собираемой статистике поведение пользователя. Например:

- Если требуется указать, что пользователь разлогинился, надо вызвать метод `syncUserId` с аргументом `null`.

Пример:

```
top100Counter.syncUserId(null)
```

- Если человек пришёл незалогиненный и залогинился в процессе работы с сайтом: в атрибутах счетчика при инициализации следует указать «`user_id: null`» и далее после авторизации передать нужный идентификатор пользователя через `syncUserId`.
- Если человек пришёл залогиненным, затем разлогинился и перелогинился: в атрибутах счетчика при инициализации следует указать исходный идентификатор пользователя, потом через `syncUserId` передать `null` (если это необходимо) и снова через `syncUserId` передать новый идентификатор.

Разметка страниц сайта для аналитики блоков

При необходимости можно настроить сбор данных по количеству показов и нажатий на интересующие блоки страниц сайта.

Для этого следует добавить в разметку страниц сайта data-атрибуты и указать в коде счетчика через параметр `attributes_dataset` используемые наименования data-атрибутов.

Накопленная статистика будет доступна для просмотра в отчете «Аналитика блоков»:

- При указании нескольких различных data-атрибутов в отчете «Аналитика блоков» статистика будет представлена в разрезе нескольких деревьев: название каждого data-атрибута станет первым уровнем дерева в отчете. Так, например, если необходимо собирать три типа событий: клики по партнёрским блокам, клики по блокам с новостями, клики по панели навигации, то для удобства работы с данными в отчете можно создать три data-атрибута: `partner`, `news`, `navigation`.
- Для отчёта «Аналитика блоков» собираются только показы блоков и клики по ним. В отчете также можно увидеть CTR блока. Показы и клики суммируются на каждом уровне дерева размеченных элементов сайта (осуществляется суммирование по вложенным элементам). Если же вам важно видеть посетителей и визиты, используйте разметку [параметров визита](#) (см. описание метода `sendCustomVars`) вместо аналитики блоков.

При разметке страниц сайта для сбора статистики по блокам следует придерживаться следующих принципов:

- Определение значения data-атрибута должно быть добавлено для каждого блока, по которому требуется собирать статистику.

Название полей, используемых для этого в разметке, должно соответствовать шаблону «data-*<название>*», где *<название>* – это наименование data-атрибута, из числа задаваемых в коде счетчика через параметр `attributes_dataset`.

Пример указания значения data-атрибута `just-test-attr` в разметке блока на сайте:

```
<a href='link' data-just-test-attr='YOUR_LOGICAL_NAME'>Link text</a>
```

Пример передачи data-атрибута `just-test-attr` в атрибутах счетчика:

```
<!-- Top100 (Kraken) Counter -->
// ...
var options = {
  project: PROJECT_ID,
  attributes_dataset: [ 'just-test-attr' ]
};
// ...
<!-- END Top100 (Kraken) Counter -->
```

Примечание: После клика на блок `YOUR_LOGICAL_NAME` будет зафиксировано событие “нажатие” `just-test-attr::YOUR_LOGICAL_NAME`.

- Для построения дерева логической разметки необходимо использовать вложенные контейнеры с проставленными data-атрибутами.

Пример:

```
<div data-just-test-attr='YOUR_LOGICAL_CONTAINER'>
  <a href='link' data-just-test-
attr='YOUR_LOGICAL_NAME_1'>Linktext</a>
```

```
<div data-just-test-attr='YOUR_LOGICAL_CONTAINER_2'>
  <a href='link' data-just-test-
attr='YOUR_LOGICAL_NAME_2'>Linktext</a>
</div>
</div>
```

Примечание: По нажатию на ссылку `YOUR_LOGICAL_NAME_2` будет зафиксировано событие `just-test-attr::YOUR_LOGICAL_CONTAINER::YOUR_LOGICAL_CONTAINER_2::YOUR_LOGICAL_NAME_2`.

- Существует возможность эмулировать вложенные блоки без привязки к DOM-разметке. Для этого на отдельный элемент DOM требуется назначить всю цепочку от 2-го уровня вложенности.

Пример:

```
<a href='link' data-just-test-
attr='YOUR_LOGICAL_CONTAINER::YOUR_LOGICAL_CONTAINER_2::YOUR_LOGICAL_NAME_2'>Link text</a>
```

Разметка разделов сайта

С помощью Топ-100 можно собирать статистику в разрезе заданного перечня страниц сайта.

Для этого каждой странице сайта нужно присвоить свой уникальный идентификатор. И указать его в добавляемом на страницу коде счетчика в качестве значения атрибута [chapters](#). Так событие просмотра пользователем страницы будет однозначно ассоциировано с соответствующим идентификатором.

Если на сайте существует определенная иерархия страниц, и необходимо ее отразить в статистике, то можно объединить идентификаторы страниц в деревья вида 'landing::page_1::page_2'. Для этого в коде счётчика каждой страницы в атрибуте chapters необходимо передавать список идентификаторов от корневой страницы до текущей.

Накопленная статистика будет доступна для просмотра в отчете «Разделы сайта».

Если в настройках счетчика задана иерархия страниц, то при подсчете событий для каждой страницы будет выполняться суммирование по дочерним страницам. В результате будет доступна как агрегированная статистика по головным страницам, так и отдельная статистика по дочерним.

На множестве страниц одного сайта можно задать несколько разных иерархий. Таким образом, в отчете «Разделы сайта» может существовать более чем одно дерево страниц.

Внимание: У каждой страницы должен быть строго один идентификатор, иначе обсчёт будет некорректным.

Пример кода экземпляра счетчика с определением атрибута chapters:

```
<!-- Top100 (Kraken) Counter -->
// ...
var options = {
  // ...
  chapters: [ // chapters страницы
    'landing',
    'page_1',
    'page_2'
  ]
};
// ...
<!-- END Top100 (Kraken) Counter -->
```

Учет хешей

Для отражения в статистике информации о хешах, используемых в навигации по страницам сайта, необходимо указать в настройках счетчика параметр `trackHashes: true`.

При включении этой настройки данные о хэшах появятся во всех отчётах Топ-100, предполагающих показ полных url-ов страниц. Например, в отчётах: «Популярное», «Страницы входа», «Страницы выхода» и «Разделы сайта».

Хэши в url-е будут учитываться, если на странице нет элемента с id равным хэшу (в этом случае предполагается, что это якорная ссылка). То есть `mysite.com#hash1` и `mysite.com#hash2` будут считаться разными страницами.

Пример кода экземпляра счетчика с определением атрибута `trackHashes`:

```
<!-- Top100 (Kraken) Counter -->
// ...
var options = {
  project: PROJECT_ID,
  trackHashes: true,
  // ...
};
// ...
<!-- END Top100 (Kraken) Counter -->
```

Обратная связь

В случае возникновения вопросов по использованию Топ-100 ознакомьтесь с [помощью по Топ-100](#). Если решение не найдено, обратитесь в [службу поддержки](#).

Изменения

Версия 1.0	Запуск обновленного Топ-100.
Версия 1.1	Реализовано автоматическое восстановление сайта в рейтинге в случае возобновления посещений после продолжительного отсутствия.
Версия 1.2	Доработана форма создания счетчика: необходимо обязательно указывать протокол в адресе сайта. Добавлена возможность идентификации пользователя по <code>user_id</code> .
Версия 1.3	Добавлена возможность учета хешей с помощью параметра <code>trackHashes</code> .