

1. Документация - ADCM.JS	2
1.1 1. Установка	3
1.2 2. Настройка триггеров	6
1.2.1 Config	8
1.2.2 Elapsed seconds	10
1.2.3 Init Call	11
1.2.4 NoBounce	12
1.2.5 Page down scroll	13
1.2.6 Push to server	14
1.2.7 Scroll percents	15
1.2.8 Scroll to	17

Документация - ADCM.JS

1. Установка

2. Настройка триггеров

- [Config](#)
- [Elapsed seconds](#)
- [Init Call](#)
- [NoBounce](#)
- [Page down scroll](#)
- [Push to server](#)
- [Scroll percents](#)
- [Scroll to](#)

Встроенные теги

Тэг	Значение
adcmjs_init	Отправляется при загрузке adcmjs
adcm_pc	(= adcm pixel call) Отправляется для всех запросов, вызываемых сторонним на ресурсе через метод .callpixel()

1. Установка

Какие данные получает код отслеживания?

После того как вы добавите на сайт загрузку библиотеки adcm.js, то при каждом посещении вашей страницы, будет производиться первичный запрос изображения(пикселя) с нашего сервера. Благодаря этому вы сможете получать самую разную информацию, в том числе:

- Порядок посещения страниц;
- Используемый им браузер и операционная система;
- Сайт-источник перехода;

Установка кода в html страницу

Для установки нужно сконфигурировать и разместить код в head секции страницы.

В приведенных ниже способах конфигурации происходит первичная настройка и загрузка скрипта.

Способ 1 - JS - Синхронный

Через вызов API adcm, в этом случае установка adcm.js должна происходить выше конфигурации и должна происходить синхронно, например:

Синхронный

```
<script src="http://tag.digitaltarget.ru/adcm.js"></script>
<script>
  window.adcm.configure({
    id:<ID>, //Required
    tags:['tag1','tag2'] //Not required
  });
</script>
```

Способ 2 - JS - Асинхронный

Через статическую предконфигурацию с использованием объекта с именем adcm_config.

В этом случае объект adcm_config необходимо объявить ранее установки скрипта adcm.js, но при этом скрипт adcm.js можно загружать асинхронно, например:

Синхронный

```
<script>
  var adcm_config = {
    id:<ID>, //Required
    tags:['tag1','tag2'] //Not required
  }
</script>
<script src="http://tag.digitaltarget.ru/adcm.js" async></script>
```

Способ 3 - Img/iframe/ajax

Можно отправлять события напрямую установив пиксель на страницу.

```

<iframe src="https://dmg.digitaltarget.ru/1/${ID кода}/i/i?i=${RND}&c=tg:one" />
```

Установка в email письма

По соображениям безопасности почтовые клиенты не позволяют выполняться JS коду, поэтому в письме можно разместить только pixel

Установка пикселя в письмо, аналогична установки в html-страницу, за исключением того, что пиксель сразу должен содержать все дополнительные данные, так как в письме нельзя отследить ничего, кроме загрузки изображения.

Установка в Баннер

Все аналогично с установкой на страницу.

Нюансы:

- Если в баннер нельзя вставить js код, то нужно использовать pixel
- Если вы хотите разделять показы разных баннеров, то должны передавать в tags соответствующий тег, который будет отличать баннеры друг от друга. Например это может быть ID баннера в рекламной системе
- Если с вашим баннером происходит дополнительное событие, помимо показа, то вам нужно будет ввести тэг, который будет отвечать характеризовать конкретный баннер и конкретное действие, которое вы отслеживаете.

Пример показ баннера:

```

```

Описание параметров:

- \${RND} - Нужно заменять на случайное число
- \${ID кода} - Нужно заменять на id вашего кода отслеживания

Описание тегов примера:

- bannerid_100 - Тэг, по которому вы будете отличать один баннер от другого
- show - Показ баннера
- banner_tracking - Тэг, для группировки данных, в случае, если код обслуживает не только работу с баннерами, а также вызывается на других системах.

Передача дополнительных данных (Tags)

Дополнительная информация. характеризующая запрос, или действие, передается в виде тегов, который могут быть использованы для построения сегментов

Символы, разрешенные для использования в тэгах: **a-zA-Z0-9_**

JS-Code

Тэги можно отправлять с помощью команды **adcm.call**

```
adcm.call({ tags: ['iphone', 'show', 'company_name'] });
```

Pixel (изображение)

В случае использования пикселя, на странице необходимо динамически создавать элемент ****, который содержит ссылку на пиксель, с информацией, которую нужно привязать к куке.

```

```

Описание параметров:

- `${RND}` - Нужно заменять на случайное число
- `${ID кода}` - Нужно заменять на id вашего кода отслеживания
- `iphone show company_name` - Тут перечислены тэги, через пробел, которыми вы маркируете пользователя, посетившего вашу страницу. Данные тэги приведены в качестве примера. Вы должны использовать свои тэги.

Отслеживание js-событий на стороне adcm.js кода

Если необходимо повесить отслеживание действий пользователя на стороне adcm.js кода, то для этого необходимо прислать нам список js-событий и селекторов для элементов, на которые нужно их повесить.

Каждому js-событию на элементе, будет утвержден тэг, который будет отправляться в амбердата, при возникновении js-события на странице.

2. Настройка триггеров

Вы можете сконфигурировать adcm для сбора метрик при возникновении определенных событий на странице

Scroll to

Триггер для маркировки элементов событиями появления в области видимости.

Системное имя триггера	scroll_to
Значение по умолчанию	enable: true

Scroll percents

Триггер сработает когда пользователь доскролит до процентов от высоты документа, указанных в конфигурации

Теги, отправляемые в Amberdata формируются по шаблону: **psp_{percent}**, где **{percent}** - проценты от TOP части экрана, при появлении которой в зоне видимости окна будет происходить фиксация события и отправка соответствующего тега.

Push to server

Триггер, срабатываем в момент инициализации пикселя на странице.

Позволяет повесить обработчики **"success"** и **"error"**, в которых можно обработать различные ситуации, связанные с проблемами при инициализации изображения на странице.

Триггер маппит функции "success" и "error" в стандартные обработчики событий тэга **img "img.onload"** и **"img.onerror"** в разрезе между инициализацией и загрузкой изображения.

Page down scroll

Триггер сработает, когда пользователь промотает страницу до конца

Системное имя триггера	page_down_scroll
Тег	<ul style="list-style-type: none">tg:pds

NoBounce

Через определенный промежуток времени (15 сек) отправляется тег noBounce, свидетельствующий о том, что пользователь еще находится на странице, поэтому мы помечаем его, как пользователя, который не попадет в сегмент отказов

Системное имя триггера	noBounce
Событие, отправляемое в amber	tg:nobounce

Init Call

Триггер инициализации adcm

Триггер срабатывает после загрузки всех внутренностей adcm. Он дергает `adcm.call()` без дополнительных параметров. По существу происходит "первичный сбор аудитории".

Системное имя триггера	init_call
------------------------	-----------

Elapsed seconds

Триггер отправляет данные о проведенном времени на странице через обозначенные временные промежутки.

Системное имя триггера	elapsed_seconds
Событие, отправляемое в <code>amber</code>	es:{sec} - где {sec} это количество секунд с момента загрузки страницы

Config

Триггер позволяет переопределять настройки adcm через html-элементы, на странице.

Это удобно использовать, чтобы задавать единый конфиг инициализации в `head` страницы и иметь возможность дополнительно конфигурировать adcm в контенте страницы, в зависимости от задач.

Системное имя триггера	config
Активирован по умолчанию (<code>enable</code>)	false

Config

Триггер позволяет переопределять настройки adcm через html-элементы, на странице.

Это удобно использовать, чтобы задавать единый конфиг инициализации в head страницы и иметь возможность дополнительно конфигурировать adcm в контенте страницы, в зависимости от задач.

Системное имя триггера	config
Активирован по умолчанию (enable)	false

Настройки, предустановленные по умолчанию

```
adcm_config = {
  triggers: {
    config: {
      enable: false,
      el_class_name: "",
      el_id: "",
      callback: function () {

      }
    }
  }
};
```

Пример использования

Сконфигурируйте adcm и обозначьте id или class элемента, в котором будут храниться настройки:

Конфигурация adcm.js

```
adcm_config = {
  triggers: {
    config: {
      // Активируем триггер
      enable: true,
      // Класс html элемента, который будет использован для поиска настроек
      el_class_name: 'adcm-config-trigger',
      // Callback будет вызван по окончании конфигурирования триггера
      callback: function () {

      }
    }
  }
};
```

Разместите метку с настройками в удобном месте на странице:

HTML элемент с атрибутами для конфигурации adcm.js

```
<div class="adcm-config-trigger" data-tags="yellow red white" data-profile-id="1" data-id="1"></div>
```

Использование атрибутов элементов "class" и "id"

Триггер позволяет задавать как имя класса, так и id элемента, из которого будут взяты настройки для adcm.js

Пример при работе с class

```
<script>
adcm_config = {
  triggers: {
    config: {
      enable: true,
      el_class_name: 'adcm-class-config-trigger',
    }
  }
};
</script>

<div class="adcm-class-config-trigger" data-tags="yellow" data-profile-id="1" data-id="1"></div>
<div class="adcm-class-config-trigger" data-tags="white"></div>
<div class="adcm-class-config-trigger" data-tags="red"></div>
```

Пример при работе с id

```
<script>
adcm_config = {
  triggers: {
    config: {
      enable: true,
      el_id: 'adcm-id-config-trigger',
    }
  }
};
</script>

<div id="adcm-id-config-trigger" data-tags="yellow red white" data-profile-id="1" data-id="1"></div>
```

⚠ Всегда стоит помнить, что на странице не может быть два элемента с одинаковыми ID.

Elapsed seconds

Триггер отправляет данные о проведенном времени на странице через обозначенные временные промежутки.

Системное имя триггера	elapsed_seconds
Событие, отправляемое в amber	es:{sec} - где {sec} это количество секунд с момента загрузки страницы
Активирован по умолчанию (enable)	false

Настройки, предустановленные по умолчанию

```
adcm_config = {
  triggers: {
    elapsed_seconds: {
      enable: false,
      timeouts: [5, 10, 30, 60, 120, 300],
      param_call: {},
      callback: function () {}
    },
  },
};
```

Пример полной конфигурации

```
adcm_config = {
  triggers: {
    elapsed_seconds: {
      // Триггер - включен/выключен
      enable: true,
      // Интервалы, через которые будут отправляться данные о присутствии пользователя на странице
      timeouts: [5, 10, 30, 60, 120, 300],
      // Дополнительные параметры, которые будут добавлены в вызов adcm.call() при отправке данных события
      param_call: {
        'tag': ['elapsed_seconds']
      },
      // Callback - функция, которая будет вызываться с каждым интервалом
      callback: function () {
        console.log('Trig: elapsed_seconds');
      }
    },
  },
};
```

Пример минимальной конфигурации

```
adcm_config = {
  triggers: {
    elapsed_seconds: {
      // Триггер - включен/выключен
      enable: true
    },
  },
};
```

Init Call

Триггер инициализации adcm

Триггер срабатывает после загрузки всех внутренних adcm. Он дергает `adcm.call()` без дополнительных параметров. По существу происходит "первичный сбор аудитории".

Системное имя триггера	init_call
Активирован по умолчанию (enable)	true

Отключение триггера

```
adcm_config = {  
  triggers: {  
    init_call: {  
      enable: false  
    }  
  }  
};
```

Вызов callback функции, после срабатывания init_call

```
adcm_config = {  
  triggers: {  
    init_call: {  
      callback: function () {  
        console.log('callback init call');  
      }  
    }  
  }  
};
```

NoBounce

Через определенный промежуток времени (15 сек) отправляется тег noBounce, свидетельствующий о том, что пользователь еще находится на странице, поэтому мы помечаем его, как пользователя, который не попадет в сегмент отказов

Системное имя триггера	noBounce
Событие, отправляемое в amber	tg:nobounce
Активирован по умолчанию (enable)	false

Настройки, предустановленные по умолчанию

```
adcm_config = {
  triggers: {
    elapsed_seconds: {
      enable: false,
      timeout: 15,
      param_call: {},
      callback: function () {}
    },
  },
}
```

Пример полной конфигурации

```
adcm_config = {
  triggers: {
    noBounce: {
      // Триггер - включен/выключен
      enable: true,
      // Время в милесекундах, через которые будут отправляться данные о присутствии пользователя на странице
      timeout: 15,
      // Дополнительные параметры, которые будут добавлены в вызов adcm.call() при отправке данных события
      param_call: {
        'tag': ['anytag']
      },
      // Callback - функция, которая будет вызвана в момент фиксации отказа пользователя
      callback: function () {
        console.log('Trig: noBounce');
      }
    },
  },
}
```

Пример минимальной конфигурации

```
adcm_config = {
  triggers: {
    noBounce: {
      // Триггер - включен/выключен
      enable: true
    },
  },
}
```

Page down scroll

Триггер сработает, когда пользователь промотает страницу до конца

Системное имя триггера	page_down_scroll
Тег	<ul style="list-style-type: none">tg:pds

Пример полной конфигурации

```
adcm_config = {
  triggers: {
    page_down_scroll: {
      // Триггер - включен/выключен
      enable: true,
      // Отступ в пикселях от нижней части страницы для подгонки под разные ситуации и необходимости
      // Допустим можно вычитать высоту футера, чтобы более точно обрисовать цель
      margin_bottom: 100,
      // Параметры, которые будут добавлены к вызову adcm.call()
      param_call: {
        tags: ['page_down_scroll_tag']
      },
      // Callback - функция, которая будет выполнена после срабатывания события
      callback: function () {
        console.log('Trigger "page_down_scroll" call');
      }
    },
  },
};
```

Пример минимальной конфигурации

```
adcm_config = {
  triggers: {
    page_down_scroll: {
      enable: true
    }
  }
};
```

Настройки, предустановленные по умолчанию

```
adcm_config = {
  triggers: {
    page_down_scroll: {
      enable: false,
      margin_bottom: 100,
      param_call: {},
      callback: function () {}
    },
  }
};
```

Push to server

Триггер, срабатываем в момент инициализации пикселя на странице.

Позволяет повесить обработчики **"success"** и **"error"**, в которых можно обработать различные ситуации, связанные с проблемами при инициализации изображения на странице.

Триггер маппит функции "success" и "error" в стандартные обработчики событий тэга **img** **"img.onload"** и **"img.onerror"** в разрезе между инициализацией и загрузкой изображения.

Системное имя триггера	push_to_server
Активирован по умолчанию (enable)	false

Особенности работы

- Если не вернуть контент при запросе пикселя, то будет вызван **"img.onerror"**
- Ответ сервера должен быть со статускодом 200/301, в противном случае, будут противоречия в ответе и нельзя гарантировать корректную обработку событий загрузки изображения

Пример обработки успешной и провальной попытки отправки данных на сервер

```
adcm_config = {
  id: 1,
  triggers: {
    push_to_server: {
      success: function(){
        console.log('Pixel load - ok');
      },
      error: function(){
        console.log('Pixel load - error');
      }
    }
  }
};
```

Scroll percents

Триггер сработает когда пользователь доскролит до процентов от высоты документа, указанных в конфигурации

Теги, отправляемые в Amberdata формируются по шаблону: **psp_{percent}**, где **{percent}** - проценты от TOP части экрана, при появлении которой в зоне видимости окна будет происходить фиксация события и отправка соответствующего тега.

Системное имя триггера	scroll_percents
Значение по умолчанию	enable: false

Пример полной конфигурации

```
adcm_config = {
  triggers: {
    scroll_percents: {
      // Триггер - включен/выключен
      enable: true,
      // Проценты от высоты страницы, на которой сработает триггер и отправит теги psp_{percent}
      percents: [25,50,60,70,90],
      // Параметры, которые будут добавлены к вызову adcm.call()
      param_call: {
        tags: ['scroll_percent_tag']
      },
      // Callback - функция, которая будет выполнена после срабатывания события
      callback: function (percentValue) {
        console.log('wasCalledOn:', percentValue);
      }
    },
  },
};
```

Пример минимальной конфигурации

```
adcm_config = {
  triggers: {
    scroll_percents: {
      // Триггер - включен/выключен
      enable: true,
    },
  },
};
```

Значения по умолчанию

```
scroll_percents: {  
  enable: false,  
  percents: [25,50,60,70,90],  
  param_call: {},  
  callback: function (percentValue) {  
  }  
},
```


Scroll to

Триггер для маркировки элементов событиями появления в области видимости.

Системное имя триггера	scroll_to
Значение по умолчанию	enable: true

Основные возможности:

- Триггер позволяет закрепить за каждым маркером html-элемент, который будет использован для детектирования прокрутки страницы до этого элемента.
- Для каждого маркера задается имя события, которое будет отправлено через `adcm.call()`
- Можно перечислить неограниченное количество маркеров для страницы.
- Есть возможность указать `callback` - функцию для каждого маркера, которая будет вызвана при отправке события.

Пример конфигурации

```
adcm_config = {
  triggers: {
    scroll_to: {
      enable: true,
      param_call: {
        tags: ['st1', 'st2']
      },
      // Процент от верхней части экрана, который используется для расчета зоны экрана, на которой
      // активируется триггер, при попадании обозначенных html-элементов на странице
      window_checkpoint_percent: 70,
      // Конфигурация селекторов для элементов-меток
      markers: [
        {
          // ID элемента в разметке страницы
          el_id: 'adcm-mark-id-bottom',
          // Тег, который будет отправлен в Amberdata
          event: 'sc_bottom',
          // Функция, которая будет вызвана при активации триггера
          callback: function () {
            console.log('BOTTOM')
          }
        },
        {
          el_id: 'adcm-mark-id-top',
          event: 'sc_top',
          callback: function () {
            console.log('TOP')
          }
        },
        {
          el_id: 'adcm-mark-class',
          event: 'sc_class',
          callback: function () {
            console.log('SCROLL CLASS')
          }
        },
        {
          el_class_name: 'adcm-mark',
          event: 'sc_mark_manyclasses',
          callback: function () {
            console.log('CLASS MARK')
          }
        }
      ],
      callback: function () {
        console.log('TRIGGER - ALL');
      }
    }
  }
};
```

Настройки, предустановленные по умолчанию

```
adcm_config = {
  triggers: {
    scroll_to: {
      enable: false,
      param_call: {},
      markers: [],
      window_checkpoint_percent: 50,
      callback: function () {}
    }
  }
};
```