

widget-utm-collector.js — ???????????????? ???? UTM/click-id ??? ???? GetCourse

Скрипт, который автоматически прокидывает рекламные метки (UTM + click-id) из URL и из first-touch storage в скрытые поля любой формы GetCourse. Работает с любыми виджетами и лендингами — не только с квизом Fitness Mama.

Заменяет vakas-tools `utm_to_gk_field.js`, у которого есть три критичных бага для современных лендингов (см. ниже).

????? ???? (vs vakas)

| Проблема vakas | widget-utm-collector |
|---|--|
| Слушает <code>DOMContentLoaded</code> — если виджет загружается позже (через инъект скрипта на действие пользователя), listener не сработает | <code>readyState</code> -чек: обрабатывает в любой момент после загрузки |
| Захватывает только <code>utm_*</code> — <code>fbclid/gclid/yclid</code> теряются | Ловит все 9 параметров |
| Нет first-touch: при возврате на сайт по закладке/прямому заходу метки потеряны | Сохраняет first-touch в <code>localStorage</code> на 90 дней |
| Заполняет только <code><input></code> | Заполняет <code><input></code> и <code><textarea></code> |
| Не диспатчит события — GC может игнорировать значение | Диспатчит <code>input</code> + <code>change</code> |
| Одна попытка заполнения | Три попытки (0мс, 500мс, 1500мс) для поздно отрисованных полей |

??? ??????????

Файл хостится в CMS GetCourse (раздаётся с того же домена `gc.fitnessmama.co` — нет CORS, нет задержек):

```
https://gc.fitnessmama.co/pl/cms/layout/js?id=92493&hash=abc932387ed49dc892bf87e6f0fa586d&bund  
le=1
```

`hash` пересчитывается автоматически при изменении файла в CMS — кэш сбрасывается сам.

Исходник в проекте: `widget-utm-collector.js` (в репозитории квиза).

???????????????????? ???? — ???
???????? ???? ? ?????

Чтобы подключить скрипт к **новой форме / новому лендингу**, нужно 3 шага.

??? 1 — ?????????? ?????????? ???? ? ??????

GetCourse

В настройках виджета формы добавьте Offer field или User field (по одному на каждую метку, которую хотите сохранять). Какие именно — зависит от того, какие данные нужны в сделке/контакте.

Минимальный набор (рекомендую):

| CSS-класс поля | Какое значение запишется | Куда сохранять (поле в GC) |
|---------------------------|--------------------------|-------------------------------------|
| <code>utm_source</code> | utm_source | deal_utm_source или user_utm_source |
| <code>utm_medium</code> | utm_medium | deal_utm_medium |
| <code>utm_campaign</code> | utm_campaign | deal_utm_campaign |
| <code>utm_content</code> | utm_content | deal_utm_content |
| <code>utm_term</code> | utm_term | deal_utm_term |

Расширенный набор (если ведёте таргет Facebook/Google/Yandex):

| CSS-класс | Значение |
|------------------------|---|
| <code>fbclid</code> | Facebook click ID |
| <code>gclid</code> | Google click ID |
| <code>yclid</code> | Yandex click ID |
| <code>utm_group</code> | Кастомный параметр (мы используем для URL квиза с ответами) |

Если лендинг сложный (квиз, многоэкранник, SPA) — форма может появляться **через минуты** после загрузки страницы, и URL к этому моменту может быть уже изменён (`history.pushState/replaceState`). В этом случае нужно **продублировать first-touch логику на самом лендинге**, чтобы метки сохранились в `localStorage` ещё при первой загрузке.

???????????? ???? ???? ??????????

Вставьте в `<head>` или в начало `<body>` лендинга:

```
<script>
(function(){
  var TRACK =
['utm_source','utm_medium','utm_campaign','utm_content','utm_term','utm_group','fbclid','gclid',
',yclid'];
  var KEY = 'fm_first_touch';
  var TTL = 90 * 86400000;
  var sp = new URLSearchParams(location.search || '');
  var data = {}, any = false;
  TRACK.forEach(function(k){ var v = sp.get(k); if (v){ data[k] = v; any = true; }});
  if (any) {
    try {
      if (!localStorage.getItem(KEY)) {
        localStorage.setItem(KEY, JSON.stringify({ts: Date.now(), data: data}));
      }
    } catch(e){}
  }
})();
</script>
```

Это «отщёлкивает» метки в storage **сразу при первой загрузке**, до того как сайт начнёт менять URL или пользователь перейдёт на другую страницу.

`widget-utm-collector.js` потом достанет их из того же `localStorage` ключа.

???? ?? ?????????? ????? ????????????? ?? ?????? ?
????????????????

Когда лендинг строит URL для перехода на форму (как наш `quiz-pink.gc.html` → `offer.html`), важно **протащить метки** через query string. Пример паттерна — функция `buildOfferUrl()` в `quiz-pink.gc.html:1185`:

```
// собираем все метки из URL/storage
const INCOMING_UTM = /* ... читаем url + localStorage ... */;

function buildOfferUrl() {
  const params = new URLSearchParams();
  // ... данные с лендинга ...

  // и в конце – дописываем метки
  for (const [k, v] of INCOMING_UTM) {
    params.set(k, v);
  }
  return OFFER_URL + '?' + params.toString();
}
```

Тогда форма получит метки и через URL (быстрый путь), и через localStorage (резервный).

???? ?? ?????????? (?????????????????)
?????????????)

Никакой клиентский скрипт не решит:

- **Прямой/органический трафик** — нечего ловить.
- **Инкогнито-режим** — localStorage стирается при закрытии вкладки.
- **Запрещённые cookies/storage** — скрипт молча игнорирует.
- **Кросс-девайс** (зашёл с телефона, купил с ноута) — localStorage привязан к устройству.
- **Прошло >90 дней** с первого захода — TTL истёк.
- **JS отключён** — скрипт не выполнится.

Для перекрытия этих случаев нужна **серверная атрибуция** (Facebook CAPI, Google Enhanced Conversions, GTM Server-side) — это отдельная задача, не для этого скрипта.

Реалистичный охват для рекламного трафика с UTM: **90-95%** сделок будут с метками.

???? ?????????? ????????

1. GetCourse → CMS → найти JS-слой с **id 92493**.
2. Внести правки в код.

3. Сохранить — GC автоматически пересчитает `hash` в URL подключения.
4. Если по какой-то причине браузер кеширует старую версию — добавить в URL `&v=2` (или поднять номер) для принудительного сброса.

???? ?????, ????? ????? ?? ????????

Чек-лист по убыванию вероятности:

1. **Проверить, что скрипт загружен:** Network → найти запрос `pl/cms/layout/js?id=92493` → статус 200, в Response виден код.
2. **Проверить first-touch:** Console → `localStorage.getItem('fm_first_touch')` → должен быть JSON с `ts` и `data` (если пользователь приходил с UTM).
3. **Проверить наличие полей с классами:** Console → `document.querySelectorAll('.utm_source, .utm_campaign').length` → должно быть `> 0`.
4. **Проверить, что класс на правильном элементе:** класс должен стоять на обёртке поля (Offer field/User field), внутри неё — `<input>` или `<textarea>`.
5. **Проверить, что скрипт реально отработал:** Console → `document.querySelector('.utm_source input').value` → не пусто.
6. **Если на лендинге URL подменяется** — убедиться, что first-touch код из секции «Дополнительно для лендингов» вставлен **до** любых `history.replaceState / pushState`.

???????????????? ?????????????????

| Где | Виджет | Лендинг | Статус |
|-------------------|---------------|-------------------------------------|-------------------------|
| Квиз Fitness Мама | Форма 1602182 | <code>gc.fitnessmama.co/quiz</code> | ☑ работает с 2026-05-18 |

При добавлении новых форм/лендингов — обновите эту таблицу.

????????? ?????????????????

- **2026-05-18** — первая версия. Замена `vakas-tools/utm_to_gk_field.js`.
 - readyState-чек (исправляет неотработку при поздней загрузке виджета).
 - Поддержка `fbclid/gclid/yclid`.
 - First-touch в `localStorage` на 90 дней.
 - Заполнение `<textarea>` в дополнение к `<input>`.
 - Диспатч событий `input / change`.
 - Три попытки заполнения с задержкой.

Revision #3

Created 2026-05-18 19:46:15 UTC by David

Updated 2026-05-18 19:49:52 UTC by David